Smart photonic networks and computer security for image data*

J. Campello[a], J. T. Gill[a], M. Morf[b] and M. J. Flynn[b]

[a] Information Systems Laboratory
[b] Computer Systems Laboratory

Stanford University, Stanford, California 94305

ABSTRACT

Work reported here is part of a larger project on "Smart Photonic Networks and Computer Security for Image Data", studying the interactions of Coding and Security, Switching Architecture Simulations, and Basic Technologies. Coding and Security: Coding methods that are appropriate for data security in data fusion networks were investigated. These networks have several characteristics that distinguish them from other currently employed networks, such as Ethernet LANs or the Internet. The most significant characteristics are very high maximum data rates; predominance of image data; narrowcasting — transmission of data from one source to a designated set of receivers; data fusion — combining related data from several sources; simple sensor nodes with limited buffering. These characteristics affect both the lower level network design and the higher level coding methods. Data security encompasses privacy, integrity, reliability, and availability. Privacy, integrity, and reliability can be provided through encryption and coding for error detection and correction. Availability is primarily a network issue; network nodes must be protected against failure or routed around in the case of failure. One of the more promising techniques is the use of "secret sharing." We consider this method as a special case of our new space-time code diversity (STCD) based algorithms for secure communication. These algorithms enable us to exploit parallelism and scalable multiplexing schemes to build photonic network architectures. A number of very high-speed switching and routing architectures and their relationships with very high performance processor architectures were studied. Indications are that routers for very high speed photonic networks can be designed using the very robust and distributed TCP/IP protocol, if suitable processor architecture support is available.

Keywords (5): Coding, Security, Networks, Switching-Architectures, Photonics

1. INTRODUCTION

Smart Photonic Networks (SPNs) provide secure and robust high speed data communication needed for capturing, processing, and transmitting images and image-format data. These tasks impose major challenges for the design of reliable and secure information networks and processing systems. Designing optimal SPNs involves specifying flexible architectures with intelligent and reconfigurable networks to cope with extremely variable requests for high rate data, and determining coding techniques to provide for security needs, such as privacy, integrity, availability, and reliability of high bandwidth optical communication links.

The very high speed optical communication characteristics of SPNs present several challenges in the area of data security. Challenges arise from the high data rates and the corresponding large latency (round trip delay measured in bits). The coding and encryption methods that were designed to perform at speeds appropriate for electronic communications may be too slow for optical data rates. Simple protocols such as ARQ (Automatic Repeat reQuest) require extremely large buffers to store real-time data for possible retransmission.

The primary security needs for SPNs are:

Privacy : The confidentiality of the data in the network should be maintained even under the assumption that an eavesdropper can tap several links in the network.

Integrity : It must be assured that the data is not modified accidentally or deliberately in transit, by replacement, insertion or deletion.

Reliability : The data communication should be robust enough to withstand link failures and misrouting.

*This work was supported by BMDO via a Dept. of the Army grant DAAH04-95-1-0123.

Availability : The communication network should be flexible enough to maintain a minimum level of throughput even under the severe constraints enforced by high rate image data and link failures.

We investigated integrated approaches that solve these data protection issues using unified coding methods. The several requirements impose different kinds of constraints that are usually not satisfied by traditional coding schemes.

As pointed out above, there seems to be an important tradeoff between reliability and privacy. The classical approach to this problem is to first encrypt the data to obtain the desired level of privacy and then use error control codes to ensure reliability. This simple approach can not be applied to the scenario envisioned here, since existing encryption techniques cannot provide the necessary privacy levels at very high data rates required by image data in all optical networks.

To achieve both privacy and reliability, we use a special form of "secret sharing" techniques, which can be considered to be a form of space-time code diversity. Instead of simply sending multiple copies of the message, shares of the secret message are transmitted over different paths.

