# Algorithm for Proving the Knowledge of an Independent Vertex Set*

Pino Caballero-Gil and Candelaria Hernández-Goya

Dept. Statistics, Operations Research and Computing,
University of La Laguna, 38271 La Laguna, Tenerife, Spain
{pcaballe, mchgoya }@ull.es

**Abstract.** A new protocol is presented that allows to convince of the knowledge of a solution to the Independent Vertex Set Problem without revealing anything about it. It is constructed from a bit commitment scheme based on the hardness of the Discrete Logarithm Problem, which guarantees its efficient performance and formal security. One of its possible applications is node identification in ad-hoc wireless network because it does not require any authentication servers. Furthermore, recent works on network security has pointed out the importance of the design of efficient Zero Knowledge Proofs of Knowledge for the Independent Vertex Set Problem in broadcast models.

## 1 Introduction

Since the introduction of the notion of Zero-Knowledge Proof ($ZKP$) in the seminal paper of Goldwasser, Micali and Rackoff [12], it has proven to be very useful both in Complexity Theory and in Cryptography, playing in this latter field a major role as a building block in the construction of different cryptographic protocols [1]. It is remarkable that most of the different $ZKP$ that have been published so far are related to the same presumably intractable problems on which Public Key Cryptography is based. Such are the cases of the identification scheme based on the discrete logarithm problem [14], and the digital signature based on the computation of square roots [7].

One of the most relevant results regarding $ZKP$ was the demonstration that the existence of perfect zero-knowledge for an $NP - complete$ problem would cause the Polynomial Time Hierarchy to collapse [9]. However, this work deals with a different $ZKP$ known as computational $ZKP$, whose existence has been proven for any $NP$-problem under the assumption that a one-way function exists [11]. In the same work, the authors provided a $ZKP$ for the 3-coloring problem and suggested the use of standard reductions to achieve a $ZKP$ for any other $NP$-problem. The efficiency of the algorithm here proposed comes from a distinct approach based on an specific design of a computational $ZKP$ for a concrete $NP$-problem. It avoids the use of general reductions by combining tools from Number

* Research partly supported by the Spanish Ministry of Education and Science and the European FEDER Fund under Project SEG2004-04352-C04-03.

R. Moreno Díaz et al. (Eds.): EUROCAST 2005, LNCS 3643, pp. 346–354, 2005.

Theory and Graph Theory. Indeed, since this latter field is a dense source of $NP$-problems, several $ZKP$ for different graph problems such as isomorphism, non-isomorphism, hamiltonian circuits, clustering and independent vertex set have been previously introduced in the literature [13] but it is remarkable that all of them are based exclusively on Graph Theory problems. Special mention regards the algorithms proposed in [5] since they are described for the same problem, the Independent Vertex Set Problem ($IVSP$), as this present work. However here a number theoretical problem, the Discrete Logarithm Problem ($DLP$), is also involved in the design of the $ZKP$ in order to improve its efficiency and security.

All the aforementioned bibliographic references include proposals related in some way to the same basic graph problem, the Graph Isomorphism [3]. The major drawback of such an approach is due to the fact that the computational complexity of this problem is not yet known and furthermore the problem seems to be easy for most random graphs [8]. On the contrary, the present work proposes a new Computational $ZKP$ for the $IVSP$ whose security relies on the hardness of a number theoretical problem, the $DLP$, whose difficulty is generally assumed in Cryptography, [4]. On the other hand, while general $ZKP$ seem to be the most promising identification method in ad-hoc wireless networks, the concrete choice of the $IVSP$ as base of our proposal comes from the necessity of efficient $ZKP$ for such a problem in broadcast models pointed out in [10].

This paper is organized as follows. First we recall the basic requirements for the design of a $ZKP$. Then, in Section 3, the problems and notations that are used throughout the work are defined. In the following Section, the proposed $ZKP$ is fully described and its security is formally proved. The adequate choice of parameters and the performance of the scheme are analyzed in Section 5. Finally, several conclusions and open questions are drawn in Section 6.

