technical correspondence

On Secure Personal Computing

In a recent paper in *Communications* [1], Dorothy Denning proposes a system design to allow personal computers (PCs) to securely store files at an untrusted central facility (CF), engage in secure communication between PCs, and exchange "digitally signed" messages. The following comments take issue with some aspects of the proposed mechanism.

The central concept developed in the paper is the use of an outboard, public-key cryptographic device to act as filter betwen the PC and the net, thus preventing borrowed programs (which might contain Trojan Horses) from leaking data into the net. It is largely on this basis that the outboard crypto unit is touted as superior to processor-based encryption facilities, even though it is admitted that "it may be possible to leak information on 'covert channels' (e.g. by encoding it in the rate or quantity of transmitted ciphertext)." The assurance that plaintext is not leaked into the network through regular (noncovert) channels is based on three design features:

1. There is no clear path from the PC to the net; all output is encrypted in a public key.
2. Switching to an alternate public key for secure communication or file sharing is manually controlled (though it is proposed that alternate public keys be loaded under program control).
3. The data path required for digitally signing messages with the user's secret key is enabled manually and used only when explicitly signed messages are required.

Analysis suggests that this design may neither provide adequate security nor prevent information leakage as claimed. For example, the first design feature is not likely to be achieved in practice. Experience with network protocols indicates that a clear path from the PC to the net is usually required to pass information essential to communication, e.g. addresses, flow control information, etc. This path can be bandwidth restricted in hardware to some extent, but a fair amount of software in the PC must be trusted to prevent this path from becoming a major covert channel.

The second design feature appears to assume that communication with other PCs is expected to be infrequent, since the manual intervention required to enable such communication is unwieldy. Even with this precaution a significant amount of software in the PC must be trusted if borrowed programs are to be denied the ability to exploit this channel. For example, it must not be possible for a Trojan Horse program to substitute or add sensitive data to the legitimate text that is transmitted. Moreover, the program-directed loading of public keys proposed offers great opportunity to leak transmitted data to a third party. Yet manual loading of public keys would be cumbersome, if not infeasible, especially in situations where communication among several parties occurs over a short time interval.

The third design feature similarly presumes that transmission of signed documents is infrequent, a not unreasonable assumption if authenticity in communication and file storage is provided by means other than those usually proposed for public-key systems. The general technique for providing authenticity and secrecy using public-key encryption is to transform the data twice, using a public key for secrecy and a secret key for authenticity (signing the data) [2]. The techniques proposed in this paper for secure storage, file sharing, and communication do not employ this double transformation; only signed messages are so transformed, and thus might allow an attacker to forge messages and files. This authenticity problem, and some subtle weaknesses with respect to information disclosure resulting from transforming data only with a public key, can be avoided through the use of appropriate encryption techniques, but these techniques differ slightly from those usually proposed for conventional cryptosystems such as the DES. Yet no mention of such precautions is made in the paper and the reader is not alerted to the need to employ one or the other of these methods.

This analysis indicates that the proposed design will not, by itself, preclude information leakage by borrowed programs nor provide the authenticity usually expected of secure file storage or communication. It is not apparent that the amount of trusted software needed to prevent such leakage and ensure authenticity is reduced by the use of an outboard crypto unit as opposed to processor-accessible units. Moreover, the inboard crypto unit approach offers greater flexibility in managing and switching among various cryptographic applications. Thus an inboard crypto unit is probably preferable in this environment.

There are several other points made in the paper which deserve comment. First, the paper suggests that the problem of disavowal of digital signatures can be solved by requiring each user to sign (by hand) an agreement assuming responsibility for any documents signed with his secret key. It is highly doubtful that this approach would be acceptable in a user community accustomed to immunity from credit card loss and similar consumer protection statutes. Moreover, although the likelihood of key exposure or loss may not be so great in this environment as in others, opportunities for exposure or loss will still exist and any digital signature scheme must take into account this problem [3].

Second, the paper claims that the problem of distribution of public

keys is solved through the use of "certificates." However, disclosure of the secret key used in signing these certificates permits forgery and thus impersonation of users, so trust in a CF that produces certificates is still required. To date, proposed techniques for managing certificates have not been able to eliminate the risk of exposure of a CF key *and* readily accommodate changes in the public-key database necessitated by exposure of (secret) user keys and fluctuations in user community membership. It appears premature to claim that problems of public key distribution are solved by the use of certificates.

