Computational Complexity of Reliability/Security in Colored Networks and Privacy Preserving Censorship

Yvo Desmedt[1,3⋆], Yongge Wang[2], and Mike Burmester[3⋆⋆]

[1] University College London, UK, y.desmedt@cs.ucl.ac.uk
[2] UNC Charlotte, USA, yonwang@uncc.edu
[3] Florida State University, USA, burmester@cs.fsu.edu

Abstract. Dolev-Dwork-Waarts-Yung linked research on reliable point-to-point networks with privacy and authenticity. In their threat model the adversary can only take over a number of nodes bounded by a threshold k. Hirt-Maurer introduced the concept of an adversary structure (i.e. the complement of an access structure). Kumar-Goundan-Srinathan-Rangan and Desmedt-Wang-Burmester generalized Dolev-Dwork-Waarts-Yung scenarios to the case of a general adversary structure.
Burmester-Desmedt introduced a special adversary structure, now called a color based adversary structure. Each platform in the network is given a color. The adversary can control all nodes that have up to k different colors.
Although the family of color based adversary structures has a trivial representation which size grows polynomial in the size of the graph, we will demonstrate in this paper that deciding reliability issues and security issues are co-**NP**-complete.
We apply this result to study censorship, which for centuries often has been viewed by authorities as an essential security tool. Authorities may require network designers to demonstrate the capability to censor the internet. We present a zero-knowledge interactive proof for the case of a color based adversary structure.
Keywords: network security, Byzantine threats, secret sharing, adversary structure, censorship, unconditional security, zero-knowledge

1 Introduction

Censorship has been used extensively during centuries. The recently recovered "Gospel of Judas" [16] has been used as an occasion to reflect back on how the church censored "non-traditional" gospels [17]. Today in many countries books remained censored. A well known example is Hitler's "Mein Kampf." Moreover texts describing in details the construction of atomic bombs, or other classified information, are also censored.

Whether censorship in a limited format is in the benefit of mankind or not, is a non-scientific topic, and therefore not discussed. Information, such as books, are passed on through a network, e.g. a distribution network, involving bookstores, etc. The communication of gossip can be modeled using social networks [19]. Whether the edges in this network are virtual or physical communication links seems irrelevant. However, as we now discuss, this conclusion may be wrong.

In the *classical model* for communication networks nodes are treated equally. So when a limited adversary (or a censor in our prior example) wants to undermine communication, it is natural to assume that there is an upperbound

⋆ A part of this work has been funded by CCR-0209092. The author is BT Professor of Information Security.
⋆⋆ A part of this work has been funded by CCR-0209092.

k of the number of nodes the adversary (or censor) can control. The first to dispute this homogeneous viewpoint was Hirt and Maurer [13]. Their paper introduces the concept of an adversary structure (i.e. the complement of an access structure [14]). An adversary structure is a list of subsets the adversary can control. Before performing the attack the adversary must choose one of these subsets. However, Hirt and Maurer do not specify how to choose such an adversary structure. Burmester-Desmedt [3] introduced a method to address this, we now discuss. Burmester-Desmedt partition the nodes in a network based on the platform used to operate the node, e.g. the router. The mapping from node to platform is modeled using a node coloring. To take into account the ease of automated attacks using computer viruses and worms, they view that the difficulty for an adversary to control one node running one platform is approximately the same as the difficulty to control all nodes running the same platform. A limited adversary corresponds in their setting to one that can control all nodes that have up to k different colors. The resulting adversary structure is called a color based adversary structure.

We believe that color based adversary structures are worth studying in more details for the following reasons:

1. it was revealed at the Blackhat 2005 conference that the operating system used on Cisco routers has serious vulnerabilities [21] (note the paper in the proceedings was pulled due to pressure by Cisco). So, a color based adversary structure corresponds to reality.
2. the family of color based adversary structures has a representation which size grows polynomial in the size of the graph. This is not the case for the general case of adversary structures, making them completely impractical to use on large graphs.