## 2 *ZKP* Design

A Zero-Knowledge Proof of Knowledge ($ZKPK$) may be defined as a two-party cryptographic protocol that allows an infinitely powerful prover Alice (A) to convince a probabilistic polynomial time verifier Bob (B), beyond any reasonable doubt, that she knows some verifiable information such as the solution of a given difficult problem, but in a way that does not help him to determine anything about this information.

The three main characteristic properties of $ZKPK$ are completeness (which means that if the claim is valid, then A convinces B of it with very high probability), soundness (if the claim is not valid, then B is convinced of the contrary with very small probability), and zero-knowledge (B does not receive any other information except for the certainty that the claim is valid). This latter property may be checked through the demonstration that the prover A can be replaced by an efficient (expected polynomial time) simulator which generates an interaction indistinguishable from the real one. The main difficulty of this proof, which is usually based on a constructive specification of the way such a simulator pro-

ceeds, is to achieve that the simulator convince the verifier about the knowledge of the secret information without actually having it. Generally, this problem is solved thanks to the rewinding capability of the simulator, which may use several tries to answer the verifier without letting him know how many tries the simulator has used.

Two basic variants of zero-knowledge may be distinguished depending on the assumed computing power of possible dishonest parties. Computational zero-knowledge arises when it would take more than polynomial time for a dishonest verifier to obtain some information about the secret, whereas perfect zero-knowledge involves that even an infinitely powerful cheating verifier could not extract any information. Both previous notions can also be characterized through the amount of computational resources necessary to distinguish between the interaction generated by the simulator and the verifier, and the one associated to the prover and the verifier.

Generally, bit commitment and cut-and-choose techniques are basic ingredients for the design of $ZKPK$. In these cases, A 'cuts' her secret solution in several parts, commits to them, and afterwards B chooses at random one of those parts as a challenge. The typical design of $ZKPK$ is also based on the existence of a concrete possibility of fraud: a cheater is usually able to answer to some types of challenges (for which he was prepared in advance) but not for all of them. So, most protocols are designed as interactive challenge-response schemes in such a way that some of A's possible responses prove A's knowledge of the secret solution, whereas the others guarantee against A's possible fraud. More concretely, an answer to one question gives no information (zero-knowledge), while answering all the questions is proved to reveal prover's knowledge (soundness). So, the security is based on the impossibility that the prover can predict verifier's questions. Also typically $ZKPK$ consist of several iterations of the atomic subroutine described below, so that by repeating it an enough number of times the verifier's confidence in the prover's honesty increases because the global fraud probability becomes smaller with the number of iterations. A. Consequently, this number $m$ of iterations should be agreed by A and B according to their different interests.

## 3 Notations and Definitions

As mentioned before, the two problems that constitute the base of the proposed algorithm are the Independent Vertex Set Problem ($IVSP$) and the Discrete Logarithm Problem ($DLP$).

On the one hand, the $DLP$ may be described as follows. Let $p$ be a prime, let $g$ be a generator of $\mathbb{Z}_p^*$ (the multiplicative group of integers modulo p) and let $x$ be an integer between 0 and $p-1$. Define $DLP_{p,g}(x)$ to be $y$ such that $0 < y < p$, $g^y = x(mod\ p)$. Such a problem is in $NPI$ class, which means that no probabilistic polynomial algorithm is known for solving it. The intractability assumption of the $DLP$ has been yet used on public-key cryptography and as single base of a $ZKPK$ [4].

On the other hand, the $IVSP$ is an $NP-complete$ problem that may be defined as follows. Given a graph $G=(V,E)$, it consists in finding a size $k$ subset $I \subseteq V$ (independent vertex set) such that no two vertices in $I$ are joined by an edge in $E$.

A useful method to hide an independent vertex set in a graph such that it is resistant to general heuristic approaches has been described in [2]. This method tries to balance the vertices degree sequence so that there is no difference between those belonging to the independent vertex set and the others. Consequently, due to its robustness it may be used in the instances generation of the proposed algorithm.

