The Pump: A Decade of Covert Fun

21st Annual Computer Security Applications Conference, Dec. 2005 Invited Classic Paper

Myong H. Kang, Ira S. Moskowitz, and Stanley Chincheck

Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, DC 20375
{mkang, moskowitz, chincheck}@itd.nrl.navy.mil

Abstract

This paper traces the ten plus year history of the Naval Research Laboratory's Pump idea. The Pump was theorized, designed, and built at the Naval Research Laboratory's Center for High Assurance Computer Systems. The reason for the Pump is the need to send messages from a "Low" enclave to a "High" enclave, in a secure and reliable manner. In particular, the Pump was designed to minimize the covert channel threat from the necessary message acknowledgements, without penalizing system performance and reliability. We review the need for the Pump, the design of the Pump, the variants of the Pump, and the current status of the Pump, along with manufacturing and certification difficulties.

1. Introduction

This paper describes the evolution of what we generically call the *Pump*. Myong H. Kang was working on a multilevel secure database (SINTRA) project at the time. He was concerned with reliable data replication from a lower level (Low) to a higher level (High). There were many contenders for such a data replicator, but they were bulky (e.g., XTS-200 requires an extra host for each network that it is connected to), expensive, and could not satisfy all of the reliability, fairness, and robustness requirements that were desired. Ira S. Moskowitz was researching the information theoretic basis of covert communication channels. Kang approached Moskowitz to discuss his concerns that the necessary message acknowledgements from High to Low could be used as a covert channel. They went on to write [8]. This then turned into a cottage industry of Pump-like papers [5,9,10,11,12,13,14,18,19,20]. We note that the first Pump paper is rather rudimentary and has plenty of rough edges. Today, the Pump has been type accredited by the Navy and is being produced and distributed as the patent pending *Network Pump*™ [22] based upon the ideas in [13]. It is currently being used operationally in many locations. The engineering lead on the *Network Pump*™ is Stanley Chincheck, also of NRL, who started this task in 1996. Chincheck had seen the need for this capability in more practical applications in the Navy. Knowing the importance of transitioning technology to the Fleet, he began a journey to take a mathematical theory and turn it into a real world product. We felt that after ten years it would be of interest to share with our colleagues the growing pains of transitioning a research idea into a "purchasable product" at a government laboratory.

One must keep in mind that the Pump is not quantum physics or the complete works of Shakespeare. It is an engineering solution to a quasi-impossible problem. We wish to stress that contrary to some common folklore, the Pump does *not* eliminate all covert channels. Rather, the Pump minimizes the covert channel risk to acceptable bounds by pragmatic engineering and parameter tweaking.

The Pump was designed in two steps. First, the "basic Pump" serves only one sender and one receiver. Second, the "network Pump" (which is the basis for the *Network Pump*™ modulo some changes) services many senders and receivers from different applications (e.g., file transfer, database replication) simultaneously. We simply use the word Pump for dealing with the entire body of work that applies across the board. That is, Pump refers to all of the ideas behind the device and the actual physical box with its specialized code [22].

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2005		**00-00-2005 to 00-00-2005**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The Pump: A Decade of Covert Fun	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,Center for High Assurance Computer Systems,4555 Overlook Avenue, SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**7**	

2. Design Requirements

We review the design requirements that led to the design of the Pump. The details are throughout the numerous papers [8,13].

1. **Assurance**: The design of the Pump should be simple and able to facilitate its being evaluated at a high assurance level. Security related functionality should be well isolated to reduce the complexity of critical components.
2. **Reliability**: The reliability requirement for the Pump can be simply stated as: no loss of messages, and no duplication of messages. The reliability should be guaranteed even if the connection between Low and High is broken temporarily.
3. **Performance**: The Pump should not arbitrarily limit the data transfer rate in order to reduce the covert channel capacity. Furthermore, the Pump, by slowing the receiving rate to alleviate congestion, should not lessen total throughput.
4. **Covert channel**: The Pump should reduce covert channel capacity as much as possible without compromising performance.
5. **Fairness**: The (network) Pump is designed to accommodate many senders and many receivers. Therefore, if the load of data traffic offered to the Pump exceeds its capability, the load reduction must be performed in a fair manner for all the sessions that share the Pump.
6. **Denial of service attack**: Since the Pump may be a shared resource among several sessions (between senders and receivers), services for other sessions can be potentially disrupted if too much resource is allocated to one particular session. The design of the Pump should prevent such a situation.

