The Graph Clustering Problem has a Perfect Zero-Knowledge Proof[*]

Alfredo De Santis [†] Giovanni Di Crescenzo [‡] Oded Goldreich [§]
Giuseppe Persiano [¶]

January 27, 1998

Abstract

The input to the *Graph Clustering Problem* consists of a sequence of integers $m_1, ..., m_t$ and a sequence of $\sum_{i=1}^{t} m_i$ graphs. The question is whether the equivalence classes, under the graph isomorphism relation, of the input graphs have sizes which match the input sequence of integers. In this note we show that this problem has a (perfect) zero-knowledge interactive proof system.

Keywords: Graph Isomorphism, Zero-Knowledge Interactive Proofs.

1 Introduction

The remarkable notion of perfect zero-knowledge proofs was introduced by Goldwasser, Micali and Rackoff [GoMiRa]. A perfect zero-knowledge proof system is a method for a prover to convince a polynomial-time bounded verifier with very high probability that a certain assertion is true without revealing any additional information (in an information-theoretic sense). Not many are the languages which have been shown to have a perfect zero-knowledge proof system; in particular, all of them share number-theoretic or random self-reducibility properties [GoMiRa, GoMiWi, GoKu, ToWo, BoFrLu], or are obtained by formulae compositions over such languages [DeDiPeYu]. Moreover, it is known that it is unlikely that all languages in NP have a perfect zero-knowledge proof system, unless unexpected consequences in complexity theory happen [Fo, AiHa]. Still, perfect zero-knowledge proofs are worth to investigate since they seem to capture the intrinsic properties of the zero-knowledge notion.

Trying to extend the domain of perfect zero-knowledge, we consider the Graph Clustering Problem (GCP), formally defined below. Loosely speaking, the input to the problem consists of a sequence of integers and a sequence of graphs, and the question is whether the equivalence classes – under the

[*]This paper combines unpublished manuscripts by Goldreich [G96], and by De Santis, Di Crescenzo and Persiano [DeDiPe].

[†]Dipartimento di Informatica ed Appl., Università di Salerno, 84081 Baronissi (SA), Italy. E-mail: ads@dia.unisa.it.

[‡]Computer Science and Engineering Department, University of California San Diego, 92093 La Jolla, CA, USA. E-mail: giovanni@cs.ucsd.edu.

[§]Work done while visiting MIT. E-mail: oded@lcs.mit.edu.

[¶]Dipartimento di Informatica ed Appl., Università di Salerno, 84081 Baronissi (SA), Italy. E-mail: giuper@dia.unisa.it.

graph isomorphism relation – of the given graphs have sizes which match the given sequence of integers. GCP is related to Graph Isomorphism, known to have a perfect zero-knowledge proof [GoMiWi], and yet seems more complex than it. The knowledge-complexity (cf., [GoMiRa, GoPe]) of GCP is easily bounded by a function of the integral part of the instance, and for some years this problem served as an example of a non-trivial case of bounded knowledge-complexity. We show that GCP has a perfect zero-knowledge proof system.

Our proof proceeds by expressing GCP as a conjunction of a constant number of monotone formulae over Graph Isomorphism (or Non-Isomorphism) statements. This allows us to apply a general result of [DeDiPeYu]. Our technique can be adapted to other random self-reducible languages, such as quadratic residuosity, discrete log, etc.

2 Preliminaries

We recall the definition of perfect zero-knowledge proof systems [GoMiRa].

Definition 1 *A pair of probabilistic interactive machines,* (P,V) *is called an* interactive proof system *for the language* L *if* V *runs in polynomial-time and*
1. (Completeness) *For all* $x \in L$*, and all constants* c*,*

$$\Pr(\mathrm{P} \leftrightarrow \mathrm{V}(x) = \mathrm{ACCEPT}) \geq 1 - |x|^{-c}.$$

where here and below $\mathrm{P} \leftrightarrow \mathrm{V}(x)$ *denotes* V*'s output after interacting with* P *on common input* x*.*
2. (Soundness) *For all* $x \notin L$*, all constants* c*, and all probabilistic Turing machines* Pj*,*

$$\Pr(\mathrm{P}^* \leftrightarrow \mathrm{V}(x) = \mathrm{ACCEPT}) \leq |x|^{-c}.$$