The concrete choice of the $IVSP$ as base for our proposal seems to be quite convenient since recent work on network security [10] has pointed as an important contribution to the field of the design of protocols for broadcast channels the definition of efficient $ZKPK$ for the $IVSP$.

Since the $IVSP$ is NP-complete, by the result of [9], we know that this problem cannot have perfect $ZKPK$ unless the polynomial hierarchy collapses, so the $ZKPK$ for the $IVSP$ described in the next section is a computational $ZKPK$.

## 4 Zero-Knowledge Proof of Knowledge for the Independent Vertex Set Problem

A Computational $ZKPK$ for the $IVSP$ that uses a bit commitment scheme based on the $DLP$ is now presented. In our proposal A's inputs are a graph $G=(V,E)$ and an integer $k$, and her goal is to convince B that she knows a size $k$ independent vertex set of $G$.

In a pre-processing stage, A generates at random a graph $G$ with $n$ vertices and an embedded secret independent vertex set $I$ of size $k$ through the method described in [2], and publishes her inputs $(G,k)$. Such a construction allows that the embedded and secret independent vertex set $I$ may be used in practice as A's secret identification because hiding the secret subset $I$ takes polynomial time. During the processing stage of the algorithm, in each iteration A generates a $c$-coloring of $G$ where the $k$ vertices of $I$ have the same colour that is not used for any other vertex. It must be pointed out that the number of colours $c$ is not restricted to any value, so the computation of such a $c$-coloring takes polynomial time.

A's secret commitment is then formed by $c$ binary $n$-dimensional vectors $a_i = (a_i^j), a_i^j \in \{0,1\}, i=1,2,...,c, j=1,2,...,n$, where each position corresponding to a vertex $j$ colored with colour $i$ contains a one and the rest contains a zero. The cardinality of each vertex subset defined by the c-coloring is given by the Hamming weight of vector $a_i$, $W_H(a_i)$ (where the Hamming weight of a vector is simply the number of nonzero digits in the vector), which is a value which plays a special role in the algorithm. After an initialization stage where A and B agree on integers $m$ and $c$, primes $p_i$ $(i=1,2,...,c)$, generators $g_i$ of $\mathbb{Z}_{p_i}^*, (i=1,2,...,c)$

and random integers $r_i \in \mathbb{Z}^*_{p_i}$ $(i = 1, 2, ..., c)$, the $IVSP - ZKPK$ algorithm consists of $m$ iterations of the following four steps:

Atomic Subroutine:

**Commitment step:** A generates the vectors $a_i = (a_i^1, a_i^2, ..., a_i^n), i = 1, 2, ..., c$, chooses secret random integers $y_j \in \mathbb{Z}^*_{p-1},\ j = 1, 2, \ldots, n$, with $p = \min_{i=1,\ldots,c}\{p_i\}$ and commits to such parameters by sending to $B$ the n-dimensional vectors: $V_i = ((r_i^{a_i^j} \cdot g_i^{y_j}) mod\ p_i), (i = 1, 2, \ldots, c,\ j = 1, 2, \ldots, n)$.

**Challenge step:** B chooses at random and sends to A one bit $b$, and if $b = 0$, he also sends two random adjacent vertices $v$ and $w$.

**Response step:** A sends to B:

- if $b = 0$, the integers $y_v$ and $y_w$
- else, the integers $y = \sum_{j=1}^{n} y_j$ and $W_H(a_i) = \sum_{j=1}^{n} a_i^j,\ i = 1, 2, \ldots, c$.

