Limited office copy

ON THE FEASIBILITY OF CONNECTING RECON TO AN EXTERNAL NETWORK

James P. Anderson Co.
Box 42 Fort Washington, Pa. 19034

JA

Consultants in Computer Technology

ACKNOWLEDGEMENT

The idea of using a cryptographic checksum to authenticate a decision regarding releasability was originally conceived by Lt. Col. Roger R. Schell, USAF, in connection with another application. Lt. Col. Schell was consulted on the conceptual development of the approach. The contribution of Mr. Charles Kellum, ORD/ISRD, to the implementation method of permitting data records to "belong" to two or more use groups and in providing the functional hardware design examples in the appendices is cheerfully acknowledged.

TABLE OF CONTENTS

		Page
1.	INTRODUCTION	1
2.	THE RECON SYSTEM	2
3.	THE RECON SECURITY PROBLEM	4
3.1	The RECON Security Environment	4
3.2	Operative Aspects of the Security Problem	5
4.	COINS	7
5.	SUMMARY OF PREVIOUSLY CONSIDERED APPROACHES TO THE PROBLEM	10
5.1	Separate Systems	10
5.2	Multi-Level Secure Operating Systems	10
5.3	Filters	11
6.	AUTHENTICATED RELEASABILITY	12
6.1	Technical Approach	12
6.2	Concept of Operation	12
6.3	Properties of the Cryptographic Checksum	13
6.4	Security Properties of the Approach	17
6.5	Security-Derived Modifications to RECON	19
6.5.1	The Need for Batch Access for External Network Users	20
6.5.2	Implementing "Batch RECON"	21
6.5.3	Whole RECON Records	22
6.5.4	Implementing Filters in RECON	23
6.5.5	Bandwidth of Covert Signaling Using Error Messages	23
6.6	Extensions of the Authenticated Releasability Concept	25
6.6.1	Summary of Key Notions	25
6.6.2	Handling Multiple Protected Categories	25
7.	OPERATIONAL IMPACT	35
7.1	RECON Operations	35
7.2	Increased Data Storage	38

Page

Appendix A - COINS Security Summary A-1

Appendix B - Adding Filters to RECON B-1

Appendix C - GUARD Implementation Example C-1

Appendix D - Application of GUARD to SAFE D-1

Appendix E - Comparison of Alternatives to Solving RECON Security Problem E-1

Illustrations

Figure 1 - COINS II Ring Architecture Concept 8

Figure 2 - Technical Approach to RECON Security Problem 14

Figure 3 - Block Chaining 16

Figure 4 - Symbolic Representative of Basic Capability 26

Figure 5 - Two or More (Disjoint) Categories 28

Figure 6 - Consolidated GUARD Functions 30

Figure 7 - Controlling Access to Two or More Groups 31

Figure 8 - Controlling Hierarchical Dissemination 33

Figure 9 - Configuration of GUARD and Checksum Generators 36

Figure 10 - On-Line GUARD Detail C-2

Figure 11 - Update GUARD Detail C-3

Figure 12 - GUARD Handling Multiple Dissemination Categories C-5

Figure 13 - SAFE (GUARD) ICC Configuration D-2

Figure 14 - SAFE (GUARD) WBC Configuration D-3

1. INTRODUCTION

This report describes the results of a feasibility study of an approach to solving the security problems associated with attaching the RECON bibliographic system to an external network. The problems were surfaced in considering the attachment of RECON to the COINS network in order to extend the services of RECON to the Intelligence Community as a whole. While the study has concentrated on the technical aspects of the problem, it has addressed some of the procedural aspects as well.

In the balance of this report, we will briefly review the RECON application, identify the security problem, discuss the COINS network, review other approaches considered, and then describe the recommended approach.

2. THE RECON SYSTEM

RECON is an on-line interactive bibliographic reference system maintained and operated by the sponsor at his headquarters. Its host computer is a 370/168 system which is one element of the Ruffing Computer Center complex (RCC). The RECON data base is a subject file index of intelligence reports from all over the Community. The data base contains citations for both raw and finished intelligence reports including collateral and SCI.

The RECON system is complemented by an in-house Automated Document Storage and Retrieval System, ADSTAR, which stores source documents in digitized form on microfilm. RECON currently serves approximately 130 terminals in the sponsor's organization through two COMTEN front-ends.

The data base contains two kinds of records: collateral and SCI. A RECON user may specify which file(s) he wishes to search (collateral, SCI, or All (meaning both)).

The RECON user interacts with the application through 20 commands, one of which is an implied SEARCH. The RECON implied SEARCH command produces a set of records that meet the search criteria. The result sets are associated with a user's work space and can be combined or limited in various ways after a search has taken place. It is possible to combine the results in two or more sets through logical operations (e.g., one can create a set (1) on KW/BIRD and another set (2) on KW/SEED, then logically combine the sets 1 AND 2 instead of having to specify that intention in the initial search as (KW/BIRD AND KW/SEED).

3. THE RECON SECURITY PROBLEM

3.1 The RECON Security Environment

In the RECON files, there are broadly speaking two kinds of titles, those which can be widely distributed and those whose distribution is restricted because they are compartmented, proprietary, or originator-controlled.

The type of distribution accorded to the "restricted" group is complex because of a desire to avoid the absurdities that can arise from an originator being denied access to a title that he created and contributed to the system under an originator-controlled label because the system applies a rule preventing distribution of originator-controlled titles. Thus, the problem cannot be solved merely by denying all external access to the restricted group of titles.

Presently in RECON, access is controlled to:

a. The RECON application (via logon-id and password).

b. The collateral or codeword (sub)files (via authority presumably contained in the user's identification record).

c. Modify and/or update commands (via authority contained in a separate SECURITY data set).

No other access control is provided.

RECON has the ability to manipulate sets to create combined sets which may then be edited to print records or any selected fields of a RECON record.

The RECON application is interactive, although an overnight batch and a canned query capability exists.

3.2 Operative Aspects of the Security Problem

Due to limited resources, the sponsor makes no attempt to validate manufacturer-supplied changes to the operating systems (MVS, VM, JES3) or vendor-supplied software packages. As a consequence, it must be prudently assumed that trapdoors exist in some or much of the sponsor's software or operating systems. The degree of threat this poses is a function of how much trust one has in the user population and the accessibility of the systems.

The sponsor has what appears to be unlimited trust in its own personnel due in part to the high standards established for clearance and a program for updating personnel investigations at nominal five-year intervals.

In connection with making RECON available to Community personnel, the sponsor correctly asserts that not all Community organizations apply the same clearance standards for access to SCI. In particular, the Military Departments apply substantially different investigative standards for clearances and SCI access. The Military Departments do not require an extended background investigation of its personnel for Top Secret clearance, and they do not use polygraph examination to verify background and investigative information about an individual.

Because of this, the sponsor asserts that the reliability of that segment of the Community is unknown and that the sponsor cannot fulfill its obligations to protect highly classified and sensitive information by giving unrestricted access to one of its systems to Community personnel.

With regard to the accessibility issue, it is obvious that putting RECON on a Community network increases its accessibility. What is less obvious is that RECON is on an internal network of substantial dimensions. If RECON is compromised to permit manipulation of the system upon which it resides, the compromise could be used to compromise the entire sponsor's internal network.

Thus, the sponsor's concerns are twofold. First, if RECON becomes accessible to user population of essentially unknown reliability, it could be potentially subject to external penetration by activating a trapdoor in the RECON application, or the underlying operating system that could be used to recover information, manipulate information, or deny service to RECON or other parts of the sponsor's internal network. Second, there is a corresponding concern that hardware or software failure in the sponsor's internal network would increase the risk of accidental exposure of sensitive information due to spillage on the external network.

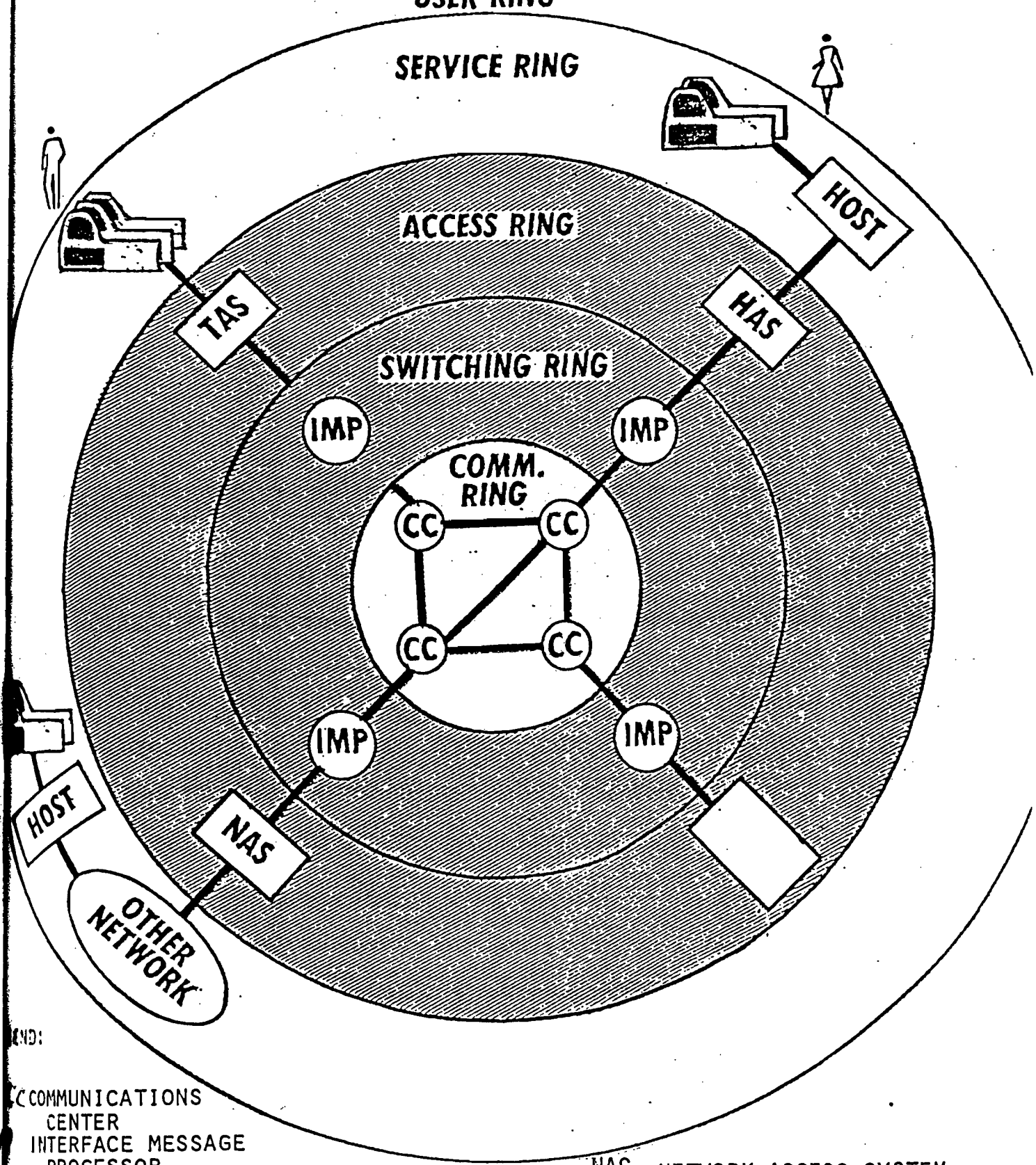
4. COINS

The COINS network has the structure shown in Figure 1. The packet switched nodes are ARPANET IMPS, interconnected by full period link-encrypted channels of the Tetrahedron communications network. At the next level, a series of interface processors is found that interfaces terminal users, server-hosts, and other networks to COINS. The interface processors, known collectively as CAS's (for COINS Access Systems), are PDP-11's running UNIX and special network software. All CAS's have network protocol software (currently NCP), software to perform logging of CAS activity for network management purposes, and access control software.

Depending on the principal function of the CAS (terminal/user support, server-host interface, network interface), function-specific software is also found. In the case of Host Access Systems (HAS), the HAS's host interface is further specialized to interface to the specific hardware of the server-host. (It will emulate the environment expected by the server-host (e.g., a channel of a specific transfer rate, etc., and the minimal protocols needed to coordinate transfer of information between systems.)

COINS users interface to COINS from terminals, either through a user's local host or through a Terminal Access System (TAS). COINS users have a restricted functionality both with respect to the data bases they query and with respect to the TAS's and hosts through which they access the network. With respect to the data bases, COINS users cannot do more than request data. A COINS-based analyst can frame queries either in the native language of the query system or, if he or she accesses COINS through a TAS, the queries will be able to be stated in a network virtual query language known as ADAPT. The ADAPT query is translated into the native language of the target system

COINS II
RING ARCHITECTURE CONCEPT



CC COMMUNICATIONS CENTER
IMP INTERFACE MESSAGE PROCESSOR
TAS TERMINAL ACCESS SYSTEM
HAS HOST ACCESS SYSTEM
NAS NETWORK ACCESS SYSTEM
COINS PMO ZONE OF CONTROL
TERMINALS

Figure 1

for the user. Once at the target system application, a COINS user's query is treated just the same as any other query; it is interpreted by the application's software to produce the requested data. The COINS user cannot affect the software or the data base. He is unable to change anything in the server-host system since the server-host is interpreting the request.

All analysts and their terminals in COINS are cleared Top Secret SI/TK as are all computer sites. The network operates in a Top Secret SI/TK System High mode (as defined in DCID 1/16).