Definition 2 *An interactive proof system* (P,V) *for* L *is called* perfect zero-knowledge proof system *for* L *if for every probabilistic polynomial-time interactive machine,* V^**, there exists a probabilistic polynomial-time machine,* S_{V^*} (called the simulator) *such that for all* $x \in L$*,*
1. With probability at most $1/2$*, the output* $S_{\mathrm{V}^*}(x)$ *is a special failure symbol, denoted* $\perp$*.*
2. Conditioned on $S_{\mathrm{V}^*}(x) \neq \perp$*, the distribution* $S_{\mathrm{V}^*}(x)$ *is identical to* $\mathrm{P} \leftrightarrow \mathrm{V}^*(x)$*.*

The above definition is taken from [G95], and is at least as strong as the original definitions [GoMiRa] where the simulator is allowed to run for *expected* polynomial-time (but is not allowed to fail).

The Graph Clustering Problem. The Graph Clustering Problem, GCP, is defined as follows:

- Input: a sequence $(G_1, \ldots, G_m; m_1, \ldots, m_c)$, where each G_i is an n-node graph, each m_j is a positive integer, and it holds that $\sum_{j=1}^{c} m_j = m$.
- Question: is there a partition $(C_1, \ldots, C_c)$ of $\{G_1, \ldots, G_m\}$ such that $|C_j| = m_j$, for $j = 1, \ldots, c$, and
    1. for any $i \in \{1, \ldots, c\}$ and any $G_k, G_l \in C_i$, it holds that the graphs G_k and G_l are isomorphic;
    2. for any $i, j \in \{1, \ldots, c\}$ such that $i \neq j$, and any $G_k \in C_i$ and $G_l \in C_j$, it holds that the graphs G_k and G_l are not isomorphic.

Each set C_j $(j = 1, \ldots, c)$ will be called a *cluster*. We will use the symbols $\approx, \not\approx$ to denote isomorphism and non-isomorphism between graphs, respectively.

Threshold. By $\texttt{Thresh}_{t,n}$ we denote the t-out-of-n threshold formula; that is, $\texttt{Thresh}_{t,n}(x_1,...,x_n)$ holds if and only if at least t out of the n Boolean variables (i.e., x_i's) are true. We use throughout the paper the fact that it is possible to construct, in time polynomial in n and t, a monotone formula that computes $\texttt{Thresh}_{t,n}(x_1,...,x_n)$. (A proof of existence of such a formula is given in [Va]; the construction of such a formula can be derived by using the result in [AKS]; other constructions are given in [Fr].)

Tools. We use the following results from [DeDiPeYu].

Theorem 1 [DeDiPeYu] *The language of all true monotone formulae over graph isomorphism statements has a perfect zero-knowledge proof system.*

Theorem 2 [DeDiPeYu] *The language of all true monotone formulae over graph non isomorphism statements has a perfect zero-knowledge proof system.*

3 A perfect zero-knowledge proof system for GCP

In this section we present a perfect zero-knowledge proof system for the language GCP. This result can be extended to other random self-reducible languages, as quadratic residuosity and discrete log, using known techniques. We start with an informal description of the protocol and then we give a formal description and a proof of correctness.

An informal description. The protocol can be divided into four parts. The first two parts are used to show that the m input graphs can be divided into *exactly* c (isomorphism) clusters. Specifically, the first part will prove that the input graphs can be divided into *at least* c clusters, and the second part will prove that the input graphs can be divided into *at most* c clusters. The remaining two parts are used to show that *if* there are c clusters *then* the sizes of the clusters are indeed equal to $(m_1,..,m_c)$. Specifically, the third part will be used to prove (tight) lower bounds on the number of clusters of specific minimum size, and the last part will be used to prove (tight) lower bounds on the number of clusters of specific maximum size.