**Verification step:** $B$ checks whether the values provided by A in previous steps are correct, that is to say,

- when $b = 0$, from the elements $V_i^v$ and $V_i^w$ (i=1,2,...,c), B checks that only two different vectors exist where the components associated to $v$ and $w$ have the value 1. $(\exists! h, l \in \{1, 2, \ldots, c\}\ |h \neq l,\ a_h^v = a_l^w = 1)$.
- when $b = 1$, B checks that
  - $\sum_{i=1}^{c} W_H(a_i) = n$,
  - $\exists i \in \{1, 2, ..., c\} | W_H(a_i) = k$,
  - $\forall i \in \{1, 2, ..., c\} : (\prod_{j=1}^{n} r_i^{a_i^j} \cdot g_i^{y_j}) = (r_i^{W_H(a_i)} \cdot g_i^{y}) mod\ p_i$.

Note that in the previous algorithm the verification step is only possible thanks to B's knowledge of $g_i$, $p_i$, $r_i$ and $V_i$ and the use of efficient modular exponentiation methods.

In order to prove the security of the protocol, we follow the approach of [6], first proving completeness, then soundness and finally the zero-knowledge property.

**Theorem 1.** *The $IVSP-ZKPK$ algorithm is a computational zero-knowledge proof of knowledge for the independent vertex set problem.*

*Sketch of Proof.* In order to prove that completeness is met there should be shown that if A knows a $k$ size independent vertex set in $G$ and both participants follow correctly the protocol, then the verifier B always accepts the proof.

If challenge $b = 0$ is requested by B, he should check that two chosen adjacent vertices $v$ and $w$ are colored with exactly two different colours. To achieve this, he computes $g_i^{y_v}$ and $g_i^{y_w}, \forall i = 1, 2, \ldots, c$ and compares these values with the corresponding components in the committed vectors $V_i$. If A has built an

appropriate coloring, B finds that there is a unique vector $V_h$ where the v-th component coincides with $r_h \cdot g_h^{y_v}$, whereas the $v$-th component in the other vectors is equal to $g_i^{y_v}$. The same occurs for some vector $V_l$ and $w$-th component.

If B chooses bit $b = 1$ as challenge, both the Hamming weight of coloring vectors $a_i$, and the value $y$ provided by A in the response step, allow B to check the following items:

- $\sum_{i=1}^{c} W_H(a_i) = n,$, so all the vertices are colored using only one colour.
- $\exists i \in \{1, 2, ..., c\}$ $|W_H(a_i) = k$, so there is at least a size $k$ independent vertex set to whose vertices the coloring has assigned the same colour.
- $(\prod_{j=1}^{n} r_i^{a_i^j} \cdot g_i^{y_j}) = (r_i^{a_i^1} \cdot g_i^{y_1}) \cdot (r_i^{a_i^2} \cdot g_i^{y_2}) \cdots (r_i^{a_i^n} \cdot g_i^{y_n}) = (r_i^{a_i^1} \cdot r_i^{a_i^2} \cdots r_i^{a_i^n}) \cdot (g_i^{y_1} \cdot g_i^{y_2} \cdots g_i^{y_n}) = r_i^{\sum_{j=1}^{n} a_i^j} \cdot g_i^{\sum_{j=1}^{n} y_j} = r_i^{W_H(a_i)} \cdot g_i^{y}$, so the $c$ committed vectors have been properly computed.

In order to prove soundness, if the prover A does not know any size $k$ independent vertex set in $G$ and the verifier B follows correctly the protocol, then no matter how A plays, B should reject the proof with high probability. In such a case, A has basically two possible ways to try to fool B. She could use an incorrect c-coloring of $G$ with some vertex subset of cardinality $k$, or she could compute correct coloring vectors $a_i$ with no vertex subset of Hamming weight $k$. In the first case, there exists at least a vector $a_i, i \in \{1, 2, \cdots, c\}$ such that two adjacent vertices are colored with the same colour, and hence B could detect the fraud if $b = 0$ with probability at least $1/|E|$ in each iteration. When a dishonest prover A uses a correct coloring, if B chooses bit $b = 1$ he always detects the fraud when checking the existence of a vertex subset of size k colored with the same colour ($\exists i \in \{1, 2, ..., c\} | W_H(a_i) = k$).