In this paper we will demonstrate that although the family of color based adversary structures has a short representation, the complexity of deciding whether a given colored graph allows to achieve reliability and/or privacy are co-**NP**-complete problems. So, the question which colors to shut down to censor such a priorly described colored network is **NP**-complete. As is well known, the equivalent problem for the classical model is in **P**.

When a point-to-point network is built the designer may be asked by the authorities whether it can be censored by controlling k platforms. This can be achieved by building trapdoors in these k platforms (for a discussion on this issue, see e.g. [18]). Evidently, it should be hard for an adversary to find these k colors. To answer this question, we present a zero-knowledge interactive proof.

The paper is organized as following. In Section 2 we survey what is known about security (privacy and authenticity) and reliability in point-to-point networks with a color based adversary structure. We also briefly survey the concept of zero-knowledge interactive proof. In Section 3 we prove the computational complexity. In Section 4 we give a zero-knowledge interactive proof for knowledge of up to k colors that will cut the colored graph. Finally we conclude with some remarks and open problems in Section 5.

2 Background

We survey the work on colored networks with a color based adversary structure. We also briefly discuss the concept of zero-knowledge interactive proof. We start by some definitions.

2.1 Definitions

Definition 1. [13] Let V be a finite non-empty set. An adversary structure $\mathcal{A}_V$ for V is a subset of the power set 2^V such that if $B \in \mathcal{A}_V$ then subsets of B are also in $\mathcal{A}_V$.

In our context, V will be vertices in a graph.

Definition 2. A vertex-*colored graph* is a tuple $G = G(V, E, C, f)$, with V the node set, E the edge set, C the color set, and f a map from V onto C. The structure

$$\mathcal{Z}_{C,k} = \{Z \mid Z \subset V \text{ and } |f(Z)| \leq k\}.$$

is called a k-*color* adversary structure. Let $A, B \in V$ be distinct nodes of G. A, B are called $(k+1)$-*color connected* for $k > 1$ if for any color set $C_k \subseteq C$ of size k, there is a path p from A to B in G such that the nodes on p does not contain any color in C_k.

It should be noted that color connectivity is unrelated to the issue of vertex disjoint paths. Indeed take the graph in Figure 1. A and 3-color connected, but not 4-color connected, as is easy to verify using an exhaustive search. However, the simple paths from A to B are not vertex disjoint. If one removes nodes to make them vertex disjoint, the resulting graph is no longer 3-color connected.

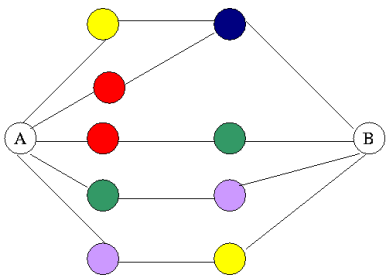

Fig. 1. A 2-color connected graph

Definition 3. Let $G(V, E)$ be a directed graph, A, B be nodes in $G(V, E)$, and $\mathcal{Z}$ be a an adversary structure on $V \setminus \{A, B\}$.

- A, B are $\mathcal{Z}$-*separable* in G, if there is a set $Z \in \mathcal{Z}$ such that all paths from A to B go through at least one node in Z. We say that Z separates A and B.
- A, B are $(\mathcal{Z}+1)$-*connected* if they are not $\mathcal{Z}$-separable in G.

Definition 4. [5] If $\mathcal{Z}_1$ and $\mathcal{Z}_2$ are adversary structures for $\mathcal{P}$, then $\mathcal{Z}_1 + \mathcal{Z}_2 = \{Z_1 \cup Z_2 : Z_1 \in \mathcal{Z}_1, Z_2 \in \mathcal{Z}_2\}$. $2\mathcal{Z}$ and $3\mathcal{Z}$ are the adversary structures $\mathcal{Z} + \mathcal{Z}$ and $\mathcal{Z} + \mathcal{Z} + \mathcal{Z}$ respectively.

Obviously, $\mathcal{Z}_1 + \mathcal{Z}_2$ is also an adversary structure for $\mathcal{P}$.

2.2 Survey of the known results

We now survey the state of the art on the research of security and reliability in point-to-point networks with a general adversary structure.