3. Logical Design

In this section, we describe logical design of the Pump: basic Pump and network Pump.

3.1. Basic Pump

The basic Pump is a device that balances the first four requirements in section 2. An abstract view of the Pump is as follows (see figure 1):

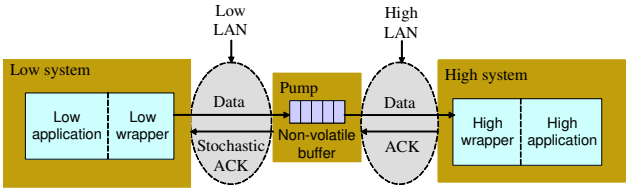

Figure 1: The basic Pump

The basic Pump places a non-volatile buffer (size *n*) between Low and High, and sends acknowledgements (ACKs) to Low at probabilistic times (i.e., stochastic ACKs) based upon a moving average of the past *m* High ACK times [8]. A High ACK time is the time from when the buffer sends a message to High to the time when High sends an ACK back to the basic Pump. By sending ACKs to Low at a rate related to High's historical response rate, the basic Pump provides flow control and reliable delivery without unduly penalizing performance and covert channel requirements. We emphasize that ACKs are not passed through the basic Pump from High to Low. In fact, the basic Pump can acknowledge receipt of messages from Low before High receives them (otherwise a buffer would not be necessary). Each ACK sent to Low is generated internally by the basic Pump only in response to a message from Low. The average rate at which these ACKs are sent from the basic Pump to Low reflects the average rate at which High acknowledges messages from the basic Pump. This guarantees that Low does not pay an undue performance penalty due to security reasons.

Since the Low ACKs are decoupled from High ACKs, the rate of the ACKs from the basic Pump to Low does represent only downward flow of information. However, the algorithm controlling the rate at which acknowledgments are returned is parameterized to allow the capacity of this timing channel to be made as small as accreditators may require.

The evaluation process of high-assurance devices is a difficult and lengthy one. Thus, we want to make the Pump a generic one-way device that supports many classes of applications. To achieve this goal, the Pump supports a specialized protocol, the *Pump Protocol*, across the local area network (LAN) interfaces (see figure 1). Within the low and high systems are *wrappers*, specialized software that supports a variety of applications. Wrappers, which run on the low and high systems, communicate with the basic Pump over their respective LANs. Although not shown in the figure, each wrapper consists of two parts. The *application-specific* part provides an application-

specific protocol on one side and the *Pump-specific* part provides Pump protocol on the other. It can be tailored to support the particular set of objects or calls to the application it expects to see. The *Pump-specific* part is a library of routines that implement the Pump protocol. It supports the Pump's application program interface (API) on one side and the Pump protocol on the other. The application-specific part can call the Pump-specific API as required. Note that the Pump protocol and the application protocol are application-level protocols. Thus, the Pump provides application-to-application reliability.

This structure of separating wrappers from the Pump (this applies to the network Pump as well) has several interesting aspects:

- The Pump's confidentiality properties depend solely on the Pump itself, not on the wrappers. Thus, wrapper software is not security-critical and can be altered or replaced without affecting system security.
- Wrappers make the Pump a generic device that is independent of a specific application. Thus, the Pump can be used in conjunction with many applications (e.g., messaging systems, file transfer, database replication).

By focusing on reliable message delivery without compromising confidentiality from a receiver to a sender, the design of the basic Pump becomes simple and easy to understand, this satisfies the assurance requirement [14].

3.2. Network Pump

The basic Pump deals with only one Low and one High. To make the Pump a generic network security device, network Pump theory was introduced [9, 13] and developed as the Network Pump™. As stated before, we refer to the broad theory that covers a network version of the Pump as the *network Pump*, whereas *Network Pump*™ refers to the particular hardware device and its specialized code, based on the network Pump, which is currently being patented. We include a section of the design of the Network Pump™ in section 3.3.

