Efficient Zero-Knowledge Proofs for Some Practical Graph Problems

Yvo Desmedt[1,2] and Yongge Wang[3]

[1] Computer Science, Florida State University, Tallahassee
Florida FL 32306-4530, USA, desmedt@cs.fsu.edu
[2] Dept. of Mathematics, Royal Holloway, University of London, UK
[3] Department of Software and Information Systems, University of North Carolina at Charlotte, 9201 University City Blvd, Charlotte, NC 28223, ywang@uncc.edu

Abstract. From a cryptographic aspect zero-knowledge protocols for graph isomorphisms, graph non-isomorphisms, and graph-coloring are artificial problems, that received lots of attention. Due to recent work in network security in a broadcast setting, it seems important to design efficient zero-knowledge protocols for the following graph problems: independent set problem, neighbor independent set problem, and disjoint broadcast lines problem. In this paper, we will introduce a new concept of k-independent set problem which is a generalization of independent set and neighbor independent set problems, and we will present efficient zero-knowledge protocols for these problems. In the end of the paper we will give some cryptographic applications of k-independent set. Especially, we will point out the applications to the concept of "threshold" and appropriate access structures. Note that k-independent set also has applications outside cryptography, such as biology, methodology of scientific research, ethics, etc., which are beyond the scope of this paper.

Key words. Zero-knowledge, graph theory, secret sharing, key-escrow, complexity.

1 Introduction

The notion of zero-knowledge proof systems was introduced in the seminal paper of Goldwasser, Micali, and Rackoff [13]. Since their introduction, zero-knowledge proofs have proven to be very useful as a building block in the construction of cryptographic protocols, especially after Goldreich, Micali, and Wigderson [11] showed that all languages in **NP** have zero-knowledge proofs assuming the existence of secure encryption functions. The zero-knowledge protocols [11] for general **NP** problems are extremely complicated. Due to their importance, the efficiency of zero-knowledge protocols has received considerable attention. Several efficient zero-knowledge protocols for some graph problems, such as graph isomorphisms, graph non-isomorphisms, and graph-coloring, have been introduced in the literature. However, from a cryptographic aspect these graph problems are very artificial.

Goldreich, Goldwasser, and Linial [10], Franklin and Yung [8], and Franklin and Wright [7] have initiated the study of secure communication and secure computation in *multi-recipient (broadcast)* models. A "broadcast channel" (such as ethernet) enables one participant to send the same message—simultaneously and privately—to a fixed subset of participants. Franklin and Yung [8], Franklin and Wright [7], and Wang and Desmedt [3, 18] have used hypergraph theory to build efficient, reliable, and private protocols for broadcast communications. Most of their results are related to the interplay of network connectivity and secure communications. Their results also show that in a communication network, if we use broadcast channels, then we may achieve better results for reliability and privacy. For example, Dolev [4] and Dolev et al. [5] showed that, in the case of k Byzantine faults, reliable communication is achievable only if the systems's network is $2k + 1$ connected. Hadzilacos [15] has shown that even in the absence of malicious failures connectivity $k + 1$ is required to achieve reliable communication in the presence of k faulty participants. Franklin and Wright [7] and Wang and Desmedt [18] have shown that if broadcast channels are used, then in the case of k Byzantine faults, probabilistic reliable communication is achievable even if the system's network is only strongly $k+1$ connected. These recent research in broadcast channels proposes the challenge for a complete study of the following problems: when constructing a (hyper) graph, one may want to prove (without revealing the trapdoor) that the graph has certain properties. For example, what is the size of the maximum independent set? what is the size of the maximum neighborhood independent set? and so on.

In this paper we will design efficient zero-knowledge protocols for the k-independent set problem and for the strongly connectivity problem. These problems are important when designing protocols for broadcast channels. The main reason why we need neighbor independent set and strongly connectivity in broadcast channels is as follows: a vertex broadcast channel means that if a vertex broadcast some message, then all his neighbors (or the receivers who share the same frequency) will get this message with privacy and with authenticity. Privacy means that vertices who are not neighbors of the sender will learn nothing about the message. Authenticity means that if a receiver gets some message, he knows who has broadcast that message. It is clear that if two paths between two vertices are neighborhood disjoint then in order for the adversary to intercept the message on the two lines, one vertex is not sufficient and he has to control two vertices. However, if the two paths have a common neighbor v, then the adversary could control the vertex v and hear all things communicated on the two paths.