Finally, the paper notes the prediction of Rivest that a two- to three-chip implementation of his public-key encryption algorithm capable of 5,000 bps will soon be feasible. Based on this estimate, the paper argues that public-key encryption soon will be "competitive" with the DES, with respect to performance. Since currently available single chip (custom logic) DES implementations achieve encryption rates in the range of 256,000 (Motorola) to 1,700,000 bps (Western Digital) and a four-chip version operates at 16,000,000 bps (Fairchild), it seems that "competitive" is not an appropriate word in this context. More importantly, it is not clear that public-key systems are fast enough to support file transfers in the context of the proposed system, although this depends on one's definition of "acceptable" file transfer speeds.

The preceding comments should not be interpreted as a criticism of the concept of using encryption in personal computers for secure file storage at central facilities, secure communication, or digital signatures. Rather, this note has pointed out some potential deficiencies in the proposed system design, disputed the superiority of the design with respect to other suggested methods of incorporating encryption facilities into computer systems, and questioned some claims about digital signatures, public-key distribution via certificates, and performance comparisons between the DES and the Rivest algorithm.

STEPHEN T. KENT
MIT
Laboratory for Computer Science
Cambridge, MA 02139

1. Denning, D. Secure personal computing in an insecure network. *Comm. ACM 22*, 8 (Aug. 1979), 476-482.
2. Rivest, R., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM 21*, 2 (Feb. 1978), 120-126.
3. Kent, S. A comparison of some aspects of conventional and public-key cryptosystems. ICC '79 Conference Record, Volume 1, June 1979, pps. 4.3.1-4.3.5.

□ Regarding "Secure Personal Computing in an Insecure Network" by Dorothy Denning in the August issue of *Communications:* I doubt the viability of the cryptographic device proposed in the article. In particular, I feel that the device is too simplistic for practical use in any but the most primitive networks. My reservations are not based upon the use of public-key cryptosystems. Rather, I am disturbed by the failure to recognize that a cryptographic function must be part of a protocol layer. The obfuscatory nature of such a protocol requires that it be carefully, systematically, and consistently placed among the other protocol layers of the network. Moreover, I feel that the use of the same cryptographic mechanism to perform simultaneously both communications security and remote file protection is ill-advised.

It is my contention that any truly useful network cryptographic device must be "sandwiched" by software (or extremely intelligent hardware) on both the "red" (nonencrypted) and "black" (encrypted) sides. A bi-directional "clear text bypass" (a nonencrypted information channel) must exist to permit red-black cooperation. The red software must be trusted not to use the clear text bypass as a means of sending clear text to the black software. Finally, and most importantly, the device should not stand astride an interface protocol, but should be used as part of a true end-to-end protocol layer. The use of encryption in the communications security protocols should be clearly separated from the use of encryption in a remote, protected file storage protocol. Whether the same underlying physical hardware is used is irrelevant as long as its uses are kept logically and functionally isolated.

My concern about the viability of the proposed device may be demonstrated by a look at the difficulties of using that device in a few common network situations. I suspect that an ingenious *ad hoc* "kludge" could be applied to resolve some of the problems in a given network. This does not diminish my argument any more than the ability of a programmer to patch badly structured programs diminishes the argument for systematic programing.

Let us start with the proposition that many networks will require the user to provide error detection, flow control, sequencing, and other protocol functions either on an end-to-end basis (e.g. TCP) or across the interface to the communications subnet (e.g. X25 level 3).

1. When the central facility (CF) acts as a physical or virtual circuit connecting two users and the interface to the circuit is functionally trivial (e.g. EIA RS-232C) the proposed cryptographic device may be used without undue distress. In this case the necessary protocol functions can be placed entirely in the personal computer (PC). (I am assuming that circuit set-up operations can be performed manually.)

2. Suppose that the CF is something like AT&T's proposed ACS and requires an X25 interface. X25 is an interface protocol, not an end-to-end user protocol. Consequently, the communications subnetwork (the CF in this example) must be able to view, act upon, and generate X25 control packets. Since the proposed device enciphers all outgoing data under

Letters intended for the Technical Correspondence department should be sent to the Editor-in-Chief, Robert L. Ashenhurst, The University of Chicago, 5640 South Ellis, Chicago, IL 60637.

either the user's public key or an alternate public key, it is impossible for the CF to interpret and act upon the X25 control packets unless the CF possesses the corresponding secret keys.