Theorem 5. *A necessary and sufficient condition for A and B to privately communicate in a point-to-point network in the presence of a Byzantine adversary, in the case all communication links (edges in the graph) are:*

two-way *is that A, B are $(2\mathcal{Z}+1)$-connected in G [15] 2002).*
one-way without feedback, *is that A, B are $(3\mathcal{Z}+1)$-connected in G [5].*

The following theorem is about 100% guaranteed reliability.

Theorem 6. *[5] Let $G = G(V, E)$ be a directed graph, A, B be nodes in G, and $\mathcal{Z}$ be an adversary structure on $V \setminus \{A, B\}$. We have $\mathcal{Z}$-reliable message transmission from A to B if, and only if, A, B are strongly $(2\mathcal{Z}+1)$-connected in G.*

Note that the issue of privacy without reliability will be addressed in the final paper of [5].

The above results for the case of color based adversary structure trivially become:

Corollary 7. *Let $G = G(V, E, C, f)$ be a vertex-colored graph and $A, B \in V$. A necessary and sufficient condition for A and B to privately communicate in a point-to-point network in the presence of a k-color adversary in the case all communication links (edges in the graph) are:*

two-way *is that A, B are $2k+1$-color connected in G*
one-way without feedback, *is that A, B are $3k+1$-color connected in G.*

Moreover, reliable message transmission from A to B with such an adversary is possible if, and only if, A, B are $2k+1$-color connected in G.

2.3 Zero-knowledge interactive proof

Interactive protocols. Following [12], 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 [12].
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 [12] 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.

3 Computational complexity

In this section we are interested in the computational complexity of deciding whether a given vertex-colored graph can achieve privacy and reliability against a k-color adversary structure. From Corollary 7 we know that the issue of $k+1$ (or $2k+1$, or $3k+1$)-color connectivity is essential.

So, from a computational problem it is sufficient to focus on the case of k-connectivity. We now prove that this problem is co-**NP**-complete. We focus on the complementary problem, which is trivial to see to correspond to the following. We call it the color separable problem. We first define, as a special case of Definition 3, the following.

Definition 8. Let $G = G(V, E, C, f)$ be a vertex-colored graph and A, B be nodes in $G(V, E)$. A, B are *k-color separable* in G, if there is a set $V' \subseteq V$ such that all paths from A to B go through at least one node in V' and $f(V') \leq k$. We say that V' is a k-color separator of A and B.

INSTANCE: A vertex-colored network $G = G(V, E, C, f)$, two nodes $A, B \in V$, and a positive integer $k \leq |C|$.
QUESTION: Are A and B k-color separable?

Theorem 9. *The color separable problem is* **NP***-complete.*

Proof. It is straightforward to show that the problem is in **NP**. Thus it is sufficient to show that it is **NP**-hard. The reduction is from the Vertex Cover (VC) problem. The VC problem is as follows (definition taken from [10]):

INSTANCE: A graph $G = (V, E)$ and a positive integer $k \leq |V|$.
QUESTION: Is there a vertex cover of size k or less for G, that is, a subset $V' \subseteq V$ such that $|V'| \leq k$ and, for each edge $(u, v) \in E$, at least one of u and v belongs to V'?

For a given instance $G = (V, E)$ of VC, we construct a vertex-colored network $G_c = (V_c, E_c, f, C)$ as follows. First assume that the vertex set V is ordered as in $V = \{v_1, \ldots, v_n\}$. Let

$$\begin{aligned} V_c &= \{A, B\} \cup \left\{e^1_{(v_i,v_j)}, e^2_{(v_i,v_j))} : (v_i, v_j) \in E \text{ and } i < j\right\} \\ E_c &= \left\{(A, e^1_{(v_i,v_j)}), (e^1_{(v_i,v_j)}, e^2_{(v_i,v_j)}), (e^2_{(v_i,v_j)}, B) : (v_i, v_j) \in E\right\} \\ C &= \left\{c_v : v \in V\right\} \\ f &= \left\{f(e^1_{(v_i,v_j)}) = c_{v_i}, f(e^2_{(v_i,v_j)}) = c_{v_j} : (v_i, v_j) \in E, i < j\right\} \end{aligned}$$

In the following, we show that there is a vertex cover of size k in G if and only if there is a k-color separator for G_c.