Sometime efficient zero-knowledge interactive proof systems are needed for these problems. For example, when one designs a network and a protocol over it for a broadcast communication, then for some reason the designer may decline to let all customers know all neighborhood disjoint paths between two vertices. But he has to prove to customers that the protocol can really work and is secure. Our efficient zero-knowledge interactive proof systems suffice for him to achieve this goal.

In the end of the paper we will give some cryptographic applications of k-independent set. Especially, we will point out the applications to the concept of "threshold" and appropriate access structures. Note that k-independent set also has applications outside cryptography, such as biology, methodology of scientific research, ethics, etc., which are beyond the scope of this paper.

2 Notations and preliminary results

In this section we give basic notations and recall the notions of interactive proof systems, zero-knowledge proof systems in the two-party model. We also give the definitions of several problems in graph theory related to broadcast channels.

Interactive protocols. Following [13], an *interactive Turing machine* is a Turing machine with a public input tape, a public communication tape, a private random tape and a private work tape. An *interactive protocol* is a pair of interactive Turing machines sharing their public input tape and communication tape. The *transcript* of an execution of an interactive protocol (P,V) is a sequence containing the random tape of V and all messages appearing on the communication tape of P and V.

Interactive proof systems. An interactive proof system for a language L is an interactive protocol in which, on an input string x, a computationally unbounded prover P convinces a polynomial-time bounded verifier V that x belongs to L. The requirements are two: completeness and soundness. Informally, completeness states that for any input $x \in L$, the prover convinces the verifier with very high probability. Soundness states that for any $x \notin L$ and any prover, the verifier is convinced with very small probability. A formal definition can be found in [13].

Zero-knowledge proof systems in the two-party model. A zero-knowledge proof system for a language L is an interactive proof system for L in which, for any $x \in L$, and any possibly malicious probabilistic polynomial-time verifier V$'$, no information is revealed to V$'$ that he could not compute alone before running the protocol. This is formalized by requiring, for each V$'$, the existence of an efficient simulator $\mathrm{S}_{\mathrm{V}'}$ which outputs a transcript "indistinguishable" from the view of V$'$ in the protocol. There exists three notions of zero-knowledge, according to the level of indistinguishability: computational, statistical, and perfect. The reader is referred to [13] for the definitions of computational, statistical, and perfect zero-knowledge proof systems. In this paper, we will only deal with computational zero-knowledge proof systems.

Let $G(V,E)$ be a graph and $V = \{v_1, \ldots, v_n\}$. The adjacent matrix $(a_{i,j})_{n\times n}$ of G is defined as follows: $a_{i,j} = 1$ if $(v_i, v_j) \in E$ and $a_{i,j} = 0$ otherwise. Note that it is always the case that the adjacent matrix of a graph is symmetric. The following properties of graphs play an important role in the design of broadcast protocols (see [8, 7, 18]).

An independent set in a graph $G(V,E)$ is a subset V' of V such that no two vertices in V' are joined by an edge in E. A vertex subset $V' \subseteq V$ of G is neighborhood independent if for any $u, v \in V'$ there is no $w \in V$ such that both (u, w) and (v, w) are edges in E.

Independent set problem (see, e.g., [9]): Given a graph $G(V,E)$ and an integer k, does there exist a size k independent set in G?

Neighborhood independent set problem (see, e.g., [18]): Given a graph $G(V,E)$ and an integer k, does there exist a size k neighborhood independent set in G?

The above two problems can be generalized to the case of k-independent set problem (which is a new concept we introduce in this paper). Let u and v be two vertices of a graph $G(V,E)$, and u-v_1-v_2-$\cdots$-v_{k-1}-v be the shortest path between u and v in G. Then we say that the *distance* between u and v is k. For two vertices which are not connected by any path, the *distance* between them is defined as ∞. A vertex subset $V' \subseteq V$ of G is k-independent if for any $u, v \in V'$, the distance between u and v in G is at least k. Note that V' is independent if and only if it is 2-independent, and V' is neighborhood independent if and only if it is 3-independent.

k-Independent set problem: Given a graph $G(V,E)$ and an integer k', does there exist a size k' k-independent set in G?