The access systems provide the capability of enforcing NTK in two forms. First, an agency may wish to restrict access by some of its personnel to some of the data on the COINS (or other attached) network. It can do so by omitting the access privilege when the user is made known to the CAS. Second, access systems can control access to applications or hosts that they support through use of a host-agency supplied access list identifying by name those users who may execute the application or access the host. The HAS can also (if directed by the server-host/agency) grant access by Agency or other smaller organizational groups.

Additional detail about COINS security can be found in Appendix A.

5. SUMMARY OF PREVIOUSLY CONSIDERED APPROACHES TO THE PROBLEM

A number of approaches have been previously advanced for solving or avoiding the RECON security problem. This section will review them.

5.1 Separate Systems

In an earlier examination of the problem, it was proposed by the sponsor that a separate computer system be provided to store and make accessible to the Community those bibliographic entries not deemed "special" as discussed above. Several subsets of this approach were considered; however, the approach was rejected because of the cost of maintaining duplicate facilities. The approach protected the sponsor's assets from penetration and exposed that portion of the data base, even if the system were penetrated, only to individuals who would be authorized to access the information under any circumstances.

5.2 Multi-Level Secure Operating Systems

As a way of defeating internal penetration by programming users, the notion of applying multi-level secure operating system technology (e.g., KSOS) to the host supporting RECON was considered.

This approach, in principle, would go far to defeat direct attacks and, if the software change controls proposed to assure the continued security properties of such systems were in place, it would defeat the placement of trapdoors and Trojan Horses. Note, however, that if a trapdoor were placed in KSOS, it would be vulnerable to external attack in the same way as the existing RECON system.

However, the realities of the technology are such that KSOS cannot currently be applied to large existing systems such as 370/158's without changing the operating system and programming interface to produce totally incompatible (with anything!) systems. Coupled with costs estimated in the millions, the approach is not feasible in this environment.

5.3 Filters

The addition of filters to the RECON system software has been examined as a means of using the inherent capabilities of RECON to limit access to just those records deemed releasable based on security, dissemination, and codeword codes located in the RECON records.

If the classification, codeword, and dissemination codes are used in combination to identify material that is not to be released to external (i.e., network) users, the preferred approach is to (invisibly to the user) apply a filter consisting of a series of AND NOT < dissemination codes and codeword codes > to each (implied) SEARCH command issued by a user to exclude restricted material from the search. (A similar scheme involving canned queries is currently used by OCR personnel who now perform RECON searches for the Community.)

The filter approach provides the granularity of access control needed to restrict access to the subset of the RECON data base considered releasable; it does nothing to control the threats of internal or external penetration. Nevertheless, the application of filters to queries originating from network users will greatly simplify the design and operation of the GUARD system discussed below. An approach to implementing filters for RECON is outlined in Appendix B.

6. AUTHENTICATED RELEASABILITY

6.1 Technical Approach

At one level, the RECON security problem is akin to the problem of "sanitizing" SCI in order to release it to activities without the proper clearances. The general approach in sanitizing systems is to permit arbitrary queries by all users, but to route the results of the queries of uncleared users to a sanitization officer who would manually examine the output before releasing it.

In low-volume situations, all sanitization officer activity could be manual. In higher-volume situations, the use of a computer-based GUARD station to support the sanitization officer (WOO79) has been developed.

While the sanitization officer/GUARD station approach would work in principle, it is not a practical solution for RECON because of the excessive delays that would be imposed by the sanitizing officer. These delays would cascade to produce response times that border on the infinite.

What is proposed to solve the RECON problem is to adopt the idea of a GUARD station, but to automate the identification of releasable citations to minimize the bottlenecks cited above.

6.2 Concept of Operation

For the purpose of exposition, we will first consider all citations in RECON categorized either as releasable (to external network users who do not possess access approvals for the "special" citations) or NOT releasable to those individuals. For each RECON entry designated by the originator as

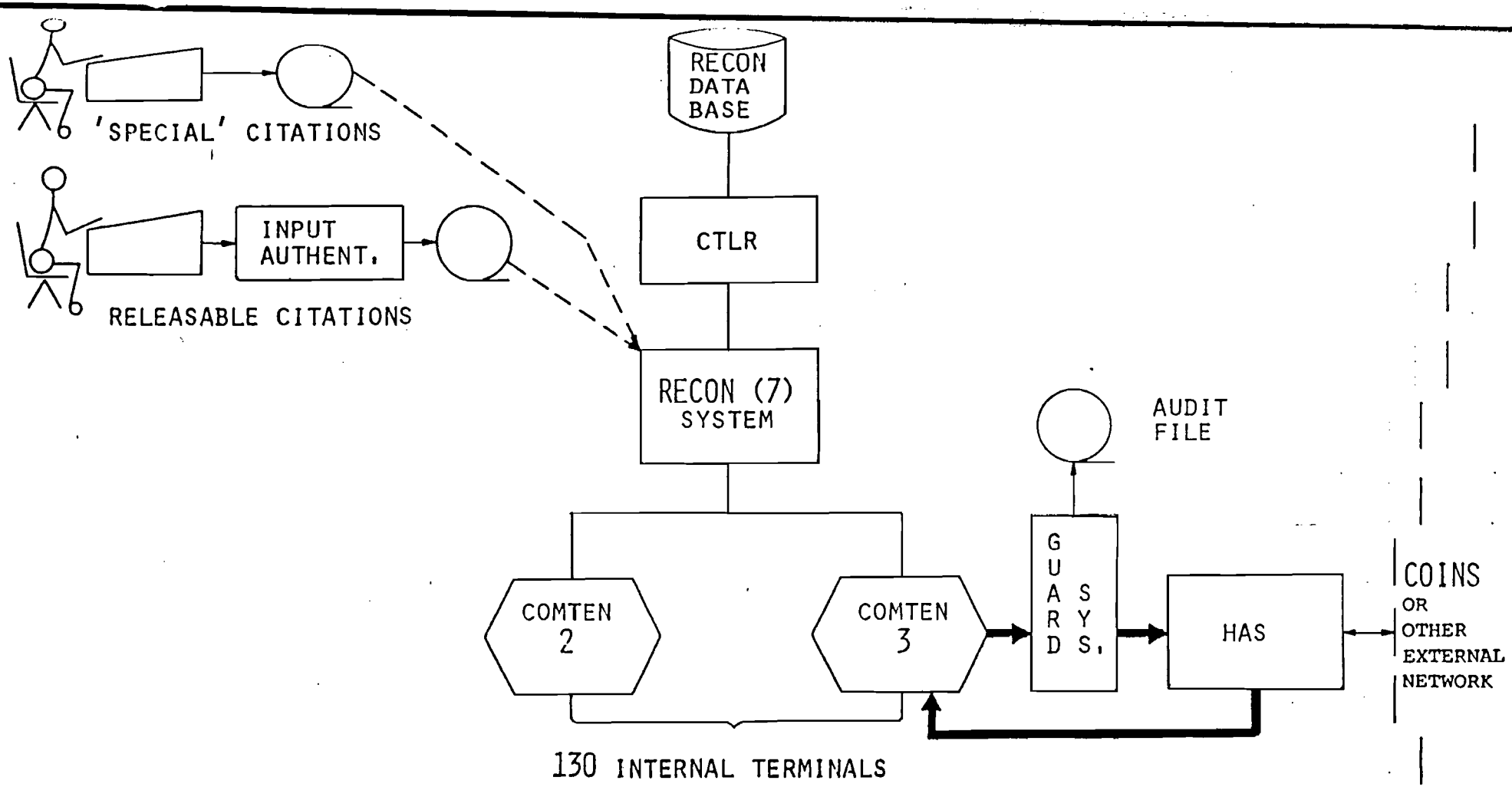
releasable to external users, a cryptographic checksum, which is a function sf the entire record, is computed by a special authentication device as the data is entered into the system. The checksum is appended to the record and stays with the record forever.

Upon being selected for output, all records for a specific destination are routed to a dedicated system (the GUARD processor) where the cryptographic checksum is recomputed. If the recomputed value is identical to the checksum appended to the record when it entered the data base, the entry can be released without further review. If the checksum check fails, the item will not be forwarded to the requestor and the record, destination, etc., will be written to an audit file.

Each RECON entry designated as "releasable" (i.e., NOT "special") will be processed through one of a set of input terminals that cause the entry to be routed through the input checksum generation device (see Figure 2) as part of preparing it for entry into the data base. The input checksum generator computes a unique, non-forgeable checksum which is appended to the entry before it is entered into the RECON system. If the entry and its checksum are subsequently forwarded to the GUARD interface for release, the checksum value is recomputed at the GUARD.

6.3 Properties of the Cryptographic Checksum

The principal problem that this approach raises is assuring that the checksum cannot be forged. This is solved by using a modern cryptographic techn6que, the National Bureau of Standards Data Encryption Standard (DES), and performing the checksum function outside of the RECON host on dedicated systems, one for computing checksums on input entries, the other for the



TECHNICAL APPROACH
TO RECON SECURITY PROBLEM

Figure 2

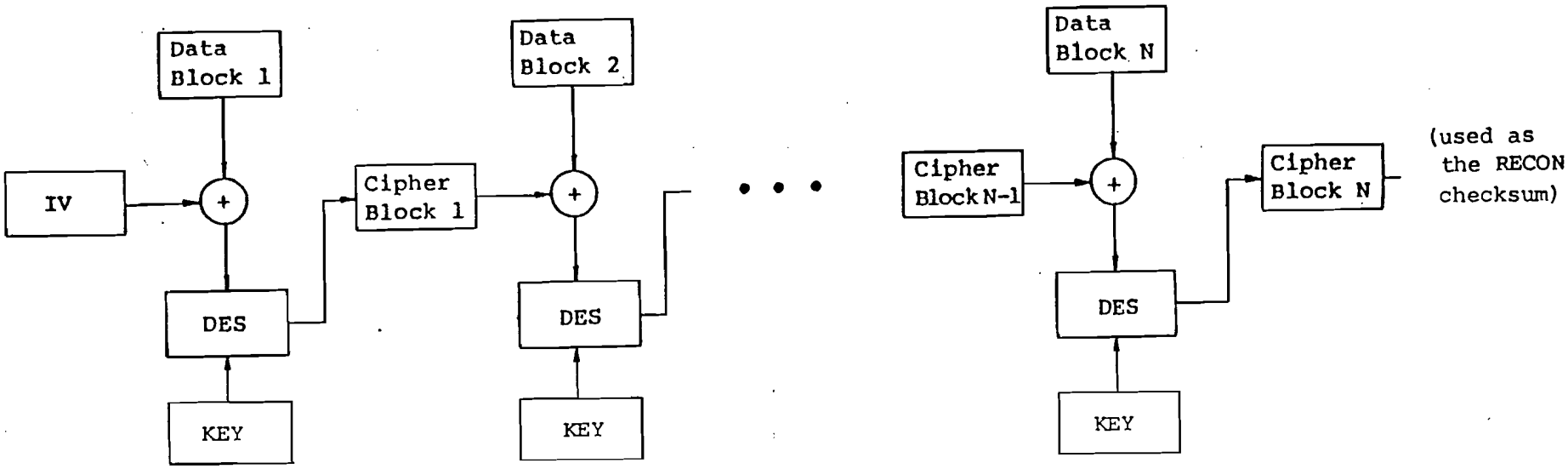
GUARD function.

The crypto checksum of the original entry is produced using a secret key known only to the input checksum device and to the GUARD interface processor(s). The key is NEVER available within the RECON system per se.

The crypto checksum is produced by block-chained encipherment of the releasable entry (see Figure 3). In block-chained encipherment, the ciphertext of each block of the item being enciphered is dependent on the contents of all previous blocks. The last block of an item is dependent on the entire entry and is used as the checksum. The secret key for this mode of use is 120 bits long (64 bits for the IV (the Initial Variable, used to provide the first value exclusive OR'd to the first block of plain text), 56 bits for the DES key).

The DES has a particularly attractive property for use within an application. As little as a one-bit change of the data being encrypted (or decrypted) will result in approximately 50% of the ciphertext (plaintext) bits being changed. This provides an excellent error (or tampering) detection quality to the scheme.

With the cryptographic checksum keys physically isolated from RECON and other RCC computers, the only other method of forging a checksum is to pick a 64-bit number at random and attach it to a RECON record. The probability of picking a "correct" checksum by accident is $\frac{1}{2^{64}}$ (the size of the checksum) or 5.24×10^{-20}.



- (+) Exclusive OR
- The secret key(s) are the Initial Variable (IV) and the KEY.

BLOCK CHAINING

Figure 3

In summary, the protection against forgery is provided by protecting the key. Key protection is provided by:

a. Physically separating the input and GUARD authentication machines from the retrieval processor.

b. Hard-wiring the key on the DES-board so that it is not even readable by the GUARD or input checksum generator.

c. Providing a security "kernel" in the GUARD and input checksum generator to control their operation. Because of the single-function nature of these devices, this kernel is simple in structure and offers no technical risk.

6.4 Security Properties of the Approach

Assuming that the GUARD system works as described and it is interposed between RECON (and its internal network) and an external network as shown in Figure 2, this approach has several interesting properties with respect to the RECON security problem. First, no failure or compromise of hardware or programs in the RECON host or its network will permit data to spill from the Agency internal network to the Community network. Second, no manipulation of RECON or its host processor (or the internal network) will release special or other material across the GUARD interface.

This is because the GUARD system will be designed to only deal with what it thinks are RECON records, and to escape the GUARD, a cryptographic checksum is recomputed from the just to be released record. If this checksum does not identically match the checksum computed when the record entered the RECON application, the record does not get released.