Without loss of generality, assume that $V' = \{v'_1, \ldots, v'_k\}$ is a vertex cover for G. Then it is straightforward to show that $C' = \{c_{v'_i} : v'_i \in V'\}$ is a color separator for G_c since each incoming path for B in G_c contains both colors of the corresponding edge's end-vertices.

For the other direction, assume that $C' = \{c_{v'_i} : i = 1, \ldots, k\}$ is a k-color separator for G_c. Let $V' = \{v'_i : c_{v'_i} \in C'\}$. By the fact that C' is a color separator for G_c, for each edge $(v_i, v_j) \in E$ in G, the path $(A, e^1_{(v_i,v_j)}, e^2_{(v_i,v_j)}, B)$ in G_c contains at least one color from C'. Since this path contains only two colors c_{v_i} and c_{v_j}, we know that v_i or v_j or both belong to V'. In another word, V' is a k-size vertex cover for G. This completes the proof of the Theorem.

4 Privacy preserving censorship

4.1 Introduction

As we discussed in the introduction, deciding whether one can censor a network using limited resources is straightforward under the classical network problem. However, it is no longer under the vertex-colored graph model. The problem is **NP**-complete.

When a network is designed, the authority may want to ask whether it is possibly to censor traffic in the network by only controlling nodes running on at most k platforms (colors). To allow the network designer to prove this censoring capability, the network designer will proof to the authority the existence of such k platforms (colors). To avoid an outsider to take control of the network the set of these k platforms (colors) should remain secret. Therefore we present a zero-knowledge interactive proof for above problem. Inspired by [7] we present a zero-knowledge interactive proof for above.

4.2 A difficulty

Many zero-knowledge proofs for NP-complete problems [1,11,2] conists of committing in a first stage. Then the verifier asks a binary question. The prover then either open all the commitments or reveals other information such that if both questions would had been asked, the secret would leak.

The problem of designing an efficient zero-knowledge proof seems rather trivial. Indeed, the prover could in the first step permute all the vertices, and permute all the colors and commit to these. The verifier then asks a question. If the question is 0, the prover opens all commitments, else reveals a set V' that separates A and B in this isomomorphic graph. In the first case, the verifier checks the commitment. In the else case, the verifier checks that the number of colors in V' is at most k and checks V' indeed separates.

Unfortunately, above protocol is not zero-knowledge. Indeed, it leaks the size of V', which it should not. The knowledge of the size of V' may help the verifier to find the k colors. Moreover, it also leaks the multiplicity of each color, etc.

4.3 Avoiding this problem

To solve this problem, we prove the following lemma.

Lemma 10. *Let $G_c = G_c(V, E, C, f)$ be a vertex-colored graph. Let $C' \subseteq C$ be such that $|C'| = k$ and $V' = \{v'_i : f(v'_i) \in C'\}$ separate A and B. Let k' be the maximum number of vertex disjoint paths in (V, E)* ignoring the colors. *Let P_1, P_2, ..., $P_{k'}$ be these vertex disjoint paths. We then have that for each of these path P_i: $P_i \cap V' \neq \emptyset$. So, on each path P_i there exists a node of a color in C'.*

Proof. The proof follows trivially by contradiction.

We now use this lemma to provide a zero-knowledge interactive proof.

4.4 The protocol

Let $G = G(V, E, C, f)$ be a vertex-colored graph and $m = |C|$. For simplicity we assume $C = (1, 2, \ldots, m)$. Let C' and V' be as in Section 4.3.

First the verifier and the prover (separetely) compute:

- k', i.e. the maximum number of vertex disjoint paths ignoring colors.
- k' vertex disjoint paths P_1,P_1, P_2, ..., $P_{k'}$.