Let A and B be two vertices in a graph $G(V,E)$. We say that A and B are strongly k-connected if there are k neighborhood disjoint paths $p_1, \ldots, p_k$ between A and B, that is, for any $i \neq j (\leq k)$, p_i and p_j have no common neighbor (except A and B). In other words, for any vertex $v \in V \setminus \{A, B\}$, if there is a vertex u_1 on p_i such that $(v, u_1) \in E$, then there is no u_2 on p_j such that $(v, u_2) \in E$.

Strong connectivity problem (see, e.g., [18]): Given an integer k, a graph $G(V,E)$, and two vertices A and B, are A and B strongly k-connected?

These problems are extensively used in the design of efficient, reliable, and private protocols in broadcast channels (see, e.g., [7, 8, 18]).

The following theorem shows that all of them are **NP**-complete. We first give a definition.

Definition 1. *A vertex v in a graph $G(V,E)$ is* isolated *if there is no edge adjacent to v, i.e., for all $w \in V$, $(v, w) \notin E$.*

Theorem 1. *1. For each $k > 1$, the k-independent set problem is* **NP***-complete.*
2. The strong connectivity problem is **NP***-complete.*

Proof. It is straightforward that all of these four problems are in **NP**. In the following we show that all of them are **NP**-hard.

1. For the case $k = 2$, this is the independent set problem. Whence it suffices to reduce the independent set problem to k-independent problem for $k \geq 3$. The input $G(V,E)$, to independent set problem, consists of a set of vertices $V = \{v_1, \ldots, v_n\}$ and a set of edges E. In the following we construct a graph $f(G) = GKI(V_G, E_G)$ such that there is an independent set of size k' in G if and only if there is a k-independent set of size k' in GKI.

Without loss of generality, we may assume that G has no isolated vertices, otherwise we can consider them separately. Let $V_G = V \cup V'$ where

$$V' = \{v_{i,j,t} : (v_i, v_j) \in E, i < j; t = 1, \ldots, k-2\} \cup \{u_1, \ldots, u_{k-2}\}$$

and

$$\begin{aligned} E_G = \{&(v_i, v_{i,j,1}), (v_{i,j,1}, v_{i,j,2}), \ldots, (v_{i,j,k-2}, v_j) : i < j, (v_i, v_j) \in E\} \\ &\cup \{(u_1, u_2), \ldots, (u_{k-3}, u_{k-2}), (u_{k-2}, u_1)\} \\ &\cup \{(v_{i,j,t}, u_t) : (v_i, v_j) \in E; t = 1, \ldots, k-2\}. \end{aligned}$$

By a simple (straightforward) analysis, it can be verified that, for any k-independent set $V_1 \subseteq V_G$, if $V_1 \cap V' \neq \emptyset$ then $|V_1| = 1$. It is also clear that for any two vertex $u, v \in V$, the distance between u and v is at least k in GKI if and only if $(u, v) \notin E$. Hence there is a k-independent set of size k' in GKI if and only if there is an independent set of size k' in G.
2. We reduce the neighborhood independent set problem to strong connectivity problem. The input $G(V, E)$, to neighborhood independent set problem, consists of a set of vertices $V = \{v_1, \ldots, v_n\}$ and a set of edges E. Let A and B be two vertices and $f(G) = GSC(V_G, E_G)$ be a graph defined as follows: $V_G = \{A, B\} \cup V$, and $E_G = E \cup \{(A, v), (v, B) : v \in V\}$. It is clear that two paths $P_1 = (A, v_i, B)$ and $P_2 = (A, v_j, B)$ are vertex disjoint and have no common neighbor (except A and B) in GSC if and only if v_i and v_j have no common neighbor in $G(V, E)$. Whence there is a size k neighborhood independent set in G if and only if GSC is strongly k-connected. This completes the proof of the theorem. Q.E.D.
In the next two sections, we will give efficient computational zero-knowledge interactive protocols for these problems. Since these problems are **NP**-complete, by the results of Fortnow [6], these problems cannot have perfect zero-knowledge interactive protocols unless the polynomial hierarchy collapses.
Some notations: Let C be a set. $sym(C)$ denotes the set of permutations over C. When writing $x \in_R C$, we mean an element chosen at random with uniform distribution from the set C. A bit-commitment function f is an encryption scheme secure as in [12] which is a probabilistic polynomial time algorithm that on an input x and a random string $r \in \{0, 1\}^n$, outputs an encryption $f(x, r)$. Decryption is unique, that is $f(x, r) = f(y, s)$ implies $x = y$.