If a checksum gets detached from its citation at any time subsequent to its creation, the only loss will be that the entry which was considered releasable will NOT be available to Community analysts. The scheme is fail-safe. It should also be obvious that the approach will not spill special data if a checksum gets attached to a "special" citation, or either the citation or checksum is manipulated by accident or design since the change of as little as one bit of either component will result in a different checksum being recomputed on output.

The approach does not directly address the threat of manipulating a system through activating trapdoors from external users to manipulate data or deny service. Manipulating data through an essentially one-way trapdoor (since the GUARD permits NO unauthorized data transmission out of RECON) is a question of how much detailed information it is assumed the manipulator has (or is able to get) from internal sources to guide his attempt at manipulation. Nevertheless, such manipulation will not cause the release of unauthorized data.

The threat of denial of service through activating trapdoors is more realistic since not too much has to be known about the target in order to foul it up. Since the threat cannot be countered by the GUARD approach, the only solution evident at this time is to shut down the external network connection when and if denial of service attempts are detected.

The single remaining security question is what happens if the GUARD system fails? While of itself no guarantee, the simple function of the GUARD will reduce the opportunity for design and implementation errors. In addition, advanced design techniques, such as formal specification, can be applied to

the GUARD design to further increase the confidence in the reliability of the system. Since it is envisioned that the GUARD will be implemented in an advanced microprocessor (see Appendix C), once the program has been thoroughly checked out, it can be placed in a Read-Only Memory (ROM).

If the hardware of the GUARD fails, it must fail in such a way as to bypass the recomputation of the cryptographic checksum. Even then, the only real risk is if the GUARD fails and RECON fails or is subverted at the same time. The interface between the GUARD and RECON is simple enough that there does not appear, at this point, any way in which the RECON system can induce a failure of the GUARD.

Finally, the GUARD can be designed in such a way as to permit RECON to test the correct operation of the GUARD by addressing various kinds of records to itself. The only records that should return are those whose checksum is correctly recomputed. (If there is a concern that a subverted RECON could use this facility to generate and test "random" checksums, it might be noted that it would take about 58,494 years to systematically try all possible 2^{64} checksums against a single record. At the rate of 10,000,000 trials/second (100 ns/trial) on average, one could expect to find the correct checksum in one-half the time, or 29,247 years.)

6.5 Security-Derived Modifications to RECON

It would be nice if the GUARD approach could be applied as is, with no effect on RECON, and while providing the same interface to all users. Unfortunately, due to the concern for covert (and direct) channels that would permit transmission of unauthorized data outward to an external network, this is not the case.

Because of these concerns, it appears that the following changes will be necessary in the user's interface to RECON (and to RECON itself).

a. The external network users will not have an interactive interface to RECON. They will only be able to supply fully specified batch queries (or query procedures using the QUERY command).

b. The external network users will not be able to select fields of the RECON record to be returned. Only the entire releasable RECON record will be returned to the external network user.

c. The filter capability (outlined in Appendix B) will be implemented for external network users.

6.5.1 The Need for Batch Access for External Network Users

The reason for eliminating interactive working with RECON for external network users is that there is no way to subject the informational messages generated by RECON to the same checksum test applied to data records because the message contents are dynamic (counts of the number of records, lists of "adjacent" keywords, and the like). If RECON were subverted, these messages could be replaced with unauthorized data for an external user-agent.

There is no safe way to permit such messages. As a consequence, it is concluded that running RECON in batch mode is the only feasible way to circumvent the problem.

The major problem found with error messages is that they are applied dynamically; i.e., as errors are encountered. With very few exceptions however, they do not contain dynamic data. Even in the cases where the error message repeats information (e.g., an accession number, file name), the error message could be recast to be static. Under these conditions, the error messages sent to external users could be checksummed to permit their transmission to external users. Coupled with some additional separate messages such as "OCCURRING IN FIRST COMMAND LINE," "OCCURRING IN SECOND COMMAND LINE," etc. (up to the maximum of 50 command lines RECON can store as a canned query), that can also be checksummed and sent with the substantive error message to help a user localize his error.

(An alternative considered earlier would be to replace the standared RECON error messages with about six generic messages, locate them in the GUARD, and permit RECON to request the transmission of from one to six of them to a designated external user. However, the additional complexity introduced to the GUARD, changes to RECON to map existing error message numbers into one of those approved, and the reduced help provided to users over the simpler changes outlined above have led to the rejection of that approach.)

6.5.2 Implementing "Batch RECON"

The first constraint does not appear to require any functional change to RECON to support. There is a "canned" query capability already built into RECON that could be used to execute a query sequence entered by an external network user. The external user would use the COINS TAS with or without ADAPT to create the RECON query. When the query is formed, he can

direct it to RECON just like he now does for other COINS batch applications. The query will be staged in the HAS interfacing RECON to the network.

When RECON accepts a query from the HAS, it passes the query as a canned query (probably in Query execute mode) to the canned query processor. If the filter recommendation is implemented, RECON may attach a filter to the query appropriate to the dissemination authorized to the particular user (or Agency, or group, etc.).

RECON will maintain a user control block for the external network connection(s) that will relate a query to a particular originator.

The exact interface between RECON and the GUARD has not been defined in this study. Whether each single RECON record that satisfies an external query will have the destination address attached and sent to the network, or whether the GUARD will be able to accept groups of records destined for a single user is a function, in part, of the amount of Random Access memory available in the GUARD, as well as whether there will be a requirement for the GUARD and HAS to multiplex responses from RECON in order to maintain throughput. It is sufficient for now to indicate that these issues will have to be resolved at the detailed design level.

6.5.3 Whole RECON Records

The change that delivers only whole RECON records to external network users can be handled administratively (by not telling external network users of the DISPLAY, TYPE, etc.), or by extending the use of the "SECURITY" file to i te commands not allowed to specified users. This

change is only to minimize the complexity of the GUARD. Working in COINS, the external users have editing and storage facilities available to them in the TAS's to provide selection and formatting of various data fields.

6.5.4 Implementing Filters in RECON

Implementing the filter approach outlined in Appendix B offers no particular technical challenge. The filter is important since it relieves the GUARD of having to handle unauthorized records except when there is a failure or error of some kind in the RECON system itself. Thus, the GUARD will act as a check on the correct operation of RECON rather than as a guarantor of its correct operation. This shift in perspective is important in minimizing the complexity of the GUARD.

6.5.5 Bandwidth of Covert Signaling Using Error Messages

By permitting any error messages at all to be returned to a designated user, there is created a covert signaling path from RECON to that user. In the absence of any additional constraints, it would be possible for a subverted RECON to pass the content of unauthorized RECON records to an external user as a binary stream merely by adopting the convention that one error message stands for a binary 1, and some other error message stands for a binary 0.

Since the method recommended of generating checksums for all error messages along with a number of messages to help localize errors is designed to permit virtually arbitrary error messages to be returned to external users, there is no way for the GUARD to easily determine whether the messages being

passed are legitimate or signals without increasing the awareness of the GUARD about its environment and, incidentally, increasing its complexity.

To determine the covert signaling rate if RECON were connected to COINS, measurements were made of the actual time to send messages of various lengths between two processes in different access systems. The measurements were taken under normal operating conditions in that both access systems were actively supporting normal network operations. These measurements approximate the signaling rate that would be available between RECON through its HAS to a process (file) on another TAS in the network.

The results of this quick study are shown in Table 1.

Message Length in Characters	Effective Character Transmission Rate (Char./Sec.)	Covert Signaling Rate Bits/Sec. (Err. Msg. = 100 Char.)
100	8.5	.085
1982	150	1.5
31777	1000	10.0

Effective Data Transmission Rates Between Access Systems

Table 1

While the table shows the effects of a relatively high fixed overhead to handle network and I/O protocols, what is of interest to us is the expected covert signaling rates under the assumption that the error messages will be 100 characters long (or shorter).

Since COINS does not accumulate traffic for a specific destination but rather forwards messages as received, the .085 bits/sec. covert signaling rate is on the order of what one could expect if the RECON error messages were used to encode unauthorized information. At this rate, a 300-character record would take approximately 7.8 hours to transmit.

Any further reduction(s) will require incorporating an error message rate detector in the GUARD to raise an alarm if error messages are being sent more frequently than a specified rate.

6.6 Extensions of the Authenticated Releasability Concept

6.6.1 Summary of Key Notions

The principal conceptual notion of the proposal outlined in this paper is the fact that the design of the GUARD is predicated on the concept of checking on the correct operation of the guarded resource. The GUARD is not made an integral part of the application. Because of this simplicity, it becomes easy to extend the concept to more realistic conditions than twice postulated at the beginning of this section to explain the concept.

6.6.2 Handling Multiple Protected Categories

In Figure 4, we represent the approach outlined in section 6.2. There is the superset A, corresponding to all RECON records, and the subset B, corresponding to those which can be released to external network users. The users of the system correspond to these two sets. There is no GUARD processor intervening between RECON and the users authorized access to all

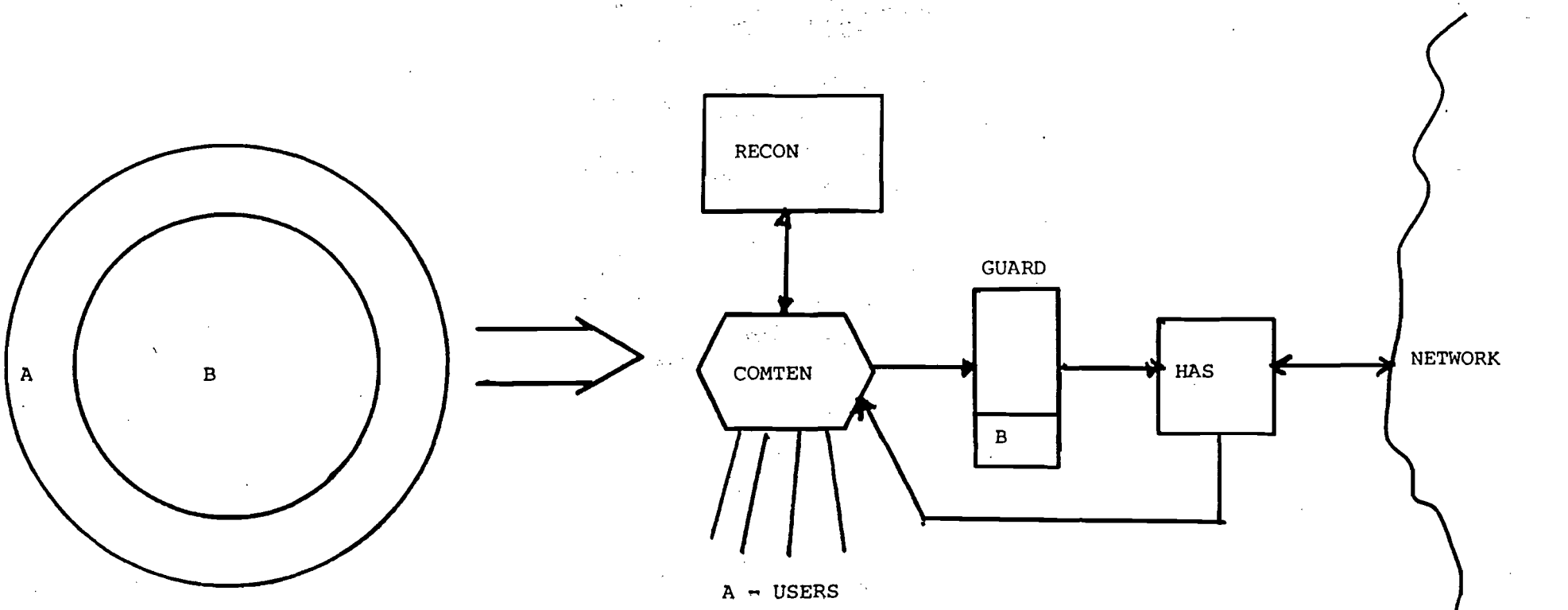


Figure 4

Symbolic Representation of Basic Capability

of the data base because it is not required. Symbolically, the key associated with the materials releasable by the GUARD is shown in the block representing that function.

In this section, we will show symbolically how various protection configurations can be handled by straightforward extension of the GUARD concept.

Figure 5 shows the arrangement of two (or more) disjoint categories (for a far-fetched example, limiting NSA access to just NSA-produced items, DIA to just DIA-produced items, etc.).

The role of the GUARD is still to check on the correct operation of RECON, which is supposed to route B category material to the B-GUARD and the C category material to the C-GUARD. The dotted end-to-end crypto boxes shown are optional depending on the degree of trust the various groups have in the correct functioning of the network. If the groups must protect their material from possible misroute to another, then the end-to-end crypto function will provide that protection. Note, however, that in the COINS network, the network is protected by full-period crypto on all links. The only protection that the end-to-end crypto would provide is against possible misaddressing in the HAS or misrouting in the COINS imps.

The categories B and C are distinguished on input in some fashion (in this example, by Agency of Origin), and the checksum computed for a record is keyed based on the information distinguishing the records. That is, B records are checksummed using the B key and C records are checksummed using the C key. The generation of the input checksums is not shown in the figures.