Similarly, the proposed device uses the user's secret key to decipher all incoming traffic. Thus the CF will not be able to generate X25 packets recognizable by the PC unless the CF encrypts them under the user's public key.

The difficulties caused by the logical placement and functionality of the proposed device makes it impossible for a PC and the CF to effectively exchange information required for circuit set-up, datagram addressing, error control, and flow control. Moreover, if a datagram interface is used, there is no mechanism to properly sequence received datagrams prior to presentation to the proposed device. Such sequencing is necessary if the cryptographic function operates in a chained block mode.

The situation becomes even more difficult if the cryptographic unit is one more complex than that proposed and is capable of generating and receiving its own messages for the purposes of key distribution or cryptographic synchronization.

3. Suppose that the CF is to be used as a secure file storage medium. Here, too, the proposed device fails to provide the functionality needed to provide acceptable service. As in the previous example, the proposed device prevents the effective use of protocols to manage errors, sequencing, and rate of flow. In addition, the proposed device couples logically separate functions. This forces the user into compromises which would otherwise be unnecessary. The user will be forced to make a trade off between security, reliability, and overhead.

It is general practice to periodically replace cryptographic keys as they "wear out" with use and time. One would expect that keys for communications security would be updated more frequently than keys for file storage. Using the proposed device, when a file is stored in the CF, the key update procedure must be coupled with a file re-encryption procedure. This is but one undesirable linkage of logically separable functions.

Another undesirable linkage occurs because the device forces the user to adopt the same cryptographic mode for both communications security and file storage. Like the Federal Data Encryption Standard, public-key cryptosystems can be used in block, cipher feedback, and block chained modes. Block mode is effective for random access files. Cipher feedback mode is best suited for communication links where bursts of noise may be expected and traffic is composed of separate bytes of terminal traffic. Both block chaining and cipher feedback work well in communication systems where data tends to flow in sequenceable streams. If the proposed device operates in block mode the user is sacrificing the ability to transmit unblocked characters and may find it difficult to transmit anything on a noisy line. Conversely, if the proposed device uses either cipher feedback or block chained mode, then random file reads are made more complex and random file writes become quite difficult.

In order for the user to store files in the CF a user-CF protocol must exist for the user to perform file naming, creation, access, and deletion operations. These commands must be comprehensible at the CF. The user of the proposed device must transmit these commands under the "Alternate Public Key." This implies that either the alternate public key is null (causing the commands to be transmitted "in the clear") or that the CF has its own cryptographic device keyed to decipher the commands. Denning recognizes this possibility and proposes a rather strange protocol requiring the CF to decipher *all* messages and intercept those it deems to be comprehensible.

The conclusion is that it is sometimes feasible, but at an extreme cost in complexity, to violate protocol layering by combining logically distinct functions or place a cryptographic device across an interface protocol. It is important to note that this conclusion is independent of the nature of the cryptographic system used, whether they be based upon conventional single-key techniques or the more recent public-key techniques.

A better approach is to revise the proposed device to include bodies of software (and processors, if necessary) on both the red and black sides of the encryption unit. A bidirectional clear text bypass will be required for software coordination. The entire network interface protocol (X25 or whatever) will be executed in the black software. The red software will execute a red end-to-end protocol. User messages will be "wrapped" in this protocol for transmission. The red software will route that portion of the message header needed by the black software through the clear text bypass. The text and the remainder of the header will be routed through the encryption unit. The black software will execute its own black end-to-end protocol as well as manage the interface to the network. The black end-to-end protocol may become trivial if the underlying communications subnetwork performs sequencing, error control, and the like. Black software will utilize the information from the clear text bypass to determine where the outgoing message should be sent, whether virtual circuits are needed, and so forth. The black software will wrap the red header and encrypted text in its own protocol. The inverse of the above operations will be performed for traffic which arrives without errors: The black protocol wrapping will be removed from arriving messages leaving the messages generated by the red protocol. The text will be passed via the decryption unit and the header via the bypass.