3 Efficient interactive zero-knowledge protocol for the *k*-independent set problem

In the following protocol, the common inputs are a graph $G(V, E)$ with $|V| = n$ and an integer k. The prover tries to convince the verifier that there is a size k independent set of G. Let $V' \subseteq V$ be an independent set of size k.

Protocol 1 The following four steps are executed n times, each time using independent coin tosses.

1. The prover P chooses a secret random permutation $\pi \in_R sym(\{1, \ldots, n\})$ and commits to each entry in the adjacency matrix in which the rows and columns are permuted according to π. More specifically, for each entry a_{ij} of the matrix (after permutation), the prover chooses a random r_{ij}, and send the new matrix $(f(a_{ij}, r_{ij}))_{n \times n}$ to the verifier.
2. The verifier V chooses a random bit $q \in_R \{0, 1\}$ and sends it to the prover.
3. If $q = 0$, then P reveals π and opens all commitments, else P opens $k * (k-1)/2$ commitments corresponding to the entries $(v_{\pi(i)}, v_{\pi(j)})$ in the permuted adjacency matrix, where $v_i, v_j \in V'$.

4. V checks whether the proofs provided by the prover in step (3) are correct. That is,
   - when $q = 0$. V checks that the commitments are correct and the committed matrix is obtained from the adjacency matrix according to π,
   - when $q = 1$. V checks that all the $k * (k - 1)/2$ commitments correspond to 0.

   If any proof is incorrect, the verifier *rejects* and stops. Otherwise the verifier continues to the next iteration.

If the verifier has completed all these iterations then it *accepts*.

Theorem 2. *Protocol 1 is a zero-knowledge interactive proof system for the independent set problem assuming that the commitment function f is secure.*

Sketch of proof. It is straightforward that the protocol is complete and sound. That is, when there is a k size independent set in G and both prover and verifier follow the protocol then the verifier always accepts. When the graph does not have a size k independent set and the verifier follows the protocol then no matter how the prover plays, the verifier will reject with probability at least $1 - 2^{-n}$. Thus the above protocol constitutes an interactive proof system for the independent set problem. We need to show that the above protocol is zero-knowledge. It is clear that the above prover conveys no knowledge to the specified verifier. We need however to show that our prover conveys no knowledge to all possible verifiers, including ones that deviate arbitrarily from the protocol. Let V$'$ be an arbitrary fixed probabilistic polynomial-time Turing machine interacting with the prover P. We will present a probabilistic polynomial-time simulator $\mathrm{S}_{\mathrm{V}'}$ that generates a probability distribution which is polynomially indistinguishable from the probability distribution induced on V$'$'s tapes during its interaction with the prover P. It suffices to generate the distribution on the random tape and the communication tape. In the following we use V$'$ to construct the simulator $\mathrm{S}_{\mathrm{V}'}$. The machine $\mathrm{S}_{\mathrm{V}'}$ monitors the execution of V$'$. In particular, $\mathrm{S}_{\mathrm{V}'}$ chooses the random tape of V$'$, reads message from V$'$'s communication tape, and writes message to V$'$'s communication tape. Typically, $\mathrm{S}_{\mathrm{V}'}$ tries to guess which question V$'$ will ask to check. $\mathrm{S}_{\mathrm{V}'}$ will encrypt an illegal adjacency matrix such that it can answer V$'$ in case if ($\mathrm{S}_{\mathrm{V}'}$) is lucky. The case in which $\mathrm{S}_{\mathrm{V}'}$ fails will be ignored: $\mathrm{S}_{\mathrm{V}'}$ will just rewind V$'$ to the last success, and try its luck again. It is crucial that from the point of view of V$'$ the case which leads to $\mathrm{S}_{\mathrm{V}'}$ success and the case which leads to $\mathrm{S}_{\mathrm{V}'}$ failure are polynomially indistinguishable. The details are standard as those simulators by Goldreich, Micali, and Wigderson [11] for the zero-knowledge proof systems for 3-colorability. And we will omit the details. Q.E.D.