The network Pump satisfies not only the first four requirements, but also the last two requirements in section 2**.** The network Pump acts as a router that connects Low applications to High applications. To provide fairness and prevent denial of service attacks, the network Pump is structured as follows (figures 2 and 3):

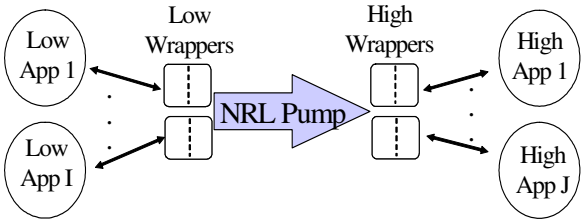

Figure 2: Network Pump

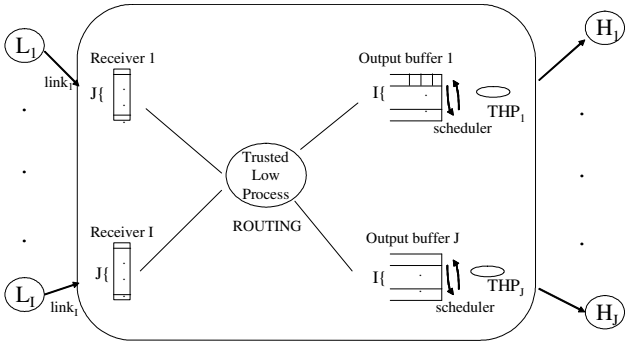

Figure 3: Internals of the network Pump

L_i (H_j) is the set of inputs (outputs) to the network Pump. We can consider them as wrappers in figure 2. $Receiver_i$ has J slots for receiving messages from L_i. $Slot_j$ stores a message from $session_{ij}$ (i.e., a connection between L_i and H_j) until it is routed by the Trusted Low Process (TLP). The TLP takes a message from a receiver and routes it to the appropriate output buffer.

After the message is routed to the output buffer, the TLP is ready to send an ACK back to the appropriate L_i. The time this ACK arrives at L_i depends on the randomization scheme. There are I logical output buffers for H_j, each denoted as $buffer_{ij}$. A message from $session_{ij}$ will be stored in $buffer_{ij}$. Trusted High Process j (THP_j) delivers a message from $buffer_{ij}$ to H_j according to a scheduling scheme. The network Pump uses round robin scheduling scheme to guaranty max-min fairness [6]. THP_j cannot deliver another message from $buffer_{ij}$ until the prior message from $buffer_{ij}$ is ACKed (by H_j).

The number of messages in $buffer_{ij}$ is important to achieve fairness [6] (the bigger the number of messages in $buffer_{ij}$ the fairer). This is because our round-robin scheduler does not take bursting into account. The way to handle bursty behavior is to have enough messages queued in $buffer_{ij}$ so that times of abundance and starvation (with respect to message arrivals) are balanced out. In fact, it is desirable to keep the queue length in $buffer_{ij}$ positive so that max-min fairness is

preserved. However, if the queue length becomes too big we have potential covert channel and denial of service problems. Thus, it is desirable to keep the message queue length at a certain level. To address this issue, we introduce the concept of *Fair size*, which is a configuration parameter targeted for the desirable number of messages [13] in the output buffer. Intuitively, the more bursty the input, the larger the Fair size must be. Note that Fair size has to be intelligently chosen so that one session cannot dominate (fill) the total output buffer and at the same time large enough to accommodate bursting. Good design requires that the total output buffer have at least Fair size surplus spaces in addition to sum of the Fair size spaces allocated for all sessions. Intuitively, if all sessions are active and behaving, this design leaves us with at least Fair size spaces in the total output buffer. The fair size and the modified ACK time scheme [13] are also help to minimize covert channel because its modified ACK scheme prevent the output buffer being full.

Since the network Pump has a built-in mechanism to share output buffers fairly among different sessions (i.e., moving average construction to control input rates), all output buffers are dynamically shared among different sessions.

3.3. Network Pump™

The primary goal of the custom hardware architecture of Network Pump™ [22] is the assurance that two networks at different security levels that are connected through a Network Pump™ will not compromise sensitive information. The Network Pump™ is implemented with separate microprocessors, memory, input/output (I/O) circuitry, etc. to connect the Low net and the High net with only a shared stable memory buffer in common. The Network Pump™ keeps the stable buffer from overflowing by controlling the rate at which the messages are acknowledged. The rate of the acknowledgements is random, with a mean based on the rate at which the High side has been accepting messages. This mean rate of accepting messages provides significant protection against the use of a covert channel to leak information from High to Low.