Hence, under the assumption that a dishonest prover A chooses at random the way to commit fraud, after $m$ successive and independent iterations with uniformly random chosen challenges, the probability that A successfully cheats B is upper bounded by $(2^{-2m} \cdot (3 - 1/|E|)^m)$.

Regarding computational zero-knowledge, we need to show that the prover A conveys no knowledge to any possible verifier, including ones that deviate arbitrarily from the protocol. From the received witnesses, B should be able to obtain the committed c-coloring and consequently the independent vertex set $I$, only if he is able to solve the $DLP$. So, according to the simulation paradigm, it is possible to build an expected polynomial time simulator that generates a probability distribution which is polynomially indistinguishable from the distribution induced during the interaction between A and B. In particular, the simulator first tries to guess B's challenge so chooses a random bit $b'$. Then, if $b' = 0$ the simulator generates at random correct coloring vectors $a_i$ and consequent witnesses passing verification. Else, if $b' = 1$, false coloring vectors $a_i$ are generated such that there is a vertex subset with cardinality $k$ colored using the same colour that is not used for any other vertex.

The simulator tries one of both possible challenges at random and if it coincides with B's challenge $b$ (which happens with probability exactly $1/2$) then its output is polynomially indistinguishable from A's output. Else, it reinitiates.

So, in an expected polynomial time the described simulator may replace A, and therefore computational zero-knowledge is proven. □

The proposed $IVSP-ZKPK$ algorithm may be applied in a quite natural way as an identification protocol with advantages compared to other similar schemes. In the corresponding identification scheme the public file containing records for each user should consist of each name and the respective auxiliary identification information composed of a graph $G$ and the size $k$ of the secret embedded independent vertex set. All users should have free read access to this public file. When a user A wishes to convince B of her identity, she invokes the identification protocol with the public file record corresponding to her name as a parameter, and B verifies her record in the public file and proceeds executing his role in the protocol. So, soundness property of the $IVSP-ZKPK$ algorithm yields that an impersonation attack is practically infeasible.

A different application of the independent vertex set Problem in Cryptography has been recently addressed in [5]. This application has to do with the computation of an access structure taking into account the relationship graph. By using the algorithm here proposed it is also possible to convince an adversary of the validity of such an access structure without revealing who the honest members are. Another cryptographic use of this problem was addressed in the same paper where the authors proposed tackling the key scrow problem via the independent vertex set problem. In general, different cryptologic applications of this problem may be described anywhere it is important that the participants with access privileges to a determined information form an independent vertex set.

## 5 Complexity Analysis

This section specifies both the techniques used to build the instances that are necessary for the $IVSP-ZKPK$ algorithm, and the complexity of the different operations that are required.

First of all, the random generation of a graph $G$ with $n$ vertices and an embedded independent vertex set of size $k$ takes $O((n-k)^2)$. Then, the coloring is accomplished through a well-known greedy heuristic that takes $O(n^3)$ under the worst-case analysis. In order to build the instances of the $DLP$, A should generate $c$ prime numbers and use the modular exponentiation algorithm $c \cdot n$ times whose complexity is $O(c \cdot n \cdot \log^3 l)$ (where $l = \max_{i=1,2,\ldots,c} \{p_i\}$). The response associated to the challenge $b = 0$ takes constant time, whereas if the challenge chosen by $B$ is $b = 1$ the computation of the answer takes linear time. Hence, the computational complexity associated to A may be estimated by $O(n^3)$, which confirms that the assumed restriction of the capabilities of A to polynomial time is possible.

The operations corresponding to $B$ when he has to verify the answers provided by A depend on the challenge chosen by him. So, if the challenge is $b = 0$, then B should compute several exponential operations that take $O(c \cdot log^3 l)$. On the other hand, also the operations associated to the process of verification when the challenge is $b = 1$ are of order $O(c \cdot log^3 l)$.

Regarding the communication complexity, in the pre-processing stage the prover should publish the graph $G = (V, E)$, so $O(|E| \cdot log(n))$ determines the size of the file containing it. Furthermore, the committed vectors are formed by $n \cdot c$ integers and the response to both challenges is composed by two integers, so the corresponding communication complexity of the algorithm is $O(m \cdot c \cdot log(n))$.