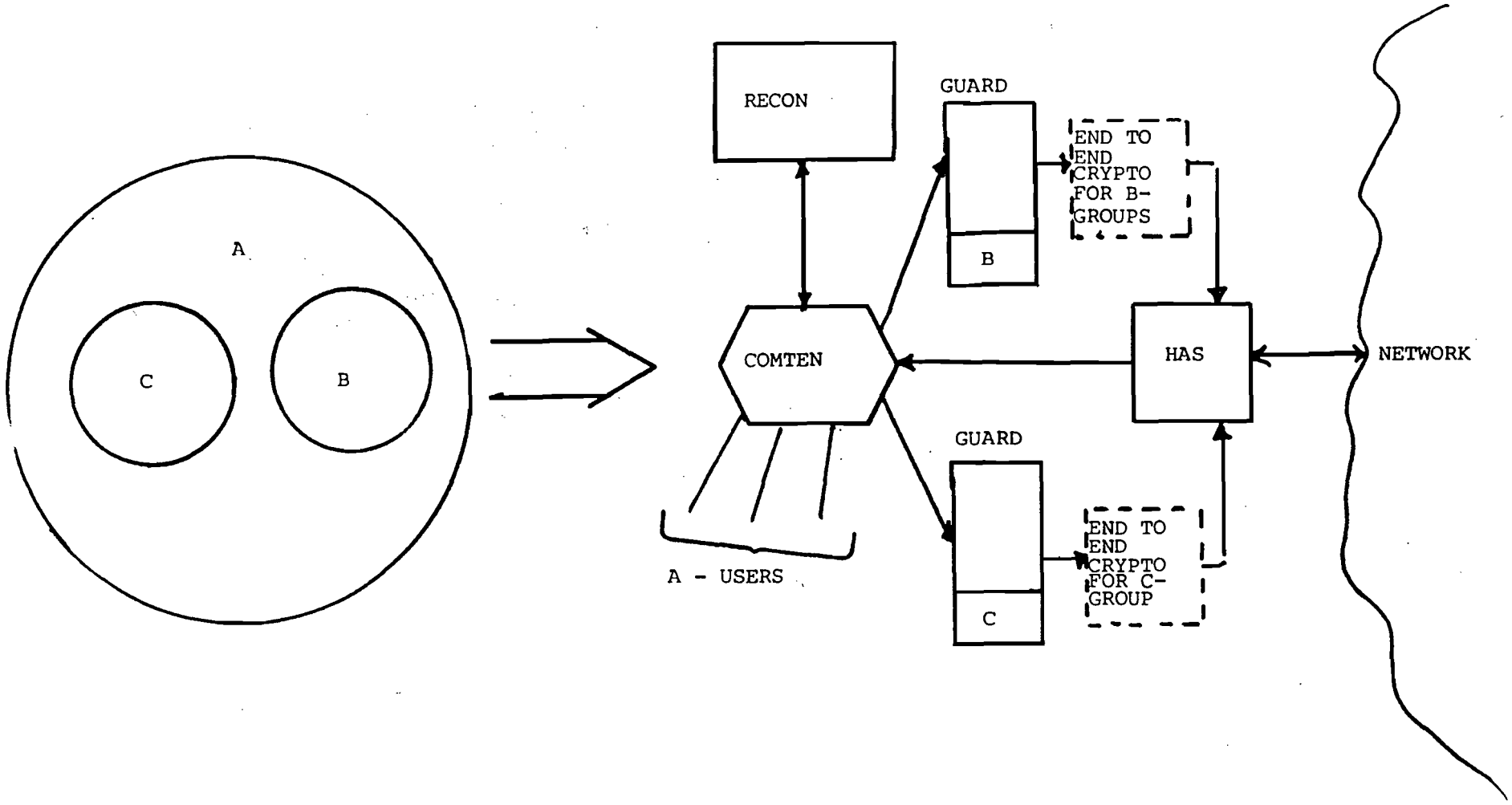


Figure 5

Two or More (Disjoint) Categories

The GUARD's are shown as separate boxes, one for each group or Community of interest involved. The cost of the GUARD is expected to be small enough that for application to the RECON-COINS case there can be a GUARD for each category (group of users).

One can imagine attempting to put all of the GUARD function in a single machine, using a group indicator of some kind to instruct the GUARD to select the appropriate key to farm the checksum, and if it checks, route the record through the appropriate end-to-end crypto function as shown in Figure 6. However, this raises the level of complexity of the GUARD by requiring it to discriminate correctly among the various categories and to not only perform the checksum correctly but to select the proper end-to-end channel as well. If the costs permit it, separate GUARD's are preferred.

Figure 7 shows the arrangement where two or more categories are involved with a superset shared. This form illustrates an arrangement that could give an Agency unrestricted access to the generally releasable material (B), as well as access to ORCON material produced by the Agency (group) making the request (e.g., C).

The degree of control this arrangement provides for ORCON material is a function of how fine a discrimination is possible among originating groups or Agencies. For instance, if the discriminant is based on Agency of Origin (e.g., DIA) and the fact that it is ORCON material, then it is possible to respond to requests from DIA and return all general records and DIA's (only) ORCON records. If DIA or the sponsor needs to break the access down further, it is necessary to further qualify the Agency of Origin (in this case) (e.g., DIA-MS4 or DIA-[illegible]1) in order to select the correct checksum key when the record is en[illegible]e system.

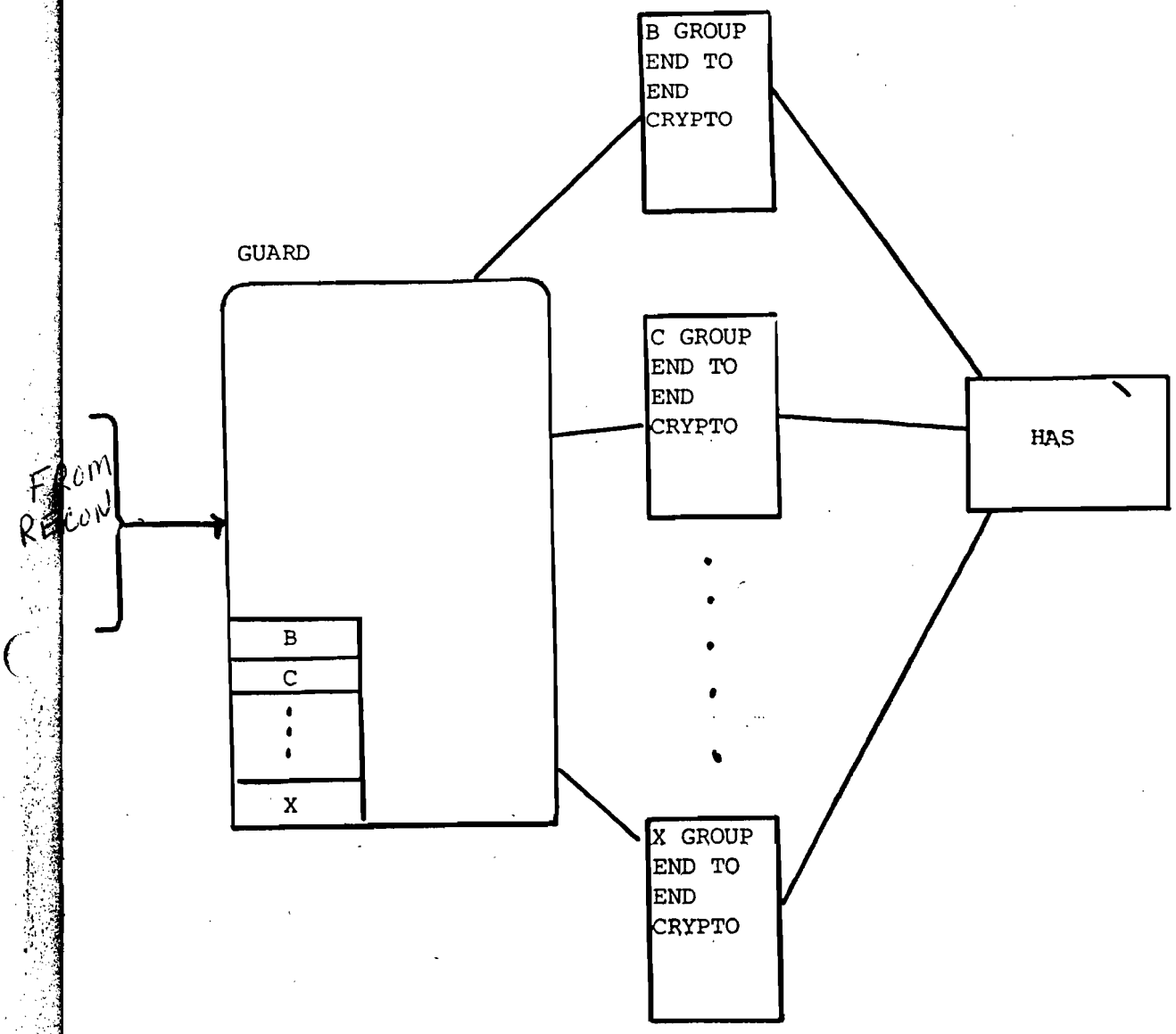


Figure 6

Consolidated Guard Functions

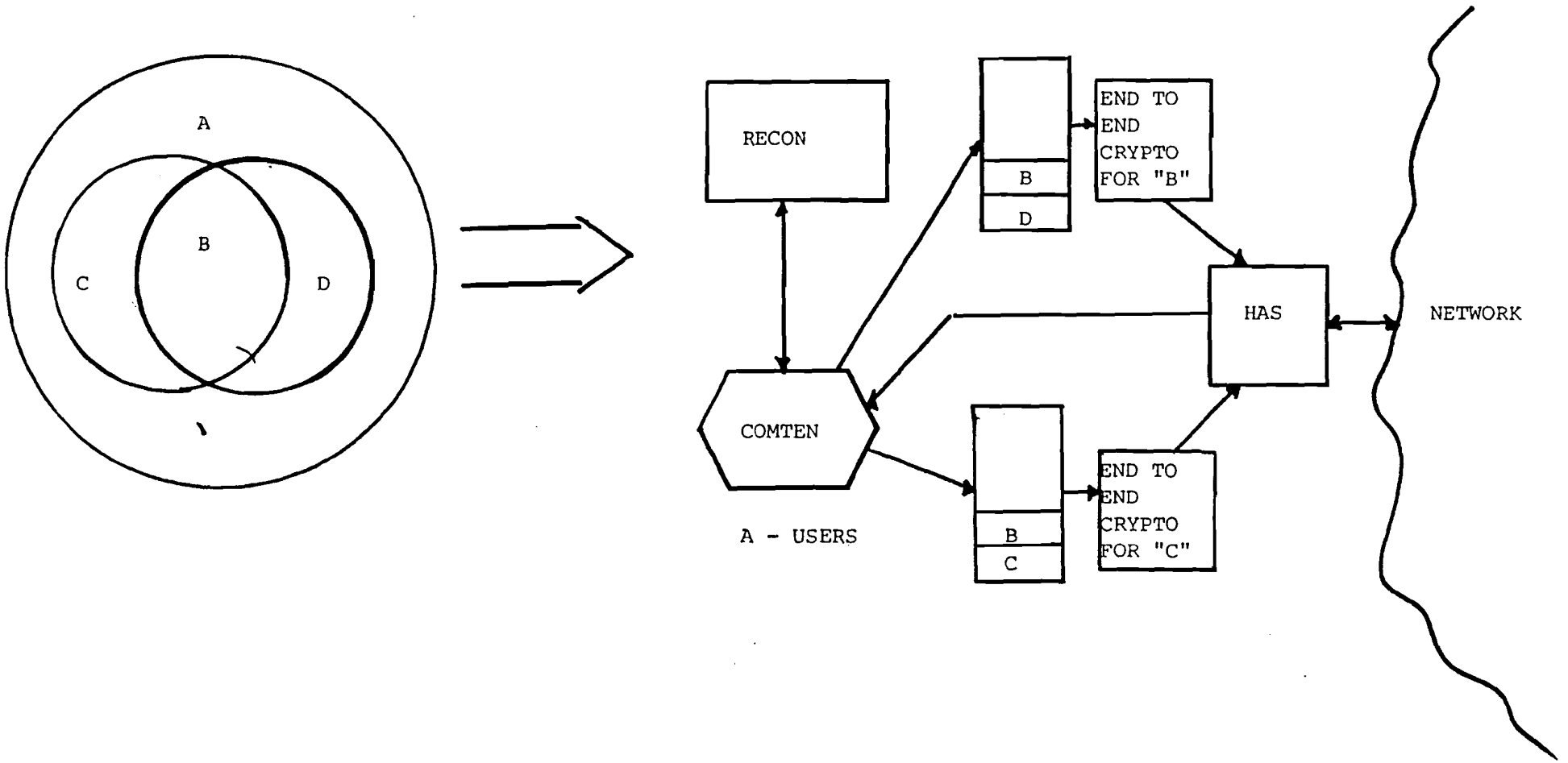


Figure 7

Controlling Access to Two or More Groups (Compartmentation)

Thus, each record in the system would still have a single checksum, the key for which is a function of the Agency of Origin only for otherwise restricted distribution material. That is, if the record being entered into RECON is generally releasable (based on classification, dissemination, and codeword values), it is checksummed using the B key (in the diagram). If the record being entered requires restricted distribution based on the classification, etc., but would be releasable to the Agency or group that originated it, it would be checksummed using the appropriate key (e.g., C).

All queries from external users are filtered as suggested in section 5.3. Ordinary external users would have a filter that permits access to B records only. Extended access users would have filters that permit access to B or C records, or B or D records, etc.