Next we consider zero-knowledge interactive proof systems for the k-independent set problem. For an integer $k \geq 3$ and a graph $G(V, E)$, we first construct another graph $G_k(V_G, E_G)$ as follows. Let $V_G = V$ and

$$E_G = \{(u, v) : \text{ the distance between } u \text{ and } v \text{ is at most } k - 1\}.$$

Obviously, the graph G_k can be constructed from G efficiently.

Lemma 1. *Given a graph $G(V, E)$ and an integer $k \geq 3$, let $G_k(V_G, E_G)$ be defined as above. Then a vertex set $V' \subset V$ is k-independent in G if and only if it is independent in G_k.*

Proof. Straightforward. Q.E.D.

By Lemma 1, the zero-knowledge interactive proof system for independent set of G_k is trivially a zero-knowledge interactive proof system for k-independent set of G.

4 Efficient interactive zero-knowledge protocol for the strong connectivity problem

Given a graph $G(V, E)$ with $V = \{v_1, \ldots, v_n\}$, a direct flow function on G is a function $F : \overline{E} \to \{0, 1\}$ where $\overline{E} = \{u \to v, v \to u : (u, v) \in E\}$, that is, $\overline{E}$ is the bi-directionlized version of E. For any $(u, v) \in E$ with $F(u \to v) = 1$ we say that there is a flow from u to v (but not a flow from v to u). Alternatively, a flow F on a graph G can be described by the matrix $(F_{ij})_{n\times n}$ where $F_{ij} = 1$ if $F(v_i \to v_j) = 1$ and $F_{ij} = 0$ otherwise. And we call this matrix the *flow matrix*. Note that a flow matrix is not symmetric (cf. the adjacency matrix).

Assume that $A = v_1$, $B = v_n$ and let $P = \{p_1, \ldots, p_k\}$ be k neighborhood disjoint paths between A and B satisfying the following the following minimal property:

- For each path $p_i = (A, v_{i,1}, \ldots, v_{i,m}, B)$, if $(v_{i,j_1}, v_{i,j_2}) \in E$, then $j_2 \leq j_1 + 2$.

A flow F_P on G is then defined as follows: for each path $p_i = Av_{i_1}v_{i_2}\ldots v_{i_t}B$, let

$$F_P(A \to v_{i_1}) = F_P(v_{i_1} \to v_{i_2}) = \ldots = F_P(v_{i_t} \to B) = 1$$

and

$$F_P(v_{i_1} \to A) = F_P(v_{i_2} \to v_{i_1}) = \ldots = F_P(B \to v_{i_t}) = 0.$$

For any edge $(u, v) \in E$ which is not on any of the paths, let $F_P(u \to v) = F_P(v \to u) = 0$. By the neighborhood independence of P, the flow F_P is well defined.

In the following protocol, the common inputs are a graph $G(V, E)$ with $|V| = n$ and an integer k. The prover tries to convince the verifier that there are k neighborhood disjoint paths between A and B. Let $P = \{p_1, \ldots, p_k\}$ be k neighborhood disjoint paths between A and B and F_P be the corresponding flow function.

Protocol 2 The following three steps are executed for sufficiently many times, each time using independent coin tosses.

1. The prover P chooses a secret random permutation $\pi \in_R sym(\{1, \ldots, n\})$ such that $\pi(1) = 1$ and $\pi(n) = n$. P commits to each entry in the adjacency matrix and each entry in the flow matrix in which the

rows and columns are permuted according to π. More specifically, let $(f_{ij})_{n\times n}$ and $(a_{ij})_{n\times n}$ be the flow matrix and the adjacency matrix (after the permutation π) respectively. For each a_{ij} and f_{ij} the prover chooses random r_{ij}, r'_{ij} and computes $a'_{ij} = f(a_{ij}, r_{ij}), f'_{ij} = f(f_{ij}, r'_{ij})$. P sends the new matrices $(a'_{ij})_{n\times n}$ and $(f'_{ij})_{n\times n}$ to the verifier.