2. SECRET SHARING REVIEW

Secret Sharing Schemes were first introduced by Blakley[1] and Shamir.[2] They were designed for the purpose of securely managing cryptographic keys. Cryptographic systems in general are a good means of "scrambling" a message so that only the person who holds the key will be able to reveal the message. The problem with this scenario is that the person might lose the key and then the contents of the message will be lost. One might then make several backups of the key to protect against the possibility of losing the key. The problem with this solution is that now there are several copies of the key lying around and therefore it is much easier for an enemy to obtain a copy of the key and therefore the contents of the message. Secret sharing schemes solve this problem by breaking the key into several pieces called *shares* in such a way that only when the group of shares in the set of allowable groups, called the *access structure*, is obtained can the key be recovered. The scheme is *perfect* if a subset of the set of shares that is not in the access structure reveals absolutely no information about the key, which is now called the secret.

The first Secret Sharing Schemes were Threshold Schemes, i.e., schemes in which the the access structure assumes the following simple form: all the subsets with k or more of the n shares is allowed to recover the secret. Early secret sharing schemes formalized the definition as: A (k, n) secret sharing system breaks the secret s, chosen from a set S, into n pieces, $v_1, v_2, \ldots, v_n$, each chosen from a set V, such that the following conditions are satisfied[3]:

1. The secret s is recoverable from any k pieces ($k \leq n$).
2. Knowledge of $k-1$ or fewer pieces provides absolutely no information about s.

   To avoid excessive data expansion, we also require:

3. $|V| \leq |S|$. That is, each piece v_i is to be no longer than s.

We are generally interested in the case where s is chosen uniformly from S and then $|V| = |S|$ to satisfy 2.

For instance, a very simple (n, n) secret sharing scheme is the one where for a secret message s the first $n-1$ shares, v_i, for $i = 1, 2, \ldots, n-1$, are random elements of $GF(q)$ and

$$v_n = s - (v_1 + v_2 + \ldots + v_{n-1}) \bmod q.$$

To recover the secret once all the shares have been obtained one simply adds all the shares, since

$$\begin{aligned} \sum_{i=1}^{n} v_i &= \left(\sum_{i=1}^{n-1} v_i\right) + v_n \\ &= \left(\sum_{i=1}^{n-1} v_i\right) + \left[s - \left(\sum_{i=1}^{n-1} v_i\right)\right] \\ &= s \bmod q. \end{aligned}$$

A generalization of this scheme is the one based on polynomial interpolation proposed by Shamir.[2] To generate the (n, k) threshold scheme, given the secret s in GF(q), where $q > n$, first generate $k-1$ other elements, $r_1, r_2, \cdots, r_{k-1}$ at random from GF(q), next form the polynomial

$$p(x) = s_0 + r_1 x + r_2 x^2 + \cdots + r_{k-1} x^{k-1}.$$

Then generate the n shares $(x(i), p(x(i)))$ where the $x(i)$'s are distinct nonzero elements of GF(q). Given k or more shares, the polynomial $p(x)$, and consequently the secret s, can be recovered by interpolation. If only $k-1$ shares are available, then to each choice of the last missing share corresponds a different key and therefore the $k-1$ shares provide no information whatsoever. Unlike the previous scheme, if up to $n-k$ shares are lost, the secret can still be recovered from the remaining shares. This advantage comes at the expense of a more complicated key generation and secret recovering schemes.

Another method of implementing the (k,n) threshold scheme that is of interest to us is the Matrix Scheme proposed by Karnin, Greene and Hellman.[3] In this scheme, a $k \times n$ matrix A over GF(q) is used for the generation of shares and the recovery of the secret. The matrix A should have the property that any $k \times k$ submatrix is nonsingular. Then to share the secret $s \in$GF(q), generate $k-1$ other elements, $r_1, r_2, \cdots, r_{k-1}$ at random from GF(q), then perform the multiplication

$$v = [s, r_1, \cdots, r_{k-1}]A.$$

Since the secret is in the first position of the multiplying vector, we will further require that the matrix A does not contain the column $[1, 0, 0, \cdots, 0]^T$. Given these conditions, it is easy to see that given any k or more shares, the $k \times k$ submatrix can be inverted and the secret recovered. If $k-1$ shares are available, then to each of the q possible values for the last share corresponds a value for the secret and therefore no information is gained by the knowledge of the $k-1$ shares.