To get the appropriate (allowed) record set past the GUARD processor, checking the releasability will require a small change to the GUARD's functionality. In the previous examples, if the recomputed checksum did not match, the record was not transmitted and an alarm raised. In the present case, the GUARD is provided with a set of keys corresponding to the access categories permitted to the users associated with that GUARD. The GUARD computes the checksum with each key in turn until either an exact match is found or all keys have been tried. If all keys have been tried and no match results, the record is not transmitted and an alarm is raised.

Finally, Figure 8 shows how the scheme could be used to control dissemination based on the hierarchical classification scheme of Top Secret, Secret, Confidential, and Unclassified. This example is not shown in the context of RECON and COINS because COINS runs in a System-High mode and is

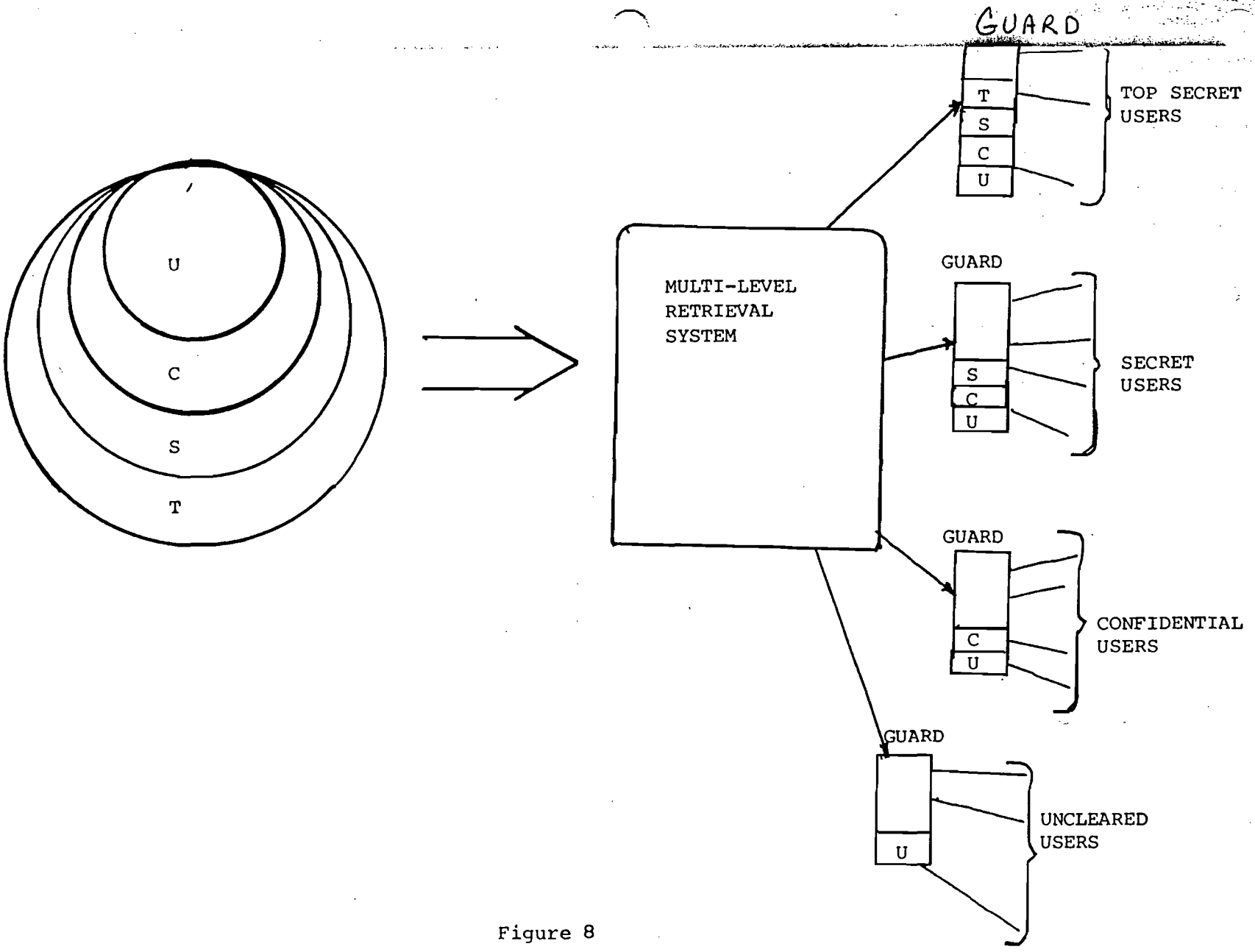


Figure 8

Controlling Hierarchical Dissemination

provided merely to illustrate the potential this technique has.

Here, records entering the retrieval system are checksummed with a key which is selected based on the classification of the record. Requests against the data base are filtered based on the clearance of the requestor. Thus, Top Secret users would be able to retrieve everything, while Uncleared users would be able to retrieve only Unclassified records.

As in the previous example, the GUARD's compute checksums until a match is found or all keys have been used.

This case raises a performance question that was not a factor in the earlier examples. In particular, if the GUARD would have to compute checksums through all stored keys until a match was found, it could reduce the throughput of the GUARD significantly. There are two possible approaches to this problem depending on its severity. First and simplest, the keys could be arranged in GUARD's in order of expected frequency of occurrence. Thus, if 80% of the data base is Unclassified, 10% Confidential, 8% Secret, and 2% Top Secret, the keys would be placed in the GUARD's so the Unclassified key was used first, then the Confidential key, and so forth.

If the checksum function really becomes a bottleneck in this mode of operation, it would be possible to compute two or more checksums in parallel by merely supplying additional DES cards. If the number of DES cards were less than the total number of checksums that had to be computed, the control for the additional parallel operation would be complicated slightly. However, it is not expected that it would be very difficult to design.

7. OPERATIONAL IMPACT

7.1 RECON Operations

The degree of operational impact this approach will have depends on how much the present automated hold file and RECON preprocessor are trusted to correctly extract classification, codewords, and dissemination codes from incoming electrical traffic.

RECON now operates this way with no problems mentioned. If the software which attaches these labels is trusted to do so properly, then a program can read the labels and divert the traffic into one or more (depending on the number of categories involved) subfiles for later computation of the input checksum. If such an arrangement is acceptable, then the impact on RECON indexing operations is nil, since RECON records will be handled by OCR personnel in exactly the same way they are now.

If all records are checksummed with a key appropriate to their releasability (i.e., releasable to network users, releasable only to sponsor personnel, etc.), and the checksum generator is placed in the RCC network as indicated in Figure 9, then if the checksum generator remains a dedicated single-function machine as described in the previous section, it could be programmed to automatically analyze the codeword, classification, and dissemination codes of each record, and select a key appropriate to the dissemination policy implied by the combination of values.

This arrangement depends critically only on the correct analysis and assignment of the classification, codeword, and dissemination codes for each record. If such assignments can be made correctly (as they apparently

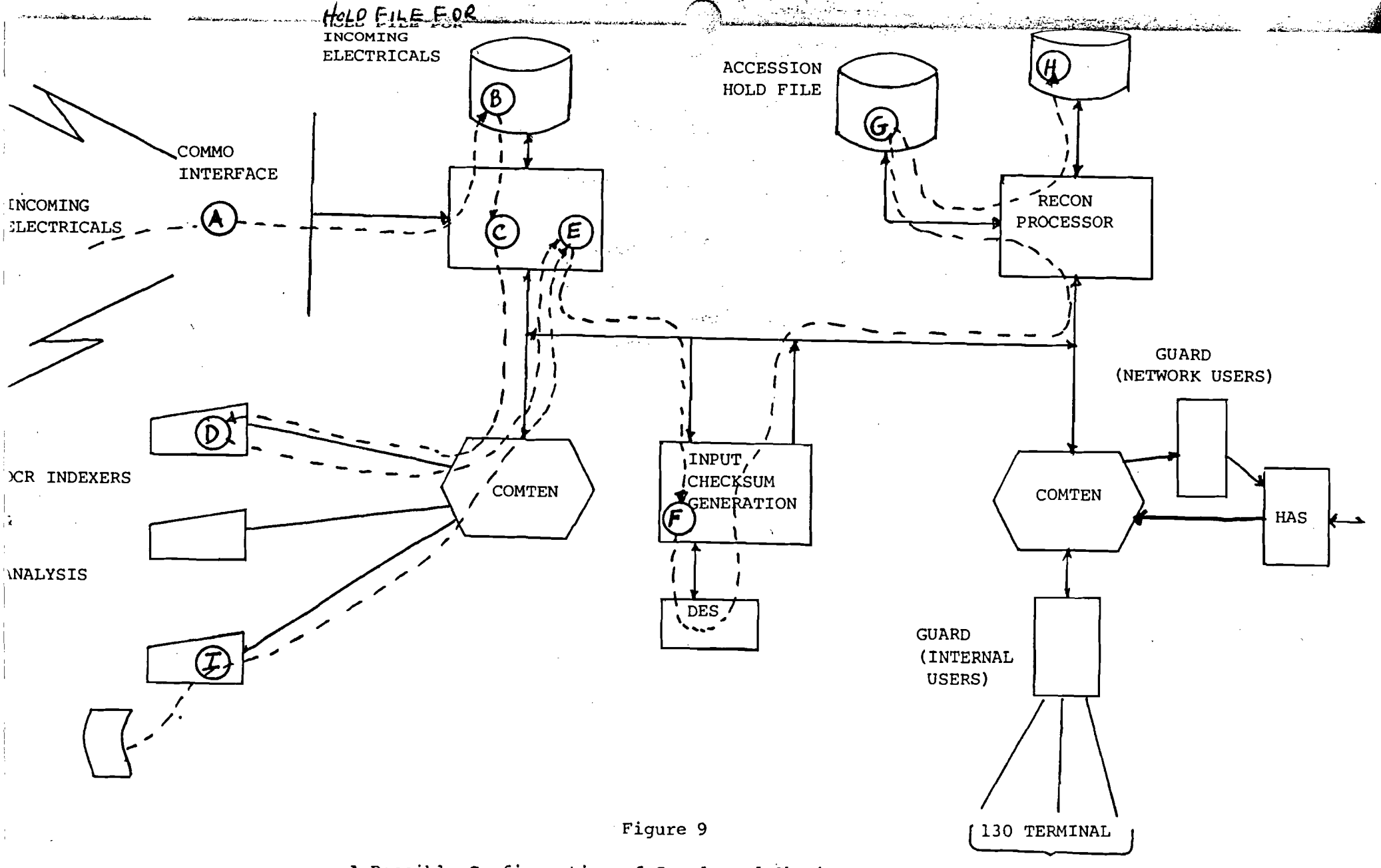


Figure 9

A Possible Configuration of Guards and Checksum Generators
to Minimize Impact of RECON Operations
Showing Flow of Data

are now), then the checksum generator and GUARD's can meet the protection objectives of the sponsor.

The data flows of the diagram are:

A. Incoming messages are processed by COMMO and released to the RCC network to be processed for RECON.

B. The incoming data is held in a buffer file for processing by OCR military personnel.

C. Prior to indexing, the message is scanned and a RECON record is partially filled out. The classification, codeword, and dissemination codes are entered in the skeleton record.

D. The skeleton RECON record is processed by the OCR analyst and subject codes, etc., are coded as the situation requires.

E. The record is released by the OCR analyst for routing to the checksum generator.

F. The record is released to the checksum generator where the classification, codeword, and dissemination codes are analyzed to select the proper checksum key.

G. The checksummed record is placed in a hold file.

H. The RECON data base is updated with the records in the accession hold file.

I. A RECON record is keyed in directly (from first principles) by an OCR analyst. It is routed to E., from which it follows the steps outlined above.

Note that if the checksum generator is bypassed by the RECON records processor, the affected records will not ever be released to either internal or external users.

7.2 Increased Data Storage

The checksum is eight bytes long. If the average size of a record is 150 characters or so, this will add about 5% to the storage required. If all 3.5 million RECON records currently had checksums, the total storage increase would be about 28 million bytes, or about 10% of a 3,350 disk.

REFERENCES AND BIBLIOGRAPHY

1. AND80 An Approach to Solving the RECON Security Problem,
J. P. Anderson, R. R. Schell
1 November 1980

2. KIN80 The < sponsor's > RECON System as a COINS Host,
H. A. Kinslow
3 March 1980

3. REC80 RECON IV, An ON-LINE Intelligence Information Retrieval System USERS Manual,
NFAC OCR/SSD
April 1980

4. WOO79 Applications for Multi-Level Secure Operating Systems,
John L. Woodward
1979 AFIPS Proceedings, pp. 319-328

APPENDIX A

COINS SECURITY SUMMARY

INTRODUCTION

Much of the present COINS access control functionality is placed in COINS Access Systems (CAS's). This appendix outlines the CAS security architecture, with the exposition being given in terms of a Terminal Access System (TAS). It should be noted that the security functions described for the TAS are available for the other CAS's as well.

The TAS architecture is responsive to the diverse and dynamic nature of the COINS network. It provides a coherent interface to server-host computers of different manufacture and to data base applications of widely varying design. It was conceived of as a means of insulating its users from much of the differences that exist in the different server-host systems and the data base query languages.

The TAS security architecture has been designed to provide maximum protection to the sensitive data in the network while keeping the end-user's interface as simple as possible.

In addition, the TAS security architecture has addressed the problem of security administration. It provides the user organizations with considerable flexibility in how the security is managed. It also allows a single TAS to support more than one organization, each of which can exercise full control over their own security management, yet be isolated from and non-interfering with other co-resident user organizations.

Specific security features of the TAS are discussed below.