It is important to logically separate the communication service function of the CF from the user level service function of a protected file manager. The CF should be merely a communications subnetwork relying solely upon the virtual circuits or message addresses supplied by the

black software. Protected file storage should be maintained in a special logical PC (probably located with the CF or even implemented on the same computer) with its own standard cryptographic device. All communication between the user PCs and this file manager would be protected by the communications security mechanism described above. The CF itself would have no ability to decrypt traffic.

The file storage mechanism need have no dependence upon the operation or existence of the communications protection mechanism. Messages for the file manager would be addressed to the file manager as a node on the network, not as part of the CF. The file manager logical PC would respond to a file access protocol. The user and file manager would interact via the communications system. Only data which their mutually agreed protocol permits would be enciphered for long term storage. This encipherment would be performed not by the communications security mechanism (although the hardware may be physically shared) but by an encryption "subroutine" available to the PC. File encryption is in addition to, not in lieu of, any encryption performed for communications purposes.

The presence of the clear text bypass in this alternate approach implies that the red software discussed above is trusted to operate, if not correctly, then at least securely. The trusted red software would limit the use of the clear text bypass. The red software must be isolated from untrusted PC software, probably by using either separate processors or a security kernel.

It is recognized that there are advantages and disadvantages to any alternative approaches. I submit that because it recognizes the nature of modern network protocols and interfaces, the alternative approach outlined here is more likely to be viable as a general approach to network security.

KARL AUERBACH
System Development Corp.
Santa Monica, CA 90406

Author's Response:

I share many of Stephen Kent's reservations about the convenience of software versus the security of hardware. I return, however, to my starting premise: I was seeking methods whereby an unsophisticated user could connect his personal computer, by telephone, to a network and have credible assurances that none of his personal data could be transmitted in the clear to the central facility. Most such users will find a hand-held key more tangible than one said to exist in software but never seen. Most such users will trust manually set data paths more than invisible electronic paths inside the machine. There is no question that my proposal limits the flexibility of the machine. It was precisely my intent to limit the flexibility, for I believe that therein lies the key to secure personal computing.

Kent may be correct that my scheme would be unsuitable for a computer network capable of extremely high bit rates, whose file transfer protocols permit accessing distant files as if they were close at hand, and whose computers could gate data rapidly among a collection of opened, distant files. I was not addressing that class of applications.

Let me elaborate on my basic premise—that users can control their levels of security without relying on the security of the central facility, of the network, or of borrowed software. I believe that hardware keys and a separate encryption unit are preferable to software keys. I defend this position for two reasons. First, as noted by Kent, certificates do not provide absolutely secure exchanges of public keys. Now, if public keys are engraved on a physical medium, such as magnetic stripe cards, I (or my associates) can plug my key into any computer equipped with the appropriate encryption device without worrying about forgery. This may well be cumbersome if I wish to communicate with a large number of other users or with users I do not see personally. However, I believe it is important to have the option of exchanging keys and carrying them around outside of the system. It is easier to do this with hard keys than with soft keys.

Second, it is necessary to store secret soft keys in the personal computers. If I wish to access the network from more than one personal computer, each must have a record of my secret key. I prefer a hard key that remains in my possession to a soft key, the distribution of which I cannot control with equal confidence.

I am grateful to Kent for noting my omission of the problem of authenticity in file storage—if I simply encipher a file with my public key before transmitting it to the central facility, its contents will be immune from disclosure but not from alteration. One solution is for me to sign the file with another secret key before enciphering it with my public key. On receipt back, I can decipher it with my secret key and then verify the correctness of the signature.

It is true that my proposed scheme does not preclude borrowed software from leaking information on overt channels if soft public keys are set by untrusted software. That is precisely my point; this leakage cannot happen with manually set hard public keys. It may be true that in the context Kent is considering, that manual setting of hardware keys is too cumbersome to be practically feasible. However, I do not believe that this is true in the context I meant to establish.

I disagree with Kent's conclusion that the network interface, which appends cleartext source and destination information to the packets of messages, would pose a serious threat. Users of public data networks, such as Telenet, which are appropriate for the application I describe, have no control over network protocols, which require this header information to have a fixed format.