The following four subsections formally describe these four parts, and prove some properties of each subprotocol in each part. Finally, we present a proof that these four subprotocols are enough to achieve our goal.

3.1 The input graphs fall to at least c clusters

We describe a subprotocol showing that there are at least c clusters in the m-tuple input $(G_1,\ldots,G_m)$. This is expressed by requiring that at least $c-1$ graphs are non-isomorphic to all their predecessors. Formally, for $i=2,\ldots,m$, define the statement U_i to hold if and only if for all $j<i$ the graphs G_i and G_j are not in the same cluster. Thus, U_i can be expressed as a Boolean formula over non-isomorphism statements; that is, we write each statement U_i as formula $f_i = (\wedge_{j<i}(G_i \not\approx G_j))$. Then, in order to prove that there are at least c clusters in $(G_1,\ldots,G_m)$, the prover proves in zero-knowledge that the formula

$$T_1 = \texttt{Thresh}_{c-1,m-1}(f_2,\ldots,f_m)$$

is true, which convinces the verifier that at least $c-1$ of the statements $U_2,\ldots,U_m$ are true. Adding-in G_1, we have proven that there exists at least c different isomorphism clusters.

Implementation. Clearly, each statement f_i is an AND over non-isomorphic statements, and thus a monotone formula over non-isomorphic statements. Then formula T_1 is a threshold gate over an AND of non-isomorphic statements, which is again a (polynomial-size) monotone formula over non-isomorphic statements, and can be proved in perfect zero-knowledge using Theorem 2.

Proof of correctness. We have the following proposition.

Proposition 1 *The formula T_1 is true if and only if the m-tuple input contains at least c clusters.*

Proof: Consider a generic cluster C and all formulae f_i associated to each graph G_i in C. Assume $G_1 \in C$; then all such formulae f_i are false. Instead, if $G_1 \notin C$, then there exists exactly one formula f_j which is true, where j is the smallest index such that $G_j \in C$. Therefore, the number of true formulae f_i is increased by 1 for all clusters but the one containing G_1, from which it follows that formula T_1 is true if and only if there are at least c clusters in the m-tuple input. ■

3.2 The input graphs fall to at most c clusters

We describe a subprotocol showing that there are at most c clusters in the m-tuple input $(G_1, \ldots, G_m)$. This is expressed by requiring that at most $(n-1)-(c-1)$ graphs are isomorphic to some predecessor. For $i = 2, \ldots, m$, define statement U_i and formula f_i as before. Then, in order to prove that there are at least c clusters in $(G_1, \ldots, G_m)$, the prover proves in zero-knowledge that the formula

$$T_2 = \texttt{Thresh}_{m-c,m-1}(\overline{f}_2, \ldots, \overline{f}_m)$$

is true, where formula $\overline{f}_i = \neg f_i = \vee_{j<i}(G_i \approx G_j)$. This convinces the verifier that at most $c-1$ of the statements $U_2, \ldots, U_m$ are true.

Implementation. Formula T_2 is a threshold gate over an OR of isomorphism statements, which is a (polynomial-size) monotone formula over isomorphism statements, and can be proved in perfect zero-knowledge using Theorem 1.

Proposition 2 *The formula T_2 is true if and only if the m-tuple input contains at most c clusters.*

Proof: Consider a generic non-empty cluster C containing z graphs, and all formulae $\overline{f}_i$ associated to each graph G_i in C. We see that the number of true such formulae $\overline{f}_i$ is exactly $z-1$, the only false one being $\overline{f}_j$, where j is the smallest index such that $G_j \in C$. Therefore, the number of true formulae $\overline{f}_i$ is equal to n minus the number of clusters, and formula T_2 is true if and only if n minus the number of clusters is $\geq n-c$, that is, if and only if the number of clusters is at most c. ■

3.3 Lower bounds on the number of clusters of specific minimum size

We describe a subprotocol for proving lower bounds on the number of clusters having a given minimum size. Suppose that the input integer sequence, $m_1, ..., m_c$, is such that for every $i = 1, ..., d$ exactly n_i integers equal $s_i > s_{i-1}$ (i.e., $n_i = |\{j : m_j = s_i\}|$), and $c = \sum_{i=1}^{d} n_i$ (with $s_0 = 0$). The prover proves in zero-knowledge the following statements:

$U_{3.1}$ = 'at least n_d clusters have size at least s_d';
$U_{3.2}$ = 'at least $n_d + n_{d-1}$ clusters have size at least s_{d-1}';
...