It is interesting to note that in the above scheme, like in Shamir's scheme, $k-1$ random elements of GF(q) are used to help obtain a perfect scheme. An alternative to this use of random elements, would be to use k different secrets. The resulting scheme would not be perfect, since each share reduces by 1 the dimensionality of the space in which the share vector $s = [s_0, s_1, \cdots, s_{k-1}]$ might be located. In the case of only one secret and $k-1$ random elements this reduction in dimensionality was not important as long as there were at least q points left in the space. Now, since there are k secretes which have a uniform distribution in GF(q), each reduction in the dimensionality implies a reduction of $\log_2(q)$ bits of uncertainty of the vector s. The original uncertainty is $k \log_2(q)$ bits and it reaches 0 when k or more shares are available. Alternatively, t secrets and l random numbers can be used, where $l + t = k$. Then if up to l shares are available, no information is gained and from then on the remaining uncertainty of the secret vector decreases linearly down to 0 with each additional compromised share. Schemes that have this linear gain in information are called *ramp schemes*.

2.1. Using Secret Sharing Schemes for Secure Data Transmission

It was pointed out by Dixon (as reported by Blakley[1]) that secret sharing schemes may be used in a *courier mode*. In this mode, instead of dividing a key into shares and giving them to not completely trustable subordinates, the messages themselves get broken into shares. The shares are given to different couriers who take them through different, not completely reliable, paths to the destination. Two different types of threats exist. The first is that a courier may not arrive at the destination. This might be due to several reasons: for instance, he might be killed during the journey (without the share being revealed to anyone), or he might be captured by the enemy, or he might simply betray the sender. in the last two cases the share he carried is turned over to the enemy. The cautious receiver should assume that the share is captured by the enemy if it doesn't arrive. The other threat is that of some of the shares he receives might be modified, that is, the courier, allied with the enemy, modifies the contents of his share in an effort to render the message unintelligible or to have it misinterpreted. Therefore, if at most l couriers do not arrive and the number of the betraying couriers that arrive is at most b, where $l + 2b = k$, then the (k, n)-threshold scheme based on matrices can be used to achieve perfect security.

The transmissions do not have to occur at the same time: some of the couriers can be sent before hand, say during times of peace, with some shares and later during war the other shares can be sent with current information in a perfect security manner. This is essentially what is done in the one-time pad.

The one-time pad is the only cryptographic scheme known to have provable security. It works as follows: first a very large sequence of random bits, r, is generated and secretly given to both the transmitter and the receiver. This sequence should be longer than the sum of the lengths of all the messages to be transmitted. Then, when time comes for a transmission, the transmitter sends

$$r_1 \oplus m_1$$

where r_1 is the first M_1 bits in the sequence r and M_1 is the length of the message m_1. If an enemy sees the transmission, he cannot obtain any information about m_1, since r_1 (and consequently $r_1 \oplus m_1$) can take on any of the 2^{M_1} possible values with equal probability. In other words it is the same as if the enemy guessed the message without seeing the transmission.

The idea of sending messages by several couriers can be extended to the context of modern computer and communication networks. Typically in a network, several different paths are available between a given source and destination pair (e.g., uses TCP/IP protocols). These different paths can be used as different couriers in the scheme described above. Thus if the number of distinct paths that reliably transmit the message is greater than the maximum number of paths an eavesdropper can tap, the concept of secret sharing can be used to provide both reliability and secrecy. The main problem with the scheme as proposed is managing the amount of bandwidth expansion, i.e., for each message, several shares of the same size as the message are sent through the network. This is an extreme case of bandwidth expansion. The other extreme case is that of computational complexity expansion. This is usually what happens in ordinary cryptographic systems. In order to achieve the desired level of security (measured in the computational complexity sense) one needs to use fairly complex encoding and decoding algorithms. Alternatively, a ramp scheme can be used in conjunction with a not too complex cryptographic system. The ramp scheme does not have a severe bandwidth expansion problem and since only information on the several messages as a whole are obtained the eavesdropper would have to simultaneously break all the cryptosystems. Therefore, only a moderately secure cryptosystem is needed and consequently the encoding and decoding can be done without excessive computational overhead.