Early in the design phase, a system-wide design decision was made to separate the Network Pump™ architecture into two functional areas, a Low Side and a High Side. The Low Side (i.e., Low LAN computer software configuration item which executes on the Low Side Microprocessor, memory, and assorted hardware support components) is responsible for all control, status, and data exchange with the Low Host via the Pump Protocol. The High Side (i.e., High LAN computer software configuration item, which executes on the High Side Microprocessor, memory, and assorted hardware support components) is responsible for all control, status, and data exchange with the High Host via the Pump Protocol.

Communication between the High Side and the Low Side of the Network Pump™ is provided via the Interprocessor Communication Channel. This Interprocessor Communication Channel is used to send Pump Messages from the Low Side to the High Side as well as moving averages from the High Side to the Low Side within the Network Pump™. There is also limited status and control information that is exchanged between the Low Microprocessor and the High Microprocessor. Other than the Interprocessor Communication Channel, there is no resource sharing between the High Side and the Low Side. This separation reduces/minimizes the risk of data flow (or leakage) from the High LAN Interface (e.g., High Host) to the Low LAN Interface (e.g., Low Host).

In addition to interfaces to the Low and High LANs, the Network Pump™ provides an interface to an Administrator Workstation. The Network Pump™ receives initial configuration and other control information across this interface and provides error and performance reports, if requested by the Administrator. The configuration information defines which Low Wrappers, specified by IP address and port number on the Low LAN are permitted to open connections (and thereby transmit messages) to which High Wrappers, specified similarly, by IP address and port number on the High LAN. The custom hardware architecture of the Network Pump™ is shown in figure 3.

To provide reliability (i.e., not losing any messages that the Network Pump™ receives), Network Pump™ is equipped with a built-in battery (as shown in figure 4). All messages in the volatile RAM will be saved into non-volatile flash memory before the Network Pump™ shuts down in case of power failure. When the power is restored, all undelivered messages will be restored to the RAM and the Network Pump™ will operate normally.

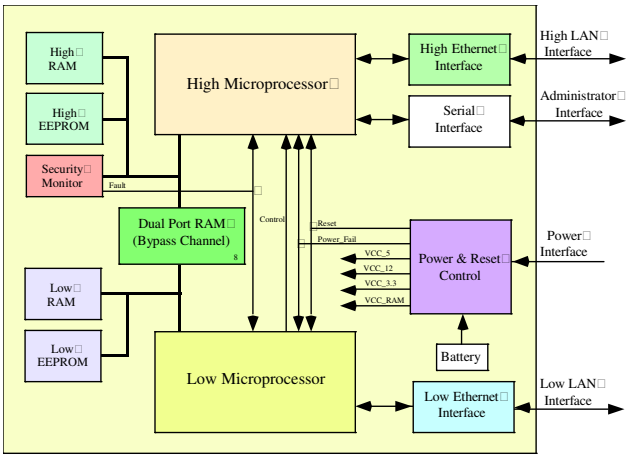

Figure 4: Network Pump™ hardware architecture

It is important to note that the Network Pump™ provides end-to-end (i.e., application-to-application) reliable message delivery. This is one of many aspects that separate the Pump from the other one-way devices.

Currently, Network Pump is produced as a rack mounted device (17.5"W x 1.75"H x 10.5"D; 19") that has Ethernet 100BaseT functionality (see Figure 5).

Figure 5: Network Pump in Production

Network Pump is being used in the Navy and other government agencies.

4. Competition and Pump Variants

The Pump was not, and still is not, accepted by all as the next best thing since sliced bread. In this section, we discuss some of the alternative ideas and an interesting variant of the Pump

4.1. One-way Link

There are other secure one-way transfer devices even though they do not satisfy all the requirements that the Pump satisfies. A one-way data diode is a straightforward way to transfer data from one domain to another domain without high assurance components. The idea is shown in figure 6:

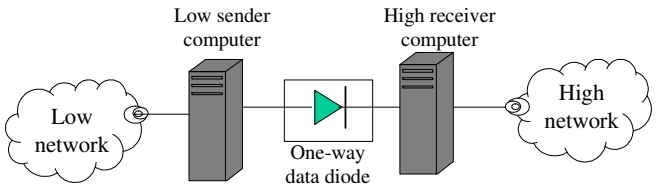

Figure 6: One-way link using one-way data diode

A one-way data diode transfers data from Low to High without acknowledgements. Since the diode can be physically validated that there is no reverse data flow, very little assurance effort is necessary to guarantee that no high information leaks to the low side. Since there is no data flow from High to Low, there is of course no covert channel. However, there is also no guarantee that the data that is sent from Low arrived at High safely. Therefore, in some applications, the same data is transferred multiple times [16] to increase the probability of a safe data transfer. In addition, a "big buffer [1] " may be necessary in the High receiving computer to prevent data loss. For example, if the data receiving rate in High receiver computer is faster than data consumption rate, data is lost in some point. The "big buffer" approach may not work if High system crashes (Low system does not know the status of the High system).