$U_{3.d}$ = 'all ($\sum_j n_j$) clusters have size at least s_1';
where a typical statement, *at least x clusters have size at least y*, is proven by saying that there all but at most x of the graphs are either isomorphic to some predecessor or isomorphic to at least y graphs (which wlog are not predecessors). (The excluded graphs are the ones first in each smaller cluster.) Thus, to prove a statement of the form *at least x clusters have size at least y*, the prover proves in zero-knowledge that the formula

$$T_3 = \mathtt{Thresh}_{m-c+x,m}(g_1, ..., g_m),$$

where $g_i = (\vee_{j<i}(G_i \approx G_j)) \vee (\mathtt{Thresh}_{y,m}((G_i \approx G_1), ..., (G_i \approx G_m)))$, is true. This subprotocol assumes that the input graphs are divided into exactly c clusters, which in turn can be proved as explained in the previous two subsections.

Implementation. Clearly, each formula g_i is a monotone formula over isomorphism statements. Then formula T_3 is a threshold gate over formulae g_i and is again a (polynomial-size) monotone formula over isomorphism statements, and can be proved in perfect zero-knowledge using Theorem 1.

Proposition 3 *The formula T_3 is true if and only if the m-tuple input contains at least x clusters having size at least y.*

Proof: Consider a generic non-empty cluster of the input graphs, and let z be its size; then there are z formulae g_i, each associated to a distinct graph in this cluster. We see that if $z \geq y$ these such z formulae are true (by virtue of the threshold part of the g_i). On the other hand, if $z < y$ exactly $z - 1$ out of these formulae are true (i.e., all but the first such g_i hold by the OR-part). Therefore, the total number of true formulae g_i can be written as

$$\begin{aligned}\sum_{j:s_j \geq y} n_j \cdot s_j + \sum_{j:s_j<y} n_j \cdot (s_j - 1) &= \sum_j n_j \cdot s_j - \sum_{j:s_j<y} n_j \\ &= m - \left(c - \sum_{j:s_j \geq y} n_j\right)\end{aligned}$$

Recalling that $T_3 = \mathtt{Thresh}_{m-(c-x),m}(g_1, ..., g_m)$, the proposition follows. ■

3.4 Lower bounds on the number of clusters of specific maximum size

We describe a subprotocol for proving lower bounds on the number of clusters having a given maximum size. Again, this subprotocol assumes that the input graphs are divided into exactly c clusters (and refers to n_i's and s_i's as above). Formally, we define the following statements:
$U_{4.1}$ = 'at least n_1 clusters have size at most s_1';
$U_{4.2}$ = 'at least $n_1 + n_2$ clusters have size at most s_2';
...
$U_{4.d}$ = 'all ($\sum_j n_j$) clusters have size at most s_d';
where a typical statement, *at least x clusters have size at most y*, is proven by saying that there are at least x graphs which are both non-isomorphic to all predecessors and non-isomorphic to at least $m - y$ graphs (which wlog are not predecessors). (The included graphs are the ones first in each small-enough cluster.) Specifically, to prove a statement of the form *at least x clusters have size at most y*, the prover proves in zero-knowledge that the formula

$$T_4 = \mathtt{Thresh}_{x,m}(\overline{g}_1, ..., \overline{g}_m),$$

where here $\overline{g}_i = (\wedge_{j<i}(G_i \not\approx G_j)) \wedge (\mathtt{Thresh}_{m-y,m}((G_i \not\approx G_1), ..., (G_i \not\approx G_m)))$, is true.

Implementation. Clearly, each formula $\overline{g}_i$ can be written as a monotone formula over non-isomorphism statements. Then formula T_4 is a threshold gate over formulae $\overline{g}_i$ and is a (polynomial-size) monotone formula over non-isomorphism statements, and can be proved in perfect zero-knowledge using Theorem 2.