The kind of flexibility provided by the use of ramp schemes in conjunction with simple encryption is particularly useful in the context of photonic networks. The reason is that in such networks, unlike standard electronics networks, the price of computational complexity is extremely high. Basically there is a large amount of bandwidth, but there is a lack of processing power to implement sophisticated encoding schemes. Hence, it is useful to be able to trade off computational complexity with bandwidth. Another reason that the mixed scheme is well suited to photonic networks is the natural difficulty in tapping photonic transmissions. Therefore the requirement that several of the links need to be tapped before any information is obtain does enhance the security level of the network.

2.2. A Simple Example

To further clarify the ideas proposed above, lets consider the very simple network depicted in Figure 1.

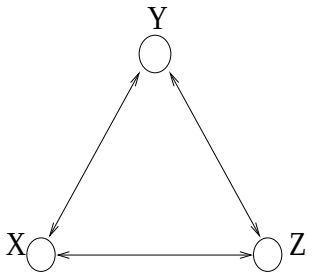

Figure 1. Simple network with tree nodes labeled X, Y and Z and three links one for each pair of nodes.

Secret sharing, even in its simplest form, requires at least two logical channels between source and destination. For secret sharing to be of any use, the eavesdropper must be unable to tap all logical channels. Hence, in the three-node case depicted in Figure 1, if we assume that the eavesdropper can tap only one of the three links, secret sharing can be used to guarantee that the eavesdropper does not obtain any information about the communication

between any pair of nodes. For instance, if nodes X and Y want to communicate with each other, the two logical channels are the direct link between X and Y, and the indirect path though node Z, who would therefore be acting as a relay station. Then to send a bit, m, of information from X to Y with perfect security, X generates a random bit r and sends the $m \oplus r$ through one of the logical channels and r through the other as shown in Figure 2. Y after

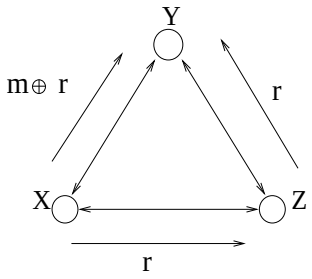

Figure 2. Example of the use of Secret Sharing to provide secure communication.

receiving the two independent transmission can easily reconstruct the message by taking the XOR ($\oplus$) of the two received bits. The eavesdropper on the other hand sees only one of the two independent transmissions, so he clearly obtains no information about the message m.

Even in the simple example above, a number of implementation issues arise. One of the apparent issues is *latency*. The two logical channels above have most likely different latencies, since they involve different number of physical links. There are several ways to manage these differences in latencies. The simplest approach would be for the receiver to wait for both independent messages to arrive, by delaying the earlier message. In this case, the overall latency of the transmission is equal to the maximum of the latencies of the two independent logical channels. However, if message latency is an issue, it would be advantageous to send the message through the lower latency logical channel (typically the direct link from X to Y), the higher latency of the other channel can be hidden by generating and transmitting the random numbers ahead of time, so that the different shares arrive just in time to be combined in the receiver to regenerate the message. Alternatively, the relay node Z can generate and broadcast the random r in a timely fashion to the other nodes, as depicted in Figure 3.

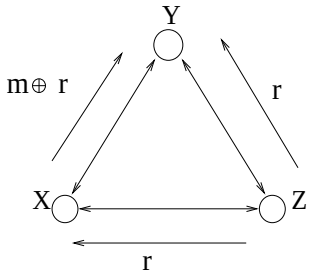

Figure 3. Z generates the random number and distributes it to X and Y.

This simple example can further be used to illustrate the use of ramp schemes, introduced above as a means of trading off the bandwidth expansion with security. For instance, in the broadcast scenario of Figure 3, to transmit one bit of information from X to Y three link usages and one random bit is required. If Y also sends a bit of information to X with perfect secrecy, another three link usages and an additional random bit are necessary. However, if a ramp scheme is used system resources can be saved, at the cost of some leak of information. For instance, once X and Y receive the random bit from Z they can both use this same random bit to transmit their messages, as shown in Figure 4. In this case, only four link usages and one random bit are required. This economy on system resources comes at the expense of some possible information leak. If the eavesdropper happens to be listening to the bidirectional link between X and Y, he obtains the two transmissions $r \oplus m_x$ and $r \oplus m_y$ and therefore can