1. Structured Network Identifiers (SNI)

All TAS users are uniquely identified with an 8-character identifier of the form:

TAAGGUUU

where:

T	is the user's home TAS.
AA	is an Agency designator representing the user's agency.
GG	is a group within an Agency.
UUU	is the user within the group.

The structured identifier uniquely identifies all COINS users entering through TAS and permits both activity and security logging of an individual's network activity. A user requires an SNI and a password to logon to a TAS.

2. Access Authorization

Each user known to a TAS (i.e., who has an SNI) has an access authorization record in the Access Authorization (AA) file. The record contains the following basic information:

a) User's name

b) Clearance level

c) Logon (to TAS) password

d) SSN

e) Agency identification

f) Organization within Agency

g) Address

h) Telephone number

i) User type (student, crisis, operational)

j) Compartments (other than SI or TK which are included as part of the clearance data)

In addition, the record contains a list of the COINS applications (e.g., RYETIP, SOLIS, DIAOLS, ADCOM, etc.) and for those applications that provide files, a list of files authorized to the user by the user's home organization.

If the application is interactive (SOLIS, NDS), the user's access authorization record contains the interactive system logon information in the form required by the interactive system. This usually includes an identifier and password. The information is used to perform a user-invisible logon to the server-host supporting an interactive application. This "surrogate logon" service of TAS insulates COINS end-users from the considerable variability in logon protocols from one kind of computer system to another.

Application and file access controls are applied to terminals as well. Each terminal connected to TAS is logically identified by TAS and has an AA record defining which applications and files within the applications may be accessed by the terminal.

A "session security level" is logically established based on the user's authorization and his terminal's authorizations. This (conceptual) level controls what data may be accessed in a session.

The user and terminal AA files are used by TAS to implement the major functions of TMA-3:

- Control of user access to a data base.

- Verification that a user/terminal is cleared to receive a batch response.

As will be seen in a later section, the AA files and the TAS security architecture are expected to play increasingly important roles in further extending COINS services to the Community.

3. Server-Host Access Authorization

When TAS was upgraded from a user-host to include server-host functions in 1978, the access authorization function was expanded to include application access authorization data.

Interactive or batch applications hosted or front-ended by a particular TAS or HAS are registered in an access authorization file on that TAS or HAS. The access authorization file contains for each application a list of terminals and user's identifiers (SNI) and passwords authorized to access the particular application. Terminals are identified by a host-id/terminal-id combination (i.e., from what Agency an access is attempted). Within an application, the user's type of access (retrieve or update) can be restricted to specific files.

4. Decentralized Security Management

The TAS security management design was influenced by the following major considerations:

a) Each using Agency would be responsible for managing the security information and access authorizations of its own users.

b) A large using Agency may wish to delegate some of the security management to functional organizations within the Agency.

c) A single TAS may be shared by two or more independent Agencies.

To meet these somewhat diverse requirements, the TAS security architecture includes three kinds of users:

TASMASTER	A single user who "owns" the TAS and creates and directly or indirectly (see Administrative User) creates all other users.
Administrative User	A user who has the delegated authority to create and administer a specified set of ordinary users.
Ordinary Users	Users authorized to use TAS and the COINS network.

An Administrative User can add, modify, or delete users within the group that can be "named" with the same single "SNI-prefix" as assigned to the Administrative User. That is, the up to 1,000 users who have the same

TAAGG (TAS, Agency, group within the Agency) prefix in their SNI.

Administrative Users CANNOT affect any records other than those belonging to them.

The TASMASTER establishes the basic access authorizations for Administrative Users. The Administrative User can further subdivide his access authorizations among users within his domain. He cannot give any user more privileges than he has himself. It is not required to give an Administrative User ALL TAS or network privileges.

APPENDIX B

ADDING FILTERS TO RECON

The classification, codeword, and dissemination codes of the RECON records can be used in combination to identify material that is not to be released to external (i.e., COINS or other) users. This fact can be used to (invisibly to the user) apply a filter consisting of a series of AND NOT < dissemination codes and codeword codes > to each (implied) SEARCH command issued by a user to exclude restricted material from the search.

In general, the RECON implied SEARCH command produces a set of records that meet the specified search criteria. The result sets are associated with a user's work space and can be combined or limited in various ways after a search has taken place. It is possible to combine the results in two or more sets through logical operations (e.g., one can create a set on KW/BIRD (1) and another set on KW/SEED (2), then logically combine the sets 1 AND 2 instead of having to specify that intention in the initial search as (KW/BIRD AND KW/SEED)).

Because of the ability to manipulate sets to create combined sets which may then be edited to print records or any selected fields, it is necessary for the filter to be applied at the point where the total request is essentially satisfied. Since it is expected that external users of RECON will only be entering batch queries, the filter should be applied just before the output set is to be transmitted to the requestor.

Mechanics

The mechanics of using the filter approach to limit access to parts of the RECON data base involves the following three steps:

a) Recognizing external users.

b) Identification of the appropriate filter.

c) How (when) to apply the filter.

Each of these elements will be discussed below.

Recognizing External Users and Identification of Filters

This is as simple as setting a single bit in the user's logon identification record to identify that the specified user is to have restricted access to the file(s) that he is authorized to search. A one-byte designator of which filter is to be applied to this user whould be more than adequate for the foreseeable future.*

These two pieces of information will have to be included in the user control block established as part of the session. (The exact sponsor's jargon for this control block is not known; it is the data kept by RECON during a session that identifies the user, his terminal address, and holds the threads to his workspace and files.)

Above, the filter was described as a series of AND NOT < restricted dissemination codes > . It would also be possible to define the filter as

---

*This model is predicated on each user, by name, being registered in the RECON data base. A less burdensome (to RECON) approach would be to register by name only those external users who are granted extended access (perhaps to ORCON material) and treat the rest of the user opoulation by generic names (e.g., NSAUSER, DIAUSER, STATEUSER). The most restrictive filter would apply to the generic users.

AND < permitted dissemination codes > . Which form is better is a function of which would lead to the smaller specification.

How (When) to Apply the Filter

The best way to apply the filter would be to prespecify the set corresponding to the filter and then apply that set invisibly to the set that is generated by the user's operand specifications. Thus, in the hypothetical example:

KW/FLOOR AND KW/NILE would result in:

SET NO	NO. RECORDS	NO. OCC.	DESCRIPTION OF SET
n	25	25	KW/FLOOR
n+1	60	60	KW/NILE
n+2	15	15	n AND n+1 (AND NOT FF)

where FF is an invisible set number used to identify the prespecified filter set.

The mechanics of applying a filter are relatively straightforward. Whenever an output command is activated, it tests the bit in the user control block to see if this is a user to whom a filter is to be applied. If the test passes, the command extracts the one-byte filter-id and then takes an EXIT EXIT that allows it to apply the filter before returning to continue the normal output processing for external users.

Summary

This appendix has outlined a method of restricting access to RECON records based on the dissemination and compartment codes assigned to the

records at their type of creation. The technique provides as much control as desired based on labels assigned to the records. The modifications required of RECON to accommodate these additional controls are minimal (probably less than one-half a man-year of one of the system programmers maintaining the application).

Adding filters to RECON is a necessary adjunct to making RECON available to the Community if the simplicity of the GUARD is to be maintained.

APPENDIX C

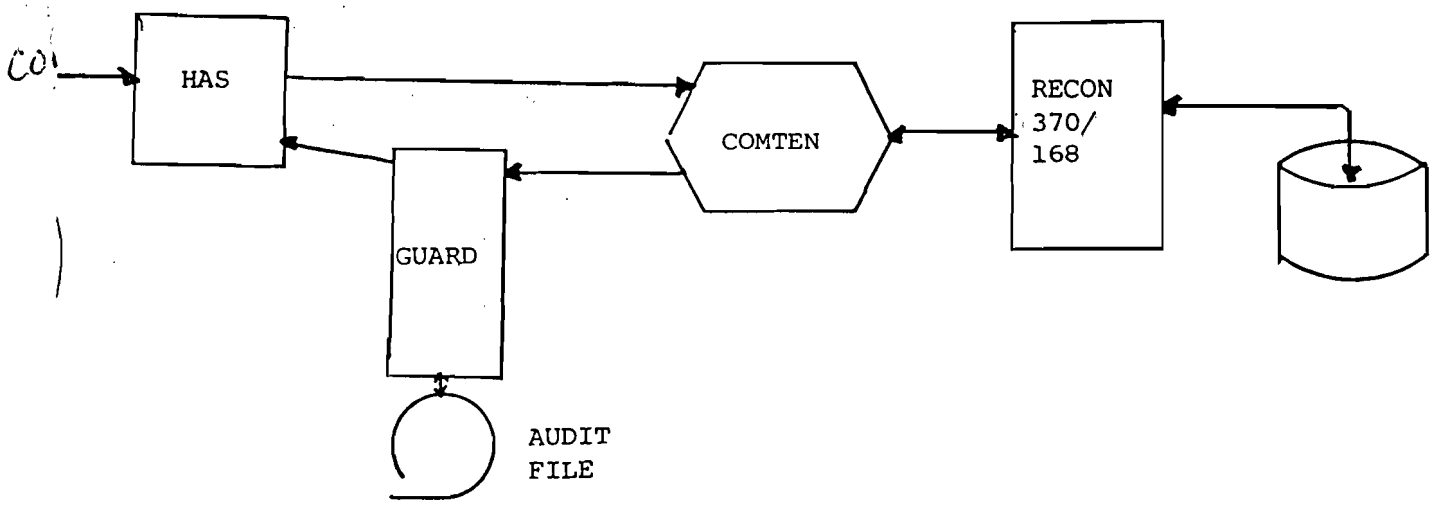
GUARD IMPLEMENTATION EXAMPLE

Configuration

The basic components of the on-line GUARD device, Figure 10, and the update GUARD device, Figure 11, which implement the cryptographic checksum function(s) are the UNIBUS-compatible 11/23 CPU and the "n" DES1100DSM printed circuit boards (PCB's). The other PCB's shown in the diagrams are used to implement the I/O and communications function for the GUARD device. The cryptographic checksum functions are critical. The components of the checksum subsection must operate correctly.

The configuration shown allows the on-line GUARD to handle multiple compartments (categories of releasability). The number of categories a GUARD device can handle is determined by how many DES1100DSM modules can be attached to the 11/23. Each DES1100DSM, operating under a separate (possibly hard-wired) key increases the problems and increases the complexity of a key management system. The categories handled by a single GUARD device can be altered by simply replacing/exchanging DES1100DSM PCB's. The GUARD device could fit within one DEC rack if constructed as indicated above.

The update GUARD device could generate "n" separate checksums by use of the Delta Data 760 T or Deltat 5000 function keys. The function keys would be used to select one of the "n" DES1100DSM modules to calculate the cryptographic checksum for a given RECON record.