This can be done in polynomial time [4]. So both prover and verifier obtain the same k' vertex disjoint paths. Let l_i be the length of the path P_i minus one, and let us call the vertices, except A and B, on this path $v_{(i,1)}$, $v_{(i,2)}$,...,$v_{(i,l_i)}$.

Then they repeat the following steps n times, where n is specified later. The randomness in each run is chosen independently.

Step 1 The prover chooses a permutation π of the colors, so $\pi \in_R sym(\{1, \ldots, m\})$. For each of the aforementioned paths P_i:

- the prover chooses a permutation $\rho_i \in_R sym(\{1,\ldots,l_i\})$, permutes the vertices (ignoring A and B) on the path P_i and sends the verifier a commitment for the permuted coloring of the permuted vertices, so formally, sends:

$$E_{(i,j)} = commit(\pi(f(v_{(i,\rho_i(j))})), r_{ij}) \text{for } j = 1,\ldots,l_i,$$

  where r_{ij} is chosen independently uniformly random, and
- for each $c_h \in C'$ $(h = 1,\ldots,k)$ sends $E'_h = commit(\pi(c_h), r'_h)$, where r'_h is chosen independently uniformly random.

Step 2 The verifier flips a coin q_1 and also chooses randomly a value $q_2 \in_R \{1,\ldots,k'\}$ and sends the prover the query (q_1, q_2).

Step 3 *If* $q_1 = 0$, *then* the prover reveals π, all ρ_i and opens all commitments of the type $E_{(i,j)}$ (Note the prover does not open E'_h.),
else the prover decommits one (of the) permuted colors of the vertex set: $P_{q_2} \cap V'$. This is done by opening:
- exactly one $E_{(q_2,j')}$, and
- exactly one E'_h

such that $f(v_{(q_2,\rho_{q_2}(j'))}) = c_h$. (Note π is not opened, and neither is ρ_{q_2}

Step 4 *If* $q_1 = 0$, *then* the verifier verifies that π and all ρ_i are permutations and all the decommitted values,
else the verifier checks that the two opened commitments and checks that they correspond to the same color.

Theorem 11. *When n is chosen such that $((k'-1)/k')^n$ is negligible, the protocol is a computational zero-knowledge interactive proof system for the color separable problem assuming that the commitment function* commit *is secure.*

Proof. (**Sketch**) We have perfect completeness, which is indeed trivial. We now prove soundness. Suppose that the graph is not k-color separable. Then a separator will have at least $k+1$ different colors. However, the prover only commits to k colors by using the commitments E'_h in the zero-knowledge proof. The prover could try to commit incorrectly to $E_{(i,j)}$ or choose π and ρ_i that are not commitments. However, the prover would be caught with probability $1/2$ if this was the case. Assume now that π, ρ_i and $E_{(i,j)}$ are correct. The best case for the dishonest prover occurs when we have that for all, except one, path P_i there is a color on the path that is in the one of the k colors committed in E'_h. The conditional probability the verifier does not catch this is $1/k'$. Thus, the conditional probability the dishonest prover fools the honest verifier is $(k'-1)/k'$. However, since the protocol is repeated independently sufficiently many times, the probability the dishonest prover convinces the verifier of an untruth is negligible.

We now prove zero-knowledge. The simulator first guesses a query (q'_1, q'_2) with the same probability distribution as a honest verifier. We now explain the simulation of Step 1. If $q'_1 = 0$, the simulator chooses random permutation π' and ρ'_i and makes commitments for these. The simulator also chooses a subset of k colors and commits to these. In the case $q'_1 = 1$, the simulator chooses a uniformly random color c'. Then the prover chooses $k-1$ other colors. He creates

commitments for these k colors and call these E'_h. All the colors of the type $E_{(i,j)}$ are chosen randomly, except for one j and for $i = q'_2$ for which the color c' is chosen.

The commitments are presented to the verifier who sends (q_1, q_2). If $(q_1, q_2) \neq (q'_1, q'_2)$, then the simulator rewinds. Otherwise the simulator continues. He is able to answer the query correctly, as is trivial to verify. Due to the assumption on the commitment function, the zero-knowledge is computational.