Although the prime factors encryption algorithm developed by Rivest, Shamir, and Adleman runs several orders of magnitude slower than DES implementations, this is not an important factor in the class of applications I was addressing: enciphering and deciphering information transmitted over telephone lines.

In this context, the 5000 bps encryption rate of the prime factors algorithm is adequate.

Finally, as I stated in the paper, we can deal with lost (or stolen) secret keys in the same way we deal with lost (or stolen) credit cards: My liability is limited as soon as I report the loss.

With regard to the comments by Karl Auerbach, I did not mean to suggest that the cryptographic functions should be entirely separate from the protocol functions. I do believe, however, that these functions should be separated more than Auerbach advocates. In retrospect, it seems clear to me that the network protocols should not be implemented on a user's personal computer as I had suggested in my paper. Rather, they should be packaged in a separate network interface unit or provided via telephone, as with Telenet. The network interface unit would then connect both the user's personal computer and encryption device (provided he was using one) to the network. Network commands (e.g. a request to store a file at the central facility) would be transmitted in the clear from the user's personal computer to the network interface unit, bypassing the encryption device. Data, on the other hand, would always pass through the encryption device. Protocols are still necessary to govern the communication of cleartext commands and ciphertext data between the user's personal computer and the network interface. However, the problem of handling both cleartext commands and ciphertext data is confined to the user's system, where the necessary protocols should be considerably less complex than the network protocols.

DOROTHY E. DENNING
Purdue University
West Lafayette, IN 47907

On Statistical Analysis

□ As a statistician primarily involved in statistical computing I sometimes find myself questioning the quality of computing theory and practice presented in statistical journals. Now I must also question the statistical analyses reported in *Communications,* in particular the article "An Implementation of Structured Walk-Throughs in Teaching Cobol Programming" by Ronald S. Lemos in the June 1979 issue.

The article describes an experiment whereby one group of students critically read and debugged each other's programs in addition to attending lectures while a control group simply attended lectures. At the end of the term both groups were examined on their abilities to recall Cobol grammar, to critically read a program, and to write a program. The basis of all statistical techniques in comparing two groups like these is to compare the difference in the average responses from the groups to the variability within the groups. If the difference is small compared to the variability we think that it may have resulted simply from random fluctuations in the data but, if it is large, we think there is a systematic difference between the groups.

One method of comparison is to compute a t-statistic and compare this value to a fixed, critical value. In this case the t-statistics would be 1.80 for the grammar score, 4.84 for the reading score, and 3.31 for the writing score if we assume that variances can be pooled. (Without this assumption the values would be 1.83, 4.56, and 3.35, respectively, so there would be no change in the conclusions.) These can be calculated in a few minutes with a hand calculator from the data in Table IV on page 339. They are compared to critical values of 1.971 for a 5 percent level test or 2.599 for a 1 percent level test to reach the conclusion that there is insufficient evidence of a systematic difference in the grammar scores but very strong evidence of a systematic difference in both reading and writing scores. This is not unexpected. Students with more experience reading programs should be able to do it better and the fact that they will have seen more types of errors will help them to avoid such errors in their own programs.

The weak point in this straightforward analysis of the data is that we are making the implicit assumption that any systematic differences between the groups are caused by the difference in teaching methods. There may in fact be a bias in the choice of groups so that the brighter students were in the experimental groups and would have attained higher scores regardless of the teaching method. It is to try to avoid such a bias that the assignment of classes to experimental or control groups was randomized. Furthermore the students were tested at the beginning of the term on their innate ability and their learning ability and this information can be used to compensate for any apparent bias by the technique of the analysis of covariance. This simply adjusts the average responses as shown in Tables V and VI after which an F-test is performed. (This is equivalent to a t-test on the adjusted data since the square of the t-statistic is the corresponding F-statistic.) The conclusion from the adjusted data is the same as from the unadjusted data: insufficient evidence of a systematic difference in the grammar scores but strong evidence of a systematic difference in the writing scores.

However, Lemos reaches a different conclusion about the reading scores. He states that none of the covariates (measures of innate ability) appeared to be affecting the reading responses so the analysis of covariance is inappropriate and therefore nothing can be concluded from the data. I agree that the analysis of covariance does not apply in this case but this only means that there is no need to compensate for differences in innate ability. The differences between the groups can be safely assumed to be due to differences in teaching methods and the conclusions from the straightforward t-test accepted. Again, this is not unexpected. Students with experience reading each others programs can do it better than those without the experience. Of course, there is still the implicit assumption that there are no other covariates which are affecting

the response but this is implicit in Lemos' other conclusions as well.