Owl Computing Data Diode (OWL) from Owl Computing Technologies is a commercial product that utilizes this idea [16], based on the patent [23]. We note that a one-way data diode was also discussed in [5]. OWL uses a pair of data diode network interface cards (NIC) that can be plug in the PCI bus of the host computers. One NIC is used as a sender interface card and the other NIC is used as a receiver interface card. Each NIC is an optical communication card that moves data uni-directionally. Two NIC cards are connected through a fiber optic cable. OWL is a common criteria EAL 2 validated product.

The major differences between one-way data diode type ideas and the Pump are as follows:

- The Pump provides reliable communication with controllable covert channel capacity. Reliable communication is essential for some applications. For example, if a transaction is lost during database replication service between a low database and a

[1] The use of a "big (enough) buffer" [16] was also an interesting alternative to the Pump. However, it does not meet all of our design requirements as stated in section 2.

high database then the high database will be out of synchronization. Thus, all applications that use the high database will be affected by this inconsistency. The Pump guarantees reliable delivery of messages even if the session between Low and High is disconnected temporarily.

- The Pump is a network device that routes traffic from one domain to another domain while providing fairness and resisting denial of service attack. On the other hand, one-way data diode does not provide any concept of routing.

4.2. Quantum Pump

Unfortunately, no one has come up with a clean closed form for the statistical covert channel [18] that arises in the Pump. One of the reasons for tweaking the Pump into a "quantum Pump" [21] was the desirability of obtaining definite closed form analysis. We still feel that such a closed form is not necessary for making pragmatic design choices; however, that is a mathematical nicety that is missing.

4.3. Current Research & Related Research

It comes as a surprise to us, but people are still publishing on the Pump. There is a recent NPS Master's thesis [7] laying the groundwork for possible common criterion considerations. There is recent work [1, 2] analyzing the covert channel that arises from connect/disconnect messages. We note that the Network Pump™ does not have this problem since the number of connections per unit time is a limited by a user-defined parameter. There is recent work [15] applying probabilistic protocol analysis to the Pump.

The ability to pass information via by affecting the time that something is in a queue or buffer is considered in other work in information theory such as [3, 4].

5. Lessons Learned

In many respects, the development of the Network Pump™ as a GOTS (Government off-the-shelf) product was easier than the effort and planning it took to transition it to a "real-world" product. Many obstacles were encountered in our efforts to prosecute the transition, everything from funding to "I'll take one when it is certified and readily available." Part of our angst revolved around the lack of an infrastructure in place for a research laboratory environment to support technology transition. Though technology transition from the research and development community is deemed important to the DoD, an overall comprehensive process to support the transition is lacking and the bulk of the work falls on the shoulders of the researchers and developers.

Several important lessons were learned in our endeavor that in many respects will apply to other efforts that take a similar path. The most notable of these observations are the followings:

- Bridge funding: Dollars critical for transitioning the product from research and development to a certified real-world product were needed
- Patient and flexible customers: Customers whose patience and understanding afford some latitude in getting the product established
- Perseverance: The quality that we, the researchers and developers, had to exhibit to make this product a reality.

Though it was a painful process, the lessons learned have made us smarter and with that, the hopes that the next time around success will be easier – especially since we are in that process …. *again*.

From a pure research point of view, we hope that one day a more general way of dealing with the subtle issues of timing channels and statistical channels will be discovered. Until then, we have real problems that must be dealt with and obtaining mathematical results that lead to information leakage bounds (instead of nice closed-form solutions) is extremely useful. Once, we have our bounds in place, we can then go on to design engineering solutions that are *good enough*! This is the philosophy that we used for the Pump, and it is probably a good philosophy to adopt for other, but not all, security solutions.

6. References

[1] A. Aldini and M. Bernado, "An Integrated View of Security Analysis and Performance Evaluation: Trading QoS with Covert Channel Bandwidth" to appear: SAFECOMP 2004.