This proved the theorem.

5 Conclusion

In practice the connectivity of a network may be small and then the research has only theoretical value. However, when wifi technology is used, this may no longer be true. Unfortunately, the results in this paper are for point-to-point communication. The work by [9,8,20,6] has demonstrated that even for an adversary bounded by a threshold, the problem of reliability and security in partial broadcast communication is much more complex. We believe that generalizing our results for a color based adversary structure to partial broadcast networks is a true challenge.

Acknowledgment

The first author thanks those who asked him how research on color based adversary structures could be used in settings where one wants censorship or prevent censorship.

References

1. M. Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, pp. 1444–1451, August 3–11, 1987. Berkeley, California, U.S.A., 1986.
2. G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2), pp. 156–189, October 1988.
3. M. Burmester and Y. G. Desmedt. Is hierarchical public-key certification the next target for hackers? *Communications of the ACM*, 47(8), pp. 68–74, August 2004.
4. T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press and Mc. Graw-Hill, 1990.
5. Y. Desmedt, Y. Wang, and M. Burmester. A complete characterization of tolerable adversary structures for secure point-to-point transmissions without feedback. In X. Deng and D. Du, editors, *To appear in: Algorithms and Computation, 16th Annual International Conference, ISAAC 2005, (Lecture Notes in Computer Science)*, pp. 277–287, 2005. December 19 - 21, 2005, Sanya, Hainan, China.
6. Y. Desmedt and Y. Wang. Perfectly secure message transmission revisited. In L. Knudsen, editor, *Advances in Cryptology — Eurocrypt 2002, Proceedings (Lecture Notes in Computer Science 2332)*, pp. 502–517. Springer-Verlag, 2002. Amsterdam, The Netherlands, April 28–May 2.
7. Y. Desmedt and Y. Wang. Efficient zero-knowledge protocols for some practical graph problems. In *Third Conference on Security in Communication Networks '02 (Lecture Notes in Computer Science 2576)*, pp. 296–308. Springer-Verlag, 2003. Amalfi, Italy, September 12-13, 2002.
8. M. Franklin and R. Wright. Secure communication in minimal connectivity models. In K. Nyberg, editor, *Advances in Cryptology — Eurocrypt '98, Proceedings (Lecture Notes in Computer Science 1403)*, pp. 346–360. Springer-Verlag, 1998. Espoo, Finland, May 31–June 4.

9. M. Franklin and M. Yung. Secure hypergraphs: Privacy from partial broadcast. *SIAM J. Discrete Math.*, 18(3), pp. 437–450, 2004.
10. 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.
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. *Journal of the ACM*, 38(1), pp. 691–729, July 1991.
12. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1), pp. 186–208, February 1989.
13. M. Hirt and U. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1), pp. 31–60, 2000.
14. M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structures. In *Proc. IEEE Global Telecommunications Conf., Globecom'87*, pp. 99–102. IEEE Communications Soc. Press, 1987.
15. M. Kumar, P. Goundan, K. Srinathan, and C. Rangan. On perfectly secure communication over arbitrary networks. In *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 193–202, 2002.
16. B. Meier and J. N. Wilford. How the gospel of Judas emerged. The New York Times, April 13, 2006. http://www.nytimes.com/2006/04/13/science/13judas.html
17. Gospels of Matthew, Mark, Luke, and John written. http://www9.nationalgeographic.com/lostgospel/timeline_04.html
18. NSA access codes have been secretly built into windows. http://ureader.co.uk/message/792934.aspx.
19. Social networks, quarterly journal.
20. Y. Wang and Y. Desmedt. Secure communication in broadcast channels. In J. Stern, editor, *Advances in Cryptology — Eurocrypt '99, Proceedings (Lecture Notes in Computer Science 1592)*, pp. 446–458. Springer-Verlag, 1999. Prague, Czech Republic, May 2–6.
21. K. Zetter. Cisco security hole a whopper. July 27, 2005. http://www.wired.com/news/privacy/0,1848,68328,00.html?tw=wn_tophead_2