2. The verifier V chooses a random question $q \in_R \{0, 1, 2, 3\}$ and sends it to the prover.
3. According to the value of q, we distinguish the following cases:
   (a) $q = 0$. P reveals π and opens all commitments corresponding to the adjacency matrix. V verifies that the commitments are correct and the committed adjacency matrix is obtained from the original adjacency matrix according to π.
   (b) $q = 1$. P reveals all entries in the first row, the last row, the first column, and the last column of the committed flow matrix. The verifier checks that in the committed flow matrix:
      - there are exactly k 1's in the first row,
      - there is no 1 in the last row,
      - there is no 1 in the first column, and
      - there are exactly k 1's in the last column.

   (c) $q = 2$. V chooses a random question $i \in_R \{2, \ldots, n-1\}$ and sends it to the prover. P opens the i-th row and the i-th column of the committed flow matrix. If there are 1's in these row and column, P also opens the corresponding edge entries in the committed adjacency matrix. For example, if the value of the entries f_{ij_1} and f_{j_2i} are 1's, then P also opens the entries a_{ij_1} and a_{j_2i} of the committed adjacency matrix. V verifies that either one of the following case is true:
      - Neither the i-th row nor the i-th column of the committed flow matrix contains an entry 1.
      - There is exactly one entry 1 in the i-th row and exactly one entry 1 in the i-th column of the committed flow matrix. If this is the case, then V also verifies that the corresponding edges exist (by checking that the entries from the adjacency matrix opened by P are 1's).

   (d) $q = 3$. V chooses a random question $(i, j, t) \in_R \{2, \ldots, n-1\}^3$ and sends it to the prover. P opens the i-th, j-th, t-th rows, the i-th, j-th, t-th columns of the committed flow matrix. P also opens the entries a_{ij}, a_{it}, and a_{jt} of the committed adjacency matrix.
      - If any entry of a_{ij}, a_{it}, and a_{jt} equals to 1, for example, assume that $a_{ij} = 1$, then V verifies that all flows passing through the i-th and the j-th vertices (after the permutation π) are consistent with the neighborhood disjoint property. That is, if there is any flow passing through the i-th vertex (after the permutation π) but not passing through the j-th vertex, then there is no flow through the j-th vertex at all. Similarly, if there is any flow passing through the j-th vertex (after the permutation π) but not passing through the i-th

vertex, then there is no flow through the i-th vertex at all. Specifically, the following conditions hold:

  - If $f_{ij} = 1$ then exactly one entry in the i-th column of the committed flow matrix equals to 1 and exactly one entry in the j-th row of the committed flow matrix equals to 1.
  - If $f_{ji} = 1$ then exactly one entry in the j-th column of the committed flow matrix equals to 1 and exactly one entry in the i-th row of the committed flow matrix equals to 1.
  - If $f_{ij} = 0$, $f_{ji} = 0$, and $f_{ij'} = 1$ for some $j' \neq j$, then $f_{jk} = 0$ for all $k \neq j'$ and $f_{kj} = 0$ for all $k \neq j'$.
  - If $f_{ij} = 0$, $f_{ji} = 0$, $f_{i'j} = 1$ for some $i' \neq i$, then $f_{ik} = 0$ for all $k \neq i'$ and $f_{ki} = 0$ for all $k \neq i'$.

- If at least two entries among a_{ij}, a_{it}, and a_{jt} equal to 1, then V verifies that the committed flow matrix is consistent with the neighborhood independent property (we will not give the details here which are similar to those in the previous item). For example, assume that $a_{ij} = a_{it} = 1$ and there is a flow passing through the i-th vertex (after the permutation π) but not passing through either the j-th or the t-th vertex, then there is no flow through the j-th or the t-th vertex at all.

If any of the above verification fails, then V *rejects* and stops the protocol. Otherwise, go to the next iteration.

If the verifier has completed all these iterations then it *accepts*.

Theorem 3. *Protocol 2 is an interactive proof system for the strong connectivity problem assuming that the commitment function f is secure.*

Sketch of proof. First we note the following fact: Step 3a in the protocol is used to check that the commitments are correct and that the submitted adjacency matrix is obtained from the original adjacency matrix according to the permutation π. Step 3b is used to check that there are k outgoing flows from A and k incoming flows to B, but there is neither incoming flow to A nor outgoing flow from B. Steps 3c and 3d are used to check that the flows are legal and the flows correspond to k neighborhood disjoint paths from A to B. Specifically, step 3c is used to check that for each vertex (excluding A and B) there is either no flow through it at all or there is exactly one incoming flow and one outgoing flow through it. Step 3d checks that the k paths from A to B corresponding to the flows are neighborhood disjoint. Now it is straightforward that the above protocol is complete and sound. Q.E.D.