Proposition 4 *The formula T_4 is true if and only if the m-tuple input contains at least x clusters having size at most y.*

Proof: Similar to the proof of Proposition 3. ■

3.5 The resulting protocol and its analysis

Let us denote by (P,V) the protocol constituted of the sequential composition of the subprotocols described in Sections 3.1,

Completeness and soundness of (P,V). In order to prove the completeness and soundness properties of (P,V), it is enough to show that the input $(G_1, \dots, G_m; m_1, \dots, m_c)$ belongs to language GCP if and only if the statements proved in the four subprotocols of (P,V) are true. Recall that the subprotocols in Section 3.1 and Section 3.2 convince the verifier that there are exactly c clusters among the input graphs $G_1, \dots, G_m$. Now, we consider the subprotocol in Section 3.3 and Section 3.4.

Proposition 5 *Suppose that the input integer sequence, $m_1, ..., m_c$, is such that for every $i = 1, ..., d$ exactly n_i integers equal s_i (i.e., $n_i = |\{j : m_j = s_i\}|$), and $c = \sum_{i=1}^{d} n_i$. Furthermore, suppose that exactly c of the statements U_i's hold. Then, statements $U_{3.1}, \dots, U_{3.d}$ and $U_{4.1}, \dots, U_{4.d}$ all hold if and only if the input is in* GCP.

Proof: One may easily verify that for input in GCP all of the statements $U_{3.1}, \dots, U_{3.d}$ and $U_{4.1}, \dots, U_{4.d}$ hold. We thus concentrate on proving the converse.

Let N_t denote the number of (isomorphism) clusters of size t. We start by observing that, for every $j = 1, \dots, d$, the statement $U_{3.j}$ can be written as

$$\sum_{t \geq s_{d-j+1}} N_t \geq n_d + \cdots + n_{d-j+1} \tag{1}$$

where the l.h.s represents the actual number of clusters having size at least s_{d-j+1} and the r.h.s the corresponding lower bound. Analogously, each statement $U_{4.j}$ can be written as

$$\sum_{t \leq s_j} N_t \geq n_1 + \cdots + n_j \tag{2}$$

Our aim is to establish, for every $j = 1, \dots, d$, the equality $N_{s_j} = n_j$. Once this is done, the proposition follows.

Recall that by the proposition hypothesis $\sum_{t \geq 1} N_i = c$ and so $\sum_{t \geq 1} N_t = \sum_{i=1}^{d} n_i = c$. Proceeding by induction on i, we show that $\sum_{t \leq s_i} N_t = n_1 + \cdots + n_i$ (and so $N_{s_i} = n_i$).

Base case ($i = 0$): Using Eq. (1) for $j = d$, we have $\sum_{t \geq s_1} N_t \geq n_d + \cdots + n_1 = c$, and so that $\sum_{t < s_1} N_t = 0$ as required.

Inductive step (from i to $i+1$): Suppose that $\sum_{t\le s_i} N_t = n_1+\cdots+n_i$, and consider $\sum_{t\le s_{i+1}} N_t$. Using Eq. (1) with $j = d-i$, we have

$$\begin{aligned}\sum_{t\ge s_{i+1}} N_t &\ge n_d+\cdots+n_{i+1}\\ &= c-\sum_{t\le s_i} N_t\end{aligned}$$

and thus, $\sum_{t\le s_{i+1}} N_t = \sum_{t\le s_i} N_t + N_{s_{i+1}}$. Using Eq. (2) with $j = i+1$, we have

$$\begin{aligned}\sum_{t\le s_{i+1}} N_t &\ge n_1+\cdots+n_{i+1}\\ &= \sum_{t\le s_i} N_t + n_{i+1}\end{aligned}$$

Thus, we have $N_{s_{i+1}} = n_{i+1}$, and so $\sum_{t\le s_{i+1}} N_t = n_1+\cdots+n_i+n_{i+1}$. ■