On Line Guard Device Structure

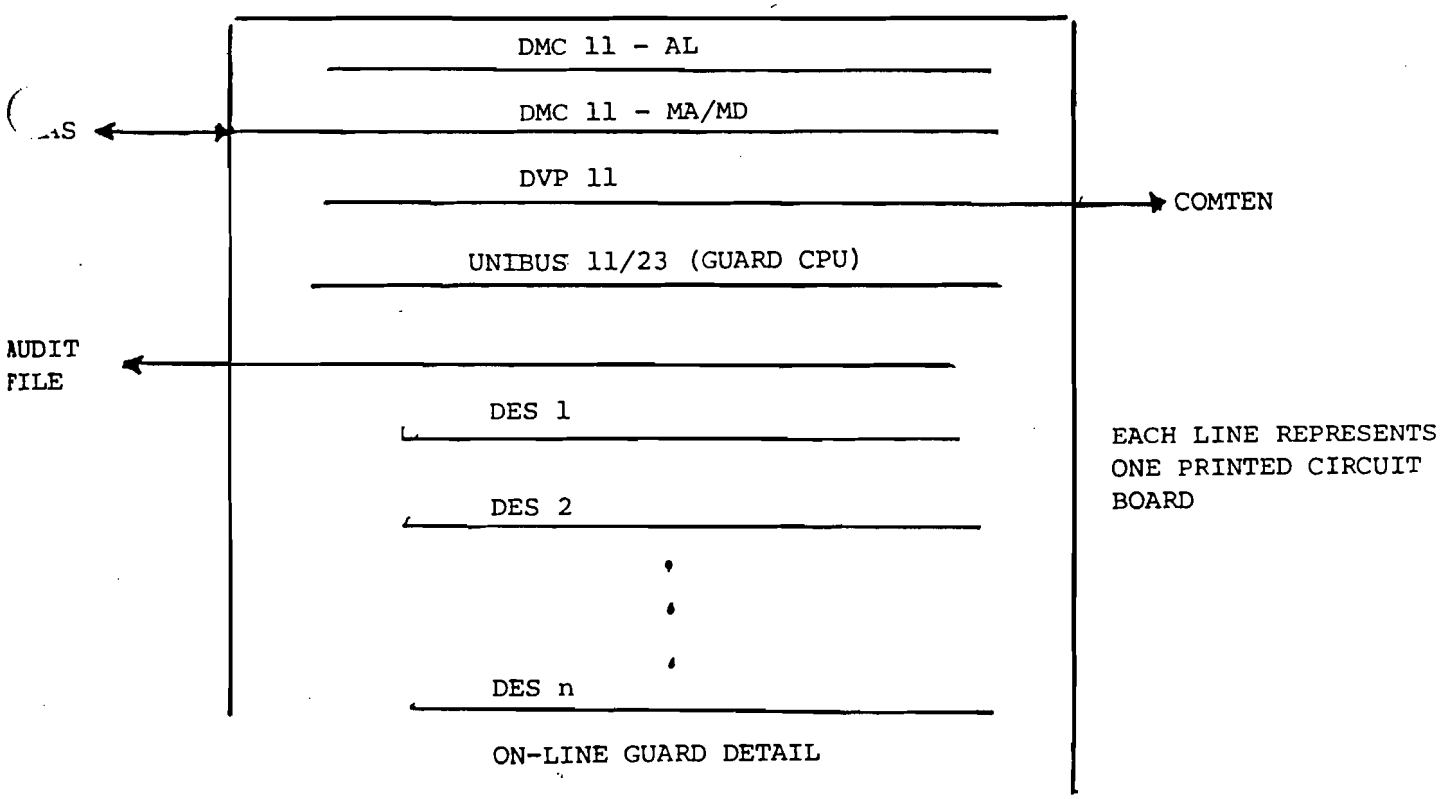
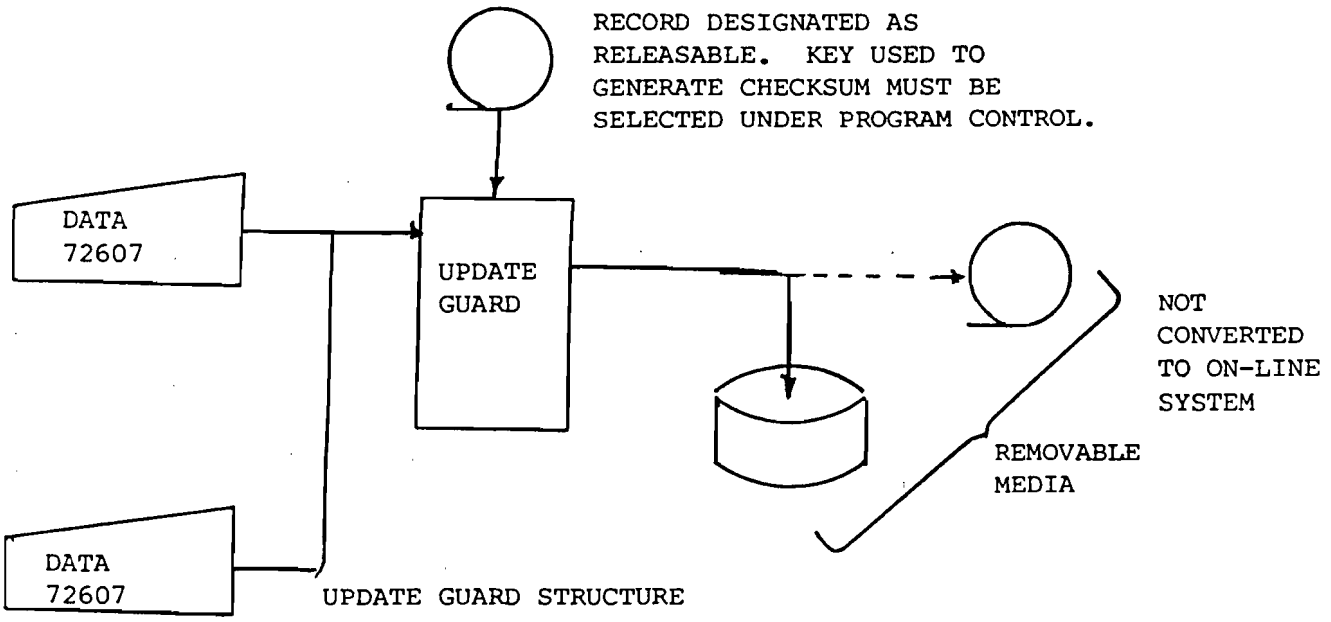
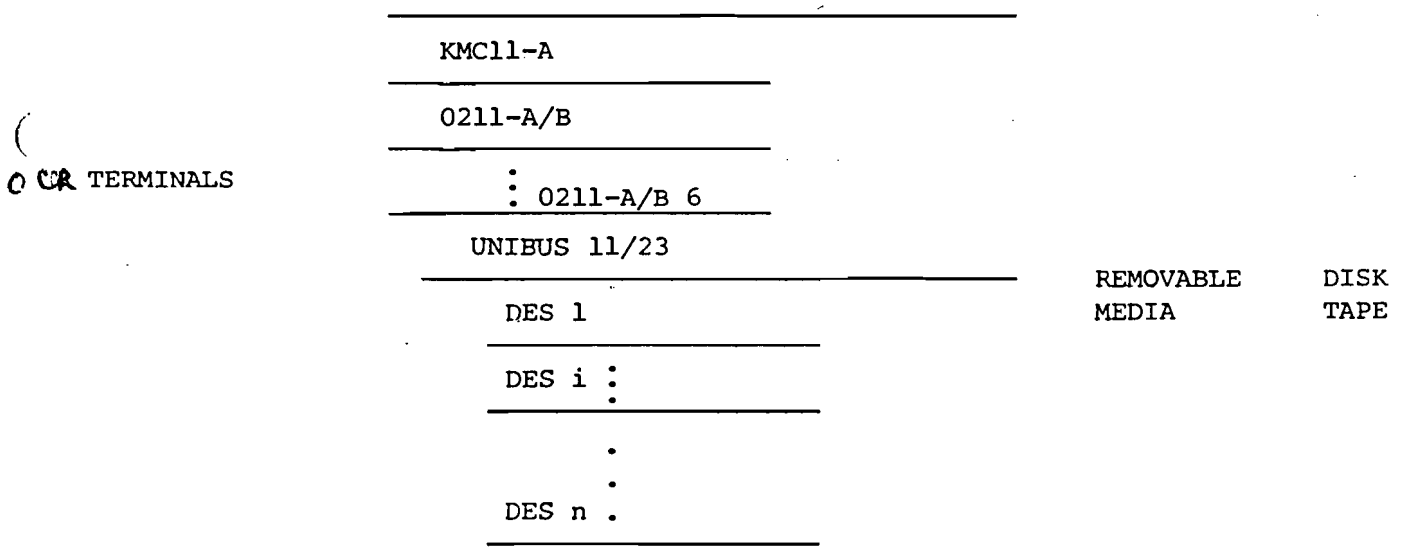


Figure 10

On-Line Guard



UPDATE GUARD STRUCTURE



UPDATE GUARD DETAIL

NOTE: The COMM IOP-n2 is an asynchronous DMA line controller. It consists of a KMC11-A auxiliary processor and up to six DZ11 asynchronous multiplexers. The DZ11 modules can each handle eight communications lines. The DMA capability relieves the update Guard CPU of the burdensome line handling functions. Flexibility and expandability are inherent in the configurations shown.

Figure 11

Update Guard

The number "n" of theoretically possible categories (compartments) that could be handled with this approach is a function of the DES modules used. The DES1100DSM is suitable for UNIBUS addresses 760000 through 777770 (octal). The DSM requires three contiguous 16-bit buffers. Thus, in the address range above, 2,728 three-word blocks are available. Obvious limitations on space, power, and I/O requirements will reduce the number of DSM's one UNIBUS can handle. However, each GUARD processor can obviously handle many categories.

Functional Description

The concept of operation is simple. As illustrated in Figure 12, the on-line GUARD device, monitoring a given channel, would contain DES1100DSM PCB's for each dissemination category designated for that channel. For a RECON record slated for output, the cryptographic checksum would be recomputed in parallel. If a match occurs with the checksum stored with the record, the data is transmitted.

This approach allows multiple categories with a single checksum field attached to each releasable RECON record. Non-releasable records would contain a blank or null checksum.

The use of the null checksum will standardize the data base structure. Dynamic data base update is then possible. However, all releasable records must have their checksums generated off-line by the update GUARD device.

This can be achieved by an additional off-peak-time procedure. Since all automatic changes to the data base will have null checksums and are thus not releasable, the following procedure could be used:

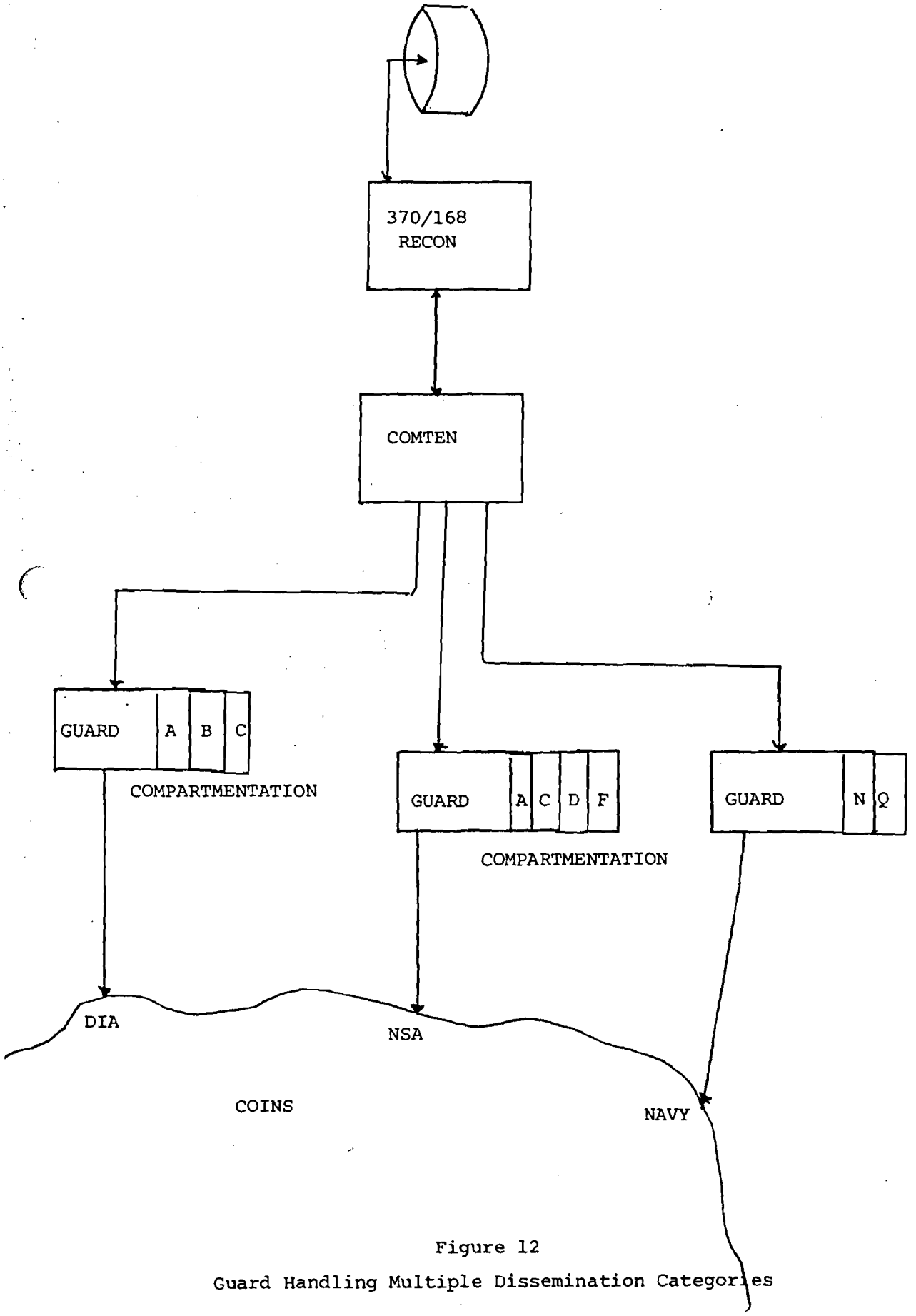


Figure 12

Guard Handling Multiple Dissemination Categories

1. All dynamically entered releasable records are extracted from the RECON data base at off-peak hours and stored on tape or removable disk packs.

2. The tape/disk containing the null checksummed releasable records is passed through the off-line checksum generator process of the update GUARD device.

3. The resultant output of RECON records with active checksums is then merged with the RECON data base during normal update.

The procedure for records entered via the Delta Data terminals would be identical to steps 2. and 3., above. The update GUARD device will accept input from the Delta Data terminals and the peripheral device containing the null checksummed releasable records from the RECON data base. The checksum subsection of the GUARD device is standard for each GUARD application (update or on-line).

APPENDIX D

APPLICATION OF GUARD TO SAFE

The GUARD device approach is applicable to controlled dissemination in the SAFE system. The checksum subsection (UNIBUS, 11/23 CPU, and "n" DESl100DSM's) would be unchanged. The peripheral interface of the GUARD would be structured to interface with the SAFE Processor Interface Units (PIU's) and/or Network Adaptors (NA's) instead of the COINS HAS computer. The PIU's provide the interface to the Hyperchannel (WBC). The NA's provide the interface to the Inter-Computer Channel (ICC). The diagram, Figure 13, illustrates a simple example of a SAFE connection. Figure 14 shows the detail for a SAFE GUARD WBC connection.