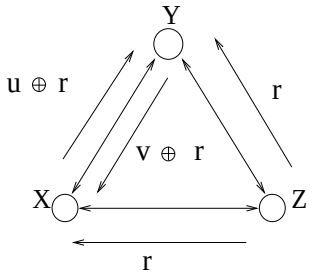

Figure 4. Example of the use of ramp schemes to tradeoff bandwidth expansion and security level.

add the two to obtain $m_x \oplus m_y$, i.e., one bit of information about the pair of messages. This leakage of information can be compensated for by increasing link encryption, therefore enabling a three way tradeoff between the aggregate bandwidth, computational overhead and the level of security, which is exactly the intent behind our Space-Time Code Diversity (STCD).

This discussion demonstrates that there may be many opportunities to extend these methods to more sophisticated functional, logical and physical scenarios.

2.3. Discussion of Different Methods

The concept of secret sharing can be used to provide both reliability and privacy, if the number of distinct paths that reliably transmit the message is greater than the maximum number of paths an eavesdropper can tap.

Integrity is also obtained by the use of secret sharing, since an intruder would have to modify at least l ($l = n-k-2b$ and b is the number of lost shares) of the shares for a particular message before he has any hope of successfully modifying the contents of that message, i.e., modifying the message without the intended receiver realizing the message was modified.

Availability is obtained when secret sharing is combined with routing and priority schemes. The idea is to send more shares than needed to provide the desired levels of privacy, reliability and integrity and then have the network decide locally on when a certain share should be dropped to resolve possible collisions of shares for different messages. This provides the necessary flexibility to cope with stringent availability constraints of highly variable demand high rate image data on optical networks.

In spite of the conceptual suitability of secret sharing to fulfill the security needs in SPNs, we must bare in mind that secret sharing was designed with another purpose in mind. Traditionally, secret sharing techniques aimed at providing reliability and privacy for a small amount of highly sensitive information, such as keys for private-key cryptosystems. Local memory efficiency and recovery complexity were generally not an issue. Existing secret sharing methods, such as polynomial interpolation schemes, are not suitable for high speed optical environments. We have explored low complexity encoding schemes that can be implemented in optical technologies under consideration.

2.4. Implementation Issues of Space-Time Code Diversity Techniques

As mentioned above, there are a number of new aspects to be considered when applying secret sharing techniques to provide the security needs of SPNs. One of the main drawbacks of such techniques is that they expand the data to be transmitted by a factor of n. This can be broken down into a factor of k to provide security and a factor of n/k to provide reliability. Therefore, one is tempted to find the smallest value of n to provide the necessary reliability. A critical issue with this approach is choosing the proper paths to send the shares, since in the high-speed high demand environment envisioned, global information will generally not be available, hence local or distributed path allocations schemes are needed. A more promising solution appears to be using a large value of n together with a priority scheme that can locally decide to drop a certain number of messages while retaining the required level of reliability. This provides an algorithm that can adapt more efficiently to changes in the network. It also is more matched to the distributed nature of TCP/IP.

Another important aspect to be considered is that of final buffering. At the destination, the message can be reconstructed only after the first k messages arrive. Since the messages are sent asynchronously, there may be a large number of other messages arriving before enough shares of the first message are received. Therefore, the amount of memory needed for buffering may be very large. This problem can be overcome by finding secret sharing schemes that allow for preprocessing that reduces the amount of memory needed, or more generally schemes that are well matched to processor memory hierarchies.

For instance, in the very simple (n, n) secret sharing scheme described in section 2, as soon as a share arrives, it is added to the accumulated sum of the previous shares. This way, only the sum of the shares received needs to be stored. Unfortunately, it is not clear how to get this kind of nice memory conserving preprocessing for general (k, n) secret sharing schemes.

As suggested earlier, to alleviate the requirements imposed by secret sharing on the communication system, it may be advisable to combine the secret sharing with encryption in time. The encryption algorithm would be fast, but not very secure. More generally, encryption algorithms can be used in combination with secret sharing schemes that are allowed to "leak" some information, i.e., subsets of less than k shares should not completely determine the secret, but may be allowed to give some information about it. The reduction in security caused by this is compensated for by the encryption in time. This combined space and time encryption may provides more flexibility for the secret sharing schemes in order to enable memory conserving preprocessing with low processing complexity.