Theorem 3 shows that Protocol 2 is sound and complete. However, this protocol is not zero-knowledge. Indeed, step 3d may leak the following information: some path in the set of neighborhood disjoint paths has length of at least 3 or 4. Let $P = \{p_1, \dots, p_k\}$ be a set of neighborhood disjoint paths between A and B. Then generally the malicious verifier V$'$ does not know the $l = \max_{1 \leq i \leq k} |p_i|$. If $l = 2$, then during the execution of step 3d in Protocol 2, V$'$ will never have the chance to see $f_{ij} = 1$

(or $f_{ji} = 1$) in his view. And if $l > 2$ then V$'$ will have the chance to see $f_{ij} = 1$ (or $f_{ji} = 1$) in his view. Whence, only from V$'$, we cannot construct a simulator to generate an indistinguishable view of V$'$ interacting with the prover. It follows that Protocol 2 is not zero-knowledge. This problem can be easily fixed by converting the graph G into a new graph G' first and then apply the protocol.
Let $G(V, E)$ be a graph with $V = \{v_1, \ldots, v_n\}$. Define a new graph $G'(V_G, E_G)$ by letting $V_G = V \cup \{v_{ij} : (v_i, v_j) \in E, i < j\}$ and $E_G = E \cup \{(v_i, v_{ij}), (v_{ij}, v_j) : i < j, v_{ij} \in V_G\}$. It is straightforward that there are k neighborhood disjoint paths between v_1 and v_n in G if and only there are k neighborhood disjoint paths between v_1 and v_n in G'. And there exists a set of k neighborhood disjoint paths $P = \{p_1, \ldots, p_k\}$ between v_1 and v_n in G' such that $\max_{1 \leq i \leq k} |p_i| \geq 4$. Now we can present a zero-knowledge proof system for the strong connectivity problem.
The common inputs to the following protocol is: a graph $G(V, E)$ with $V = \{v_1, \ldots, v_n\}$, two vertices $A = v_1, B = v_n$, and an integer k. The prover tries to convince the verifier that there are k neighborhood disjoint paths between A and B.

Protocol 3

1. P construct a graph G' from G as mentioned above. P computes k neighborhood disjoint paths $P = \{p_1, \ldots, p_k\}$ between A and B in G' such that $\max_{1 \leq i \leq k} |p_i| \geq 4$ and let F_P be the corresponding flow function.
2. P notifies V of the graph G' as the common inputs. V verifies that G' is got from G properly.
3. P and V execute the Protocol 2 on the common inputs: graph G', two vertices A, B, and the integer k.

Theorem 4. *Protocol 3 is a zero-knowledge interactive proof system for the strong connectivity problem assuming that the commitment function f is secure.*

Sketch of proof. By Theorem 3, Protocol 3 is complete and sound. The proof of the zero-knowledge property of the protocol follows from the discussion before Protocol 3 and is similar to the proof of Theorem 2. The details are omitted here. Q.E.D.

5 Applications of *k*-independent set

The concept of k-independent set challenges the concept of "threshold" and has applications to computing an appropriate access structure. Indeed, given a number of participants, a graph is used to identify relationships (e.g. potential conflicts of interest: boss, family member, etc.). A secret sharing scheme based on a threshold may be inappropriate. Although general access structures have been defined by Ito, Saito, and Nishizeki [16] and have been heavily studied (see, e.g., Blundo, De Santis, Stinson, and Vaccaro [1]), no scientific way has been suggested to construct such an access structure. We suggest that a set of participants