Lemos is to be congratulated for his careful analysis of the grammar and writing scores but, with regard to the reading scores, he went astray. There is no need to abandon the data and say that no conclusions can be drawn from it simply because the straightforward analysis is sufficient and the sophisticated analysis is unnecessary.

DOUG BATES
University of Alberta
Edmonton, Canada T6G 2G1

Author's Response:

The letter from Doug Bates regarding my paper, "An Implementation of Structured Walk-throughs in Teaching Cobol Programming" focuses upon a very important (and much misunderstood) issue in the design of experiments in "real world" situations. Given a specific experimental setting, what is the most appropriate research design to analyze the data? Bates supports the analysis performed on the grammar and writing data, but questions my decision not to make an inference on the apparent differences in reading scores between the two treatment groups. Essentially, he is suggesting that these data (including writing and grammar scores) could have been analyzed with simple *t*-tests (he is even so kind as to calculate these values). Unfortunately, this approach would have been entirely inappropriate for this particular research design, even though *t*-tests are quite frequently misused in statistical, computing, and educational journals.

The simple *t*-test is only appropriate where subjects have been randomly assigned to treatment groups. This is the only means of experimentally (as opposed to statistically) controlling for pre-existing differences between groups. Otherwise, one could erroneously infer that differences in the experimental outcomes were due solely to the treatment introduced in the experiment. Without randomization, the data are subject to many alternative hypotheses that could explain experimental outcomes.

Table I

	Experimental Group ($n=3$)		Control Group ($n=4$)			
	Mean	SD	Mean	SD	t	probability
Reading Score	115.47	25.69	86.32	13.22	1.99	.10 not sig.

This experiment was clearly not characterized by random assignment of the 215 subjects (students) to the two treatment groups. It was the seven class sections that were (randomly) assigned to the treatment groups. As clearly stated in the article, this is what distinguishes a quasi-experimental design from a true experimental design. The only way I could have correctly used the *t*-test would have been to designate the seven class sections as the units of analysis. I would have then needed no covariates since the classroom is now considered the subject and randomization has theoretically taken care of initial group differences. However, my sample size would have been reduced to seven. This would make for a better experimental design but a weaker experiment. The table above reflects this problem when the data are analyzed as a true experimental design using a *t*-test.

As can be seen, the result is not significant at the .05 level. While the population variance has decreased, the small sample size affects the power of the test. Clearly, Bates' suggestion that "The differences between the groups can be safely assumed to be due to differences in teaching methods and the conclusions from the straightforward *t*-test accepted" is not at all appropriate in the research design that was used. The biggest problem in empirical studies is that researchers often make inferences about data even when the underlying research design is faulty or they have misapplied a particular statistical technique.

Bates has also obviously misread me when he claims that my paper "states that none of the covariates (measures of innate ability) appeared to be affecting the reading responses so the analysis of covariance is inappropriate and therefore nothing can be concluded from the data." What the paper states is that none of the covariates "proved to satisfy all necessary requirements for the statistical model" (p. 339). Thus the effects did exist, but could not be incorporated into the statistical model. When this situation arises, it is the researchers' obligation to refrain from making spurious inferences.

I feel that Bates' letter has focused on a very important issue in the design of experiments. There is a general lack of understanding on the application of statistical tools to "real world" situations. In school, the texts and instructors always assume random assignment to treatment groups. Of course this is important to describe validly the statistical models, but most people do not really understand the consequences of relaxing the assumption of randomnness in experimental designs. While the true experimental design is definitely preferable, I feel that a familiarity with quasi-experimental designs gives one an appreciation of the strengths and weaknesses of statistical models and provides a valuable tool for experimentation involving human subjects grouped into naturally assembled collectives such as classrooms, work groups, or departments.

RONALD S. LEMOS
California State University
Los Angeles, CA 90032

Correction. In the list of Contributors to the C³S Report which constitutes the Appendix of "Curriculum '78," published in *Communications*, March 1979, the name of John T. Gorgone, Bentley College, was inadvertently left out. The omission is sincerely regretted.