The raw communications links proposed for data fusion networks have small error rates, and relatively straightforward forward error correction methods can be used to reduce the link error rate to insignificant values. Data loss due to network congestion will be the dominant factor.

By sending several secret shares of a message over different links and/or at different times, the entry node in effect assumes the worst case. If it can be guaranteed that the requisite number of shares of each message reach its destination, then the message need not be stored at its entry node. Increasing the success probability can be accomplished either by reducing the number of shares needed for message reconstruction, which reduces security against eavesdropping, or by increasing the number of shares sent. The latter approach increases network congestion and therefore decreases the availability of the network.

Two dimensional (or array) coding techniques can be used to encode information both in space and in time. This is an alternative to secret sharing, where instead of shares of secrets being sent through the different paths, the messages are first encrypted using some new fast encryption algorithms (in very high-speed networks), then error control codes are used through paths. This may provide a better tradeoff between reliability and bandwidth. However, this method seems to require higher complexity codes which may result in higher latency due to processing, unless highly pipelined processing is used.

3. ARCHITECTURAL ISSUES

Once efficient implementations of Space-Time Code Diversity type techniques are considered for very high-speed networks, the issue of proper network architecture becomes very important, in particular, the network computer memory hierarchy.

All-optical highly pipelined processing implies that all optical memories are available. More complex routing schemes also require all optical memories. Not coincidentally, two dimensional (or array) coding techniques are being proposed for parallel optical memories (POMs) [A. Craig, AFOSR]. Except for the potential high density, currently envisioned POMs are not very well matched to high-speed photonic networks, because the inputs and outputs of POMs are electronic in nature. For instance, a single photonic POM output channel converts 100 photons into an electronic signal by integration within a microsecond, i.e., many orders of magnitudes from GHz rates! We have considered ways to change this situation, and make POMs more useful for all optical networks in general, and provide for potentially necessary memory for high-speed coding applications.

Pipelining needs to be employed at all levels, from devices such as photonic transceivers to circuits and subsystems, and all the way up to systems and networks, in a coherent way.

Coding techniques such as secret sharing, or more generally, Space-Time Code Diversity techniques, can also be applied to data storage in general, since information storage is a form of "data communication to the future." Such

techniques are especially useful for storage applications that involve either unreliable or nonsecure access situations, such as in wireless networks, or in the case of unreliable or nonsecure storage systems. More generally these techniques could be applied to secure distributed operating systems.

Last but not least, photonic technologies have to be matched to photonic architectures by carefully considering all the interactions between functional, logical, and physical aspects of an architecture during the design process. The most painful lessons learned in the past in the design of computer and communication systems is that the mappings between functional, logical, and physical aspects of high performance architectures are highly nontrivial. The somewhat simplistic call for "form to follow function" is useful conceptually (e.g., for defining user models). However, this call does not result automatically in high performance and secure architectures.

We have in a small way started to explore the issues discussed above. Some earlier results in this area involving photonic technologies were reported in.[4]

REFERENCES

1. G. R. Blakley, "Safeguarding Cryptographic keys," in Proc. Amer. Fed. Inform. Proc. Soc. 1979 NCC, vol. 48, pp. 313-317, June 1979.
2. A. Shamir, "How to Share a Secret", in Comm. Assoc. Comput. Mach., vol. 22, no. 11, pp. 612-613, Nov. 1979.
3. E. D. Karnin, J. W. Greene, and M. E. Hellman, "On Secret Sharing Systems", IEEE Trans. on Information. Theory, vol. 29, no. 1, 1983, pp. 35-41.
4. J.S. Powell, J.A. Trezza, M. Morf, and J.S. Harris, Jr. "Vertical Cavity X-Modulators for Reconfigurable Optical Interconnection and Routing," in *International Conference on Massively Parallel Processing using Optical Interconnections*, MPPOI 1996, Maui, HI, October 1996.