should only be authorized by the access structure if the participants forms a k-independent set (note that this kind of access structures are not monotone). Whence the access structure should satisfy the property of k-independence in the relationship graph for some integer k. Note that a k-independent set is not necessarily an access structure though.
Similarly, we may also conjecture that the k-independent set problem will have some applications in key-escrow protocols [17]. Note that at present, the number of trusted agencies in a key-escrow system is very limited (e.g., 2 or 3). However, in the future there may be possibility that the number of trusted agencies will increase considerably. Whence the problem of choosing trusted agencies for the key-escrow system need to be solved. If we choose several agencies who often collude, it will not help us to get privacy. They may collude to recover our private keys without the official approval. The k-independence provides us a good solution for this problem. That is, the trusted agencies for a key-escrow system should be chosen as being independent as possible.
We close our discussions with some digression. Indeed, the properties of k-independence of vertices have many other applications which may not be cryptographic ones. One may first note the close relationship between the distance of vertices and the Erdös numbers [14]. In each graph, a distinguished vertex p is the outstanding, prolific, and venerable Paul Erdös. The distance from a vertex u to p is known as u's *Erdös number.* Thus, for example, Paul Erdös's co-authors have Erdös number 1. Those people with finite Erdös number constitute the *Erdös component* of the graph. Whence if we extend the notion of Erdös number by allowing several unknown Paul Erdöses in a graph, then the problem of finding the Paul Erdöses in a graph is related to the problem of finding a k-independent set for some k. Note that this will be an important problem for classifying the literature (e.g., classifying the literature of cryptography) and for finding new disciplines. Also it could be noted that the problem of k-independence may find applications in the areas like: (1). Classifying animals by building the graph according to the gene similarity. (2). Reviewing a paper (or a proposal, etc.) by avoiding Byzantine humans that may conspire. Our above discussion shows that it makes sense that a formal study of ethics uses tools and definitions from cryptography, e.g. k-independent set.

References

1. C. Blundo, A. De Santis, D. Stinson, and U. Vaccaro. Graph decompositions and secret sharing schemes. In: *Advances in Cryptology, Proc. of Eurocrypt '92*, LNCS 658, pages 1–24, Springer Verlag, 1992.
2. A. De Santis, G. Di Crescenzo, O. Goldreich, G. Persiano. The Graph Clustering Problem has a Perfect Zero-Knowledge Interactive Proof. *Information Processing Letters* **69**(4): 201-206, 1999.
3. Y. Desmedt and Y. Wang. Approximation hardness and secure communication in broadcast channels. In: *Advances in Cryptology, Proc. Asiacrypt '99*, LNCS 1716, pages 247–257, Springer Verlag, 1999.

4. D. Dolev. The Byzantine generals strike again. *J. of Algorithms*, **3**:14–30, 1982.
5. D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *J. of the ACM*, **40**(1):17–47, 1993.
6. L. Fortnow. The complexity of perfect zero-knowledge. In: *Proc. ACM STOC '87*, pages 204–209, ACM Press, 1987.
7. M. Franklin and R. Wright. Secure communication in minimal connectivity models. *Journal of Cryptology*, **13**:9–30. 2000.
8. M. Franklin and M. Yung. Secure hypergraphs: privacy from partial broadcast. In: *Proc. ACM STOC '95*, pages 36–44, ACM Press, 1995.
9. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of* **NP**-*Completeness.* W. H. Freeman and Company, San Francisco, 1979.
10. O. Goldreich, S. Goldwasser, and N. Linial. Fault-tolerant computation in the full information model. *SIAM J. Comput.* **27**(2):506–544, 1998.
11. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in **NP** have zero-knowledge proof systems. *J. of the ACM*, **38**(1):691–729, 1991.
12. S. Goldwasser and S. Micali. Probabilistic encryption. *J. of Comp. and Sys. Sci.*, **28**(2):270–299,1984.
13. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comp.*, **18**(1):186–208, 1989.
14. J. Grossman and P. Ion. On a portion of the well-known collaboration graph. *Congressus Numerantium*, **108**:129–131, 1995.
15. V. Hadzilacos. *Issues of Fault Tolerance in Concurrent Computations.* PhD thesis, Harvard University, Cambridge, MA, 1984.
16. M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structures. In: *Proc. IEEE Global Telecommunications Conf., Globecom '87*, pages 99–102, IEEE Communications Soc. Press.
17. S. Micali. Fair public-key cryptosystem. In: *Advances in Cryptology, Proc. of Crypto '92*, LNCS 740, pages 113–138, Springer Verlag, 1992.
18. Y. Wang and Y. Desmedt. Secure communication in multicast channels. *J. of Cryptology* **14**:121–135, 2001.