So, a question regarding the use of $IVSP - ZKPK$ algorithm that deserves special attention is the choice of parameters such as $n, c$ and $p_i$. In order to guarantee the security of the scheme, graph's size $n$ and primes $p_i$ should be large enough, whereas it is convenient that the number of colours $c$ be small in order to reduce communication complexity. Also generation and computer representation of random graphs $G$ are important factors having a profound effect on the complexity of the algorithm. In particular, experimental analyzes of generations of graphs guaranteeing the difficulty of the $IVSP$ recommends the use of $n > 1000$, [2].

## 6 Conclusions

In this work a new computational zero-knowledge proof of knowledge has been described for the independent vertex set Problem. Its validity based on the difficulty of the discrete logarithm problem has been formally established.

Among known schemes based on NP-complete problems, the one proposed in this paper is one of the most efficient when parameters are adequately chosen. Due to its efficiency and to the basic problem of the independent vertex set problem, the proposed scheme may be used for identification and access control systems, as well as for the design of secure network protocols for broadcast channels and for the computation of access structures.

A full comparison among the computational efficiency of the proposed algorithm and other previous schemes is part of a work in progress. Also a forthcoming version of this work will include some open questions such as the range of parameters that guarantees both security and efficient performance.

## References

1. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. Journal of Computer and System Sciences **37** (1988) 156–189.
2. Brockington, M., Culberson, J.: Camouflaging independent vertex sets in quasi-random graphs. Johnson, D., Trick, M., eds.: Cliques, Coloring and Satisfability. Volume XXVI. American Mathematical Society (1994) 75–88.
3. Caballero, P., Hernández, C.: Strong Solutions to the Identification Problem. Wang, J., ed.: Computing and Combinatorics, Berlin, Springer-Verlag (2001) 257–261 Lecture Notes in Computer Science Volume 2108.
4. Chaum, D., Evertse, J.H., van de Graaf, J., Peralta, R.: Demonstrating possession of a discrete logarithm without revealing it. Odlyzko, A.M., ed.: Advances in Cryptology - Crypto '86, Berlin, Springer-Verlag (1986) 200–212 Lecture Notes in Computer Science Volume 263.

5. Desmedt, Y., Wang, Y.: Efficient Zero-Knowledge Protocols for Some Practical Graph Problems.Third Conference on Security in Communication Networks'02, Springer-Verlag (2003) 296–308 Lecture Notes in Computer Science Volume 2576.
6. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. Journal of Cryptology **1** (1988) 77–95.
7. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. Odlyzko, A.M., ed.: Advances in Cryptology - Crypto '86, Berlin, Springer-Verlag (1986) 186–194 Lecture Notes in Computer Science Volume 263.
8. Fortin, S.: The Graph Isomorphism Problem. Technical Report TR 96-20, University of Alberta, Department of Computer Science (1996).
9. Fortnow, L.: The complexity of perfect zero-knowledge. Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, STOC 87. (1987) 204–209.
10. Franklin, M., Wrigh, R.: Secure communication in minimal connectivity models. Journal of Cryptology **13** (2000) 9–30.
11. Goldreich, O., Micali, S., Wigderson, A.: How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. Odlyzko, A.M., ed.: Advances in Cryptology - Crypto '86, Berlin, Springer-Verlag (1986) 171–185 Lecture Notes in Computer Science Volume 263.
12. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems.Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing (STOC 85). (1985) 291–304.
13. Santis, A.D., Crescenzo, G.D., Goldreich, O., Persiano, G.: The graph clustering problem has a perfect zero-knowledge proof. Information Processing Letters **69** (1999) 201–206.
14. Schnorr, C.P.: Efficient identification and signatures for smart cards. Brassard, G., ed.: Advances in Cryptology - Crypto '89, Berlin, Springer-Verlag (1989) 239–252 Lecture Notes in Computer Science Volume 435.