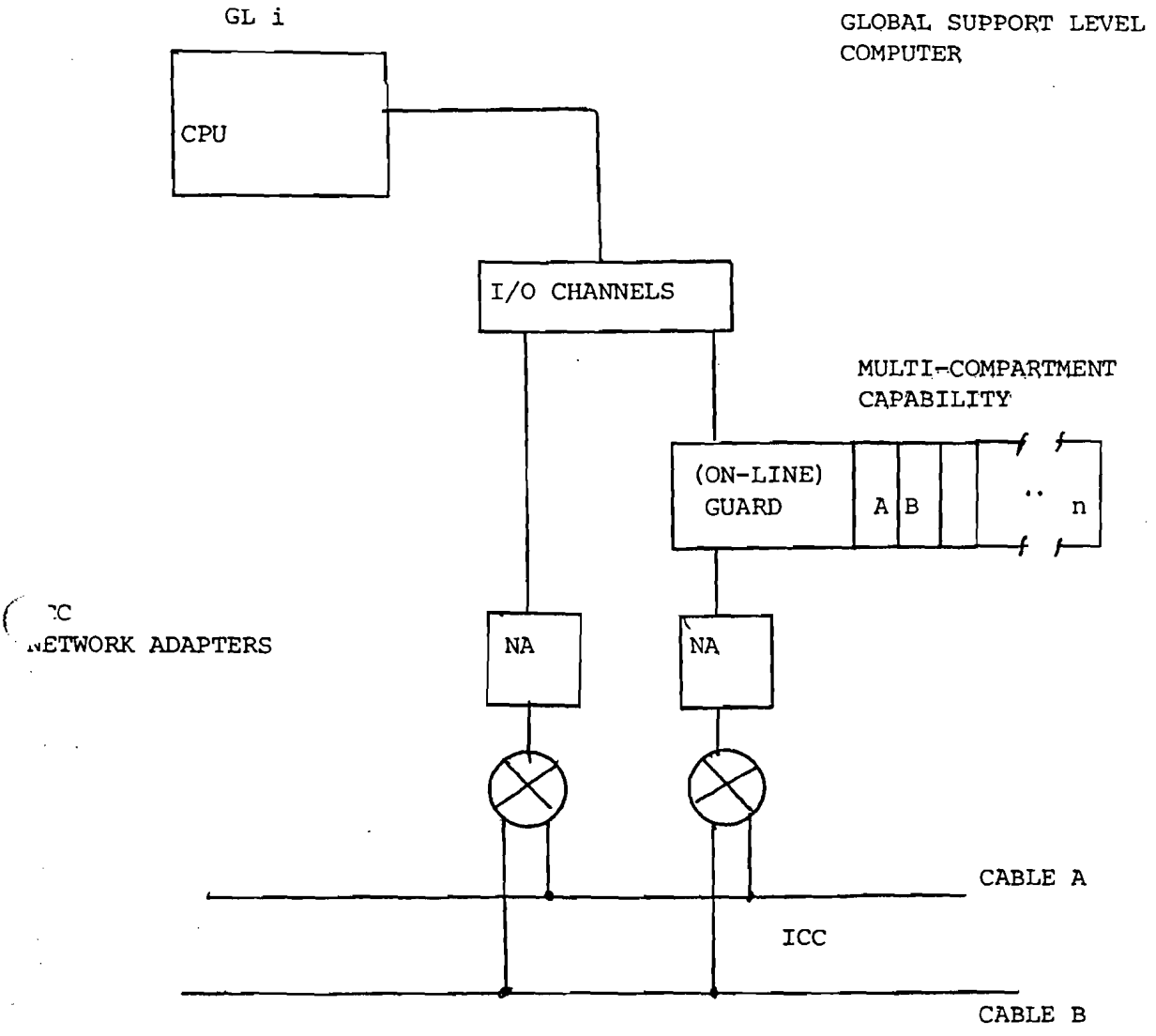
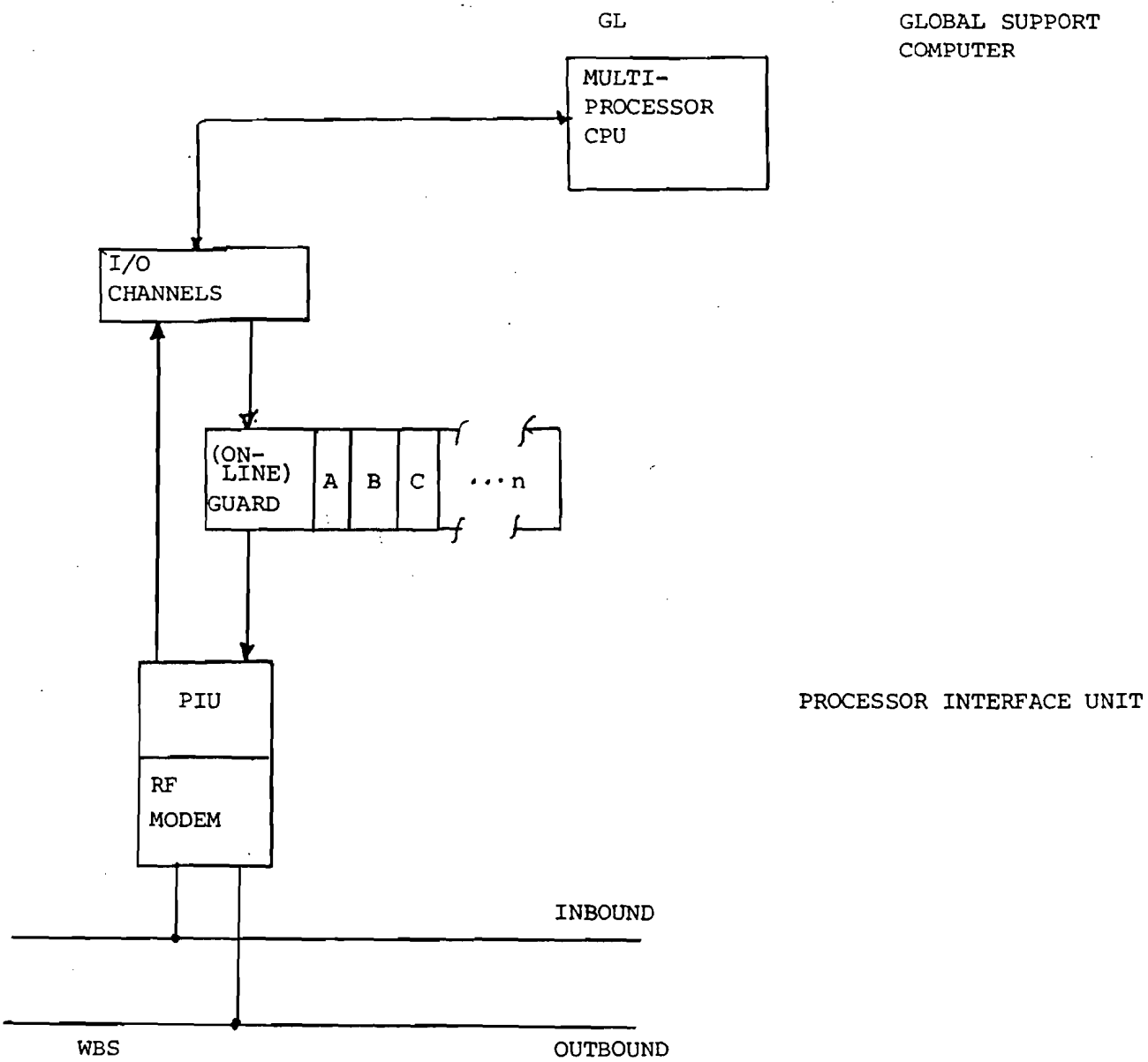


Figure 13

Safe (Guard) ICC Configuration



NOTE: In the ICC or WBC configuration, the structure and function of the Guard checksum subsection is identical to that of the Guard device proposed for the present RECON system. The only change is in the I/O interface structure of the Guard.

Figure 14

Safe (Guard) WBC Configuration

APPENDIX E

COMPARISON OF ALTERNATIVES TO SOLVING RECON SECURITY PROBLEM

A summary of the advantages and disadvantages of the various methods proposed for the solution of the RECON security problem is attached.

A COMPARISON OF VARIOUS ALTERNATIVES
TO SOLVING THE RECON SECURITY PROBLEM

	Present RECON	KSOS	Separate Systems	Filter	Authentication
Direct (internal) Penetration	No defense	Good defense Very low risk	Zero risk Best defense	No defense	No defense
Trapdoor (external) Penetration	No defense	No defense	Zero risk	No defense	Probability of unauthorized release is 5.24×10^{-20}
Spillage	No defense	No defense Very low risk of software error	Zero risk	No defense	Very low risk (guard machine fails and RECON processor fails)
Change of Functionality	No	None	? - could be large	No	No change for internal users, external users see a batch-only system
Modification to RECON ?	None	Substantial	None - some minor new software	Moderate	Some
New Hardware Required ?	None	Yes	Substantial (whole new system)	None	Yes - guard processor and input citation authenticator
Number of Access Classes Permitted	One (Agency internal)	Unknown - should be unlimited	Two	Unlimited	Several hundred
Cost	None	Very high 2 million +	Highest 2-5 million	Under $100,000	$250,000 - $300,000
Lead Time	None	3-5 years	2-3 years	6 months	1 - 1½ years

Xerox office copy

James P. Anderson Co.
Box 42 Fort Washington, Pa. 19034
215 646-4706

On The Bandwidth of Covert Signaling In RECON Using Error Messages - A Correction

April 4, 1981

Introduction

This note is a correction to data contained in an earlier report on the feasibility of connecting RECON to an external network [1].

In that report, the bandwidth of the covert signaling channel available through using error messages was erroneously reported as .085 bits/second. That rate was established under an incorrect understanding of how the COINS Host Access System (HAS) passed through 'messages' (records, etc.) from a host to a network user.

It was assumed that the messages were passed through to the destination user as soon as they were received from the host. In fact, the HAS accumulates an ENTIRE collection of records from the host before attempting to transmit the collection to the user (i.e. destination host or TAS). The reason HAS collects the entire set of output records for transfer to the user is presumably to package the output in maximum length packets to maximize the efficiency of the transmission through the communications network.

This aspect of the HAS was not understood when the report was written. The effect of the way the HAS REALLY works is to increase the bandwidth of the signaling channel using error messages.

Scenario

The scenario for the use of the covert signaling channel is that a Guard processor described in the report is in place between RECON and the COINS network, and that through some means, the RECON host has been subverted. The only way the subverter can transmit unauthorized data into the network is to signal the data as a binary stream by adopting the convention that one error message stands for a binary 1 and some other error message stands for a binary zero.

If the presumed subversion takes place, the subverter can signal arbitrary data by sending a sequence of error messages to the receiver. Because the error messages are static, they can be checksummed, and they will successfully pass the Guard (as proposed).

HAS collects an entire 'file' from the host it supports, then packages the file in maximum sized packets of 32000 characters for transmission over the network. Thus to signal out a nominal 300 character unauthorized RECON record, the subverted RECON would have to transmit 2400

'error messages' to the selected recipient. If the error messages are approximately 100 characters long, the unauthorized record would look like a file of twenty-four hundred 100 character records. Because the error messages would be correctly checksummed, they would pass the Guard, and the entire 'file' would be transmitted to the HAS. The HAS would then package the 240000 characters into 7.5 32000 character packets for transmission in COINS.

Computing the Effective Inter-Access System Transmission Rate

In order to determine the correct covert signaling bandwidth possible due to unconstrained use of error messages in RECON, it is necessary to determine the actual transmission time in the network. We can do this approximately by using the data that was collected for Table 1 in the report. That data established the access-system to access-system transmission times for messages of lengths varying from 100 characters to 31777 characters. The times measured included the time to open a connection from the sender to the receiver, and the time to acknowledge receipt of the transmission.

We can approximate the efective transmission speed through the network by assuming that the time to open and close a connection and transmit a receipt are essentially the same regardless of the length of the message(s).

The effective transmission time through the network can then be computed using the following formula:

$$L(M(1))-L(M(2))/T(M(1))-T(M(2))$$

Where L(M(i)) is the length of message i, and T(M(i)) is the observed time to transmit messages of that length. Note that the observed times have two components: the time required to open and close the connection between the access systems, and the actual time to send and process the characters of the message. In the formula, the difference in times eliminates the effect of opening and closing the connections and sending receipts since these are assumed to be fixed regardless of message length.

The average time to transmit messages of lengths 100, 1982 and 31777 characters each are shown in table 1, along with the number of observations used to establish the average.

Length of Message (Characters)	Average Transmission Time (Secs)	Number of Observations
100	12.86	29
1982	14.08	49
31777	32.73	52

Average Transmission Time for Messages

Table 1

Using the data in the table, the effective transmission time through the network is approximately 1600 characters per second (as shown below).

31777-100/32.73-12.86 = 1594 characters/second.

1982-100/14.08-12.86 = 1542 characters/second.

3177-1982/32.73-14.08 = 1597 characters/second.

Computing the Covert Signaling Rate

The effective covert signaling rate is the minimum of the transmission rates over the various paths, (Z,A,B,C and D) shown in Figure 1. We will neglect the delays imposed by the processing in the COMTEN and Guard as being very small compared to the transmission rates over the various paths. The processing time in the HAS and TAS is included in the effective transmission time through the network.

The transmission rates over paths Z,A and C are greater than the rates over paths B and D. The transmission rate over the B (through the DES to recompute the checksum) is on the order of 1200 characters/second (assuming the use of a 9600 baud rate DES chip). Thus the covert signaling rate is between 12 and 16 bits per second assuming 100 character 'error messages' rather than the .095 bits/second contained in [1]. This rate would permit covert transmission of a 300 character RECON record in between 2.5 and 3 minutes.

Countermeasures

A signaling rate of 12 to 16 bits/second is high. To counter the risk this poses will require a different approach to handling error messages.

An earlier examination of the error messages determined that there are only about 8 to 12 generic error messages in RECON. With some loss in the ability to pinpoint errors, and an increase in the complexity of the Guard, the generic