Perfect zero-knowledge of (P,V). First of all we notice that protocol (P,V) is a sequential composition of many perfect zero-knowledge proof systems, proving formulae T_1, T_2, T_3, T_4. By the above characterization, for each input $(G_1,\ldots,G_m; m_1,\ldots,m_c) \in$ GCP, all formulae T_1, T_2, T_3, T_4 are true, and therefore the protocols executed to prove them are also perfect zero-knowledge over the language GCP (and not only over the specific language proven). Then, the perfect zero-knowledge property of protocol (P,V) follows from the fact that a sequential composition of perfect zero-knowledge protocols results in a perfect zero-knowledge protocol, as proved in [GoOr].

We thus obtain our main result

Theorem 3 *The protocol (P, V) is a perfect zero-knowledge interactive proof system for GCP.*

References

[AKS] M. Ajtai, J. Komlos, and E. Szemeredi, *An $O(n\log n)$ Sorting Network*, Proceedings of 15th Annual ACM Symposium on Theory of Computing, 1983.

[AiHa] W. Aiello and J. Håstad. *Statistical Zero Knowledge Can Be Recognized in Two Rounds Journal of Computer and System Sciences*, vol. 42, 1991, pp. 327–345.

[BoFrLu] J. Boyar, K. Friedl, and C. Lund, *Practical Zero-Knowledge Proofs: Giving Hints and Using Deficiencies*, *Journal of Cryptology*, n. 4, pp. 185–206, 1991.

[DeDiPe] A. De Santis, G. Di Crescenzo, and G. Persiano, *The Non-Parameterized Graph Clustering Problem is in PZK*, manuscript, 1997.

[DeDiPeYu] A. De Santis, G. Di Crescenzo, G. Persiano, and M. Yung, *On Monotone Formula Closure of SZK*, Proceedings of 35th IEEE Symposium on Foundations of Computer Science, 1994.

[Fo] L. Fortnow, *The Complexity of Perfect Zero Knowledge*, Proceedings of 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 204–209.

[Fr] J. Friedman, *Constructing $O(n \log n)$ Size Monotone Formulae for the k-th Threshold Function of n Boolean Variables*, *SIAM Journal on Computing*, vol. 15, n. 3, 1986.

[G95] O. Goldreich, *Foundation of Cryptography – Fragments of a Book*, February 1995. Available from `http://theory.lcs.mit.edu/~oded/frag.html`.

[G96] O. Goldreich, *The Graph Clustering Problem has a Perfect Zero-Knowledge Proof,* Theory of Crypto Library, 1996.

[GoKu] O. Goldreich and E. Kushilevitz, *A Perfect Zero Knowledge Proof for a Decision Problem Equivalent to Discrete Logarithm,* "Advances in Cryptology – CRYPTO 88", vol. 403 of "Lecture Notes in Computer Science", S. Goldwasser editor, Springer Verlag.

[GoMiWi] O. Goldreich, S. Micali, and A. Wigderson, *Proofs that Yield Nothing but their Validity or All Languages in NP Have Zero-Knowledge Proof Systems*, *Journal of the ACM*, vol. 38, n. 1, 1991, pp. 691–729.

[GoOr] O. Goldreich and Y. Oren, *Definitions and Properties of Zero-Knowledge Proof Systems, Journal of Cryptology*, vol. 7, 1994, pp. 1–32.

[GoPe] O. Goldreich and E. Petrank, *Quantifying Knowledge Complexity,* Proceedings of 32nd IEEE Symposium on Foundations of Computer Science, 1994.

[GoMiRa] S. Goldwasser, S. Micali, and C. Rackoff, *The Knowledge Complexity of Interactive Proof Systems*, *SIAM Journal on Computing*, vol. 18, n. 1, 1989.

[ToWo] M. Tompa and H. Woll, *Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information*, Proceedings of 28th IEEE Symposium on Foundations of Computer Science, 1987.

[Va] L. Valiant, *Short Monotone Formulae for the Majority Function*, *Journal of Algorithms*, vol. 5, n. 3, 1984, pp. 363–366.