[2] A. Aldini and M. Bernado, Measuring the Covert Channel Bandwidth in the NRL Pump, technical report 2003, http://mefisto.web.cs.unibo.it/PubblSedeC0.html

[3] V. Anantharam, and S.Verdú, "Bits through queues," IEEE Transactions on Information Theory, Volume: 42 , Issue: 1 , Jan. 1996.

[4] V. Anantharam and S. Verdú, ``Reflections on the 1998 Information Theory Society Paper Award: Bits through Queues," IEEE Information Theory Society Newsletter vol. 49, no. 4, Dec. 1999.

[5] J. Froscher, D. M. Goldschlag, M. H. Kang, C. Landwehr, A. P. Moore, I. S. Moskowitz, and C. Payne, "Improving Inter-Enclave Information Flow for a

Secure Strike Planning Application," Proceedings of the 11th Annual Computer Security Applications Conference, pp.89 - 98 (1995).
[6] E. L. Hahne. "Round-robin scheduling for max-min fairness in data networks," IEEE J. Select. Areas Commun., vol. 9, no. 7, Sep. 1991.
[7] J. S. Holmgren and R. P. Rich, Metric Methodology for the Creation of Environments and Processes to Certify a Component: The NRL Pump, Naval Postgraduate School Monterey CA, March 2003.
[8] M. H. Kang and I. S. Moskowitz, "A Pump for rapid, reliable, secure communication," Proceedings of the first ACM Conference on Computer and Communications Security, 1993.
[9] M. H. Kang, I. S. Moskowitz and D. C. Lee, "A Network Version of the Pump," Proc. 1995 IEEE Computer Society Symposium on Research in Security and Privacy. May 1995.
[10] M. H. Kang, J. Froscher, and I. S. Moskowitz, "A Framework for MLS Interoperability," Proc. HASE'96, Niagara-on-the-Lake, Canada, October 1996.
[11] M. H. Kang, I. S. Moskowitz, B. E. Montrose, and J. J. Parsonese, "A Case Study of Two NRL Pump Prototypes," 12th Annual Computer Applications Security Conference 1996.
[12] M. H. Kang and I. S. Moskowitz, "A data Pump for communication," NRL Memorandum Report, 5540-95-7771, 1995.
[13] M. H. Kang, I. S. Moskowitz. and D. C. Lee, "A Network Pump," IEEE Transactions on Software Engineering, vol. 22, no. 5, 1996.
[14] M. H. Kang, A. P. Moore, and I. S. Moskowitz, "Design and Assurance Strategy for the NRL Pump," 2nd IEEE High-Assurance System Engineering Workshop (1997). IEEE Computer Magazine, Vol. 31, No 4, 1998.
[15] R. Lanotte, A. Maggiolo-Schettini, S. Tini, A. Troina, and E. Tronci, Automatic Covert Channel Analysis of a Multilevel Secure Component, Proc. Int. Conf. on Information and Communications Security, LNCS 3269, pp. 249-261, 2004.
[16] J. McDermott, "The B2/C3 problem: How Big Buffers Overcome Covert Channel Cynicism in Trusted Database Systems," in Biskup, J., M. Morgenstern, and C. E. Landwehr, eds. Database Security, VIII: Status and Prospects. IFIP Transactions A-60, Elsevier Science B.V., Amsterdam, 1994.
[17] R. Mraz, "Secure Directory File Transfer System", Proc. 12th Annual Canadian Information Technology Security Symposium, 2000.
[18] I. S. Moskowitz and M. H. Kang, "Discussion of a statistical channel," Proceedings of IEEE-IMS Workshop on Information Theory and Statistics, Alexandria, VA, (1994).
[19] I. S. Moskowitz and M.H. Kang, **"**The Modulated-Input Modulated-Output Model," Proc. IFIP WG11.3 Workshop on Database Security, NY, August 1995.
[20] I. S. Moskowitz and C. Meadows, **"**Covert Channels-A Context Based View," Proc. Workshop on Information Hiding, Cambridge, UK, May/June 1996.
[21] N. Ogurtsov, H. Orman, R. Schroeppel, S. O'Malley, O. Spatscheck, "Experimental results of covert channel limitation in one-way communication systems," Network and Distributed System Security, 1997.
[22] US Patent application, 10/627,102, July 25, 2003.
[23] US Patent 5,703,562, Method for Transferring Data from an Unsecured Computer to a Secured Computer, C.A. Nilsen Dec 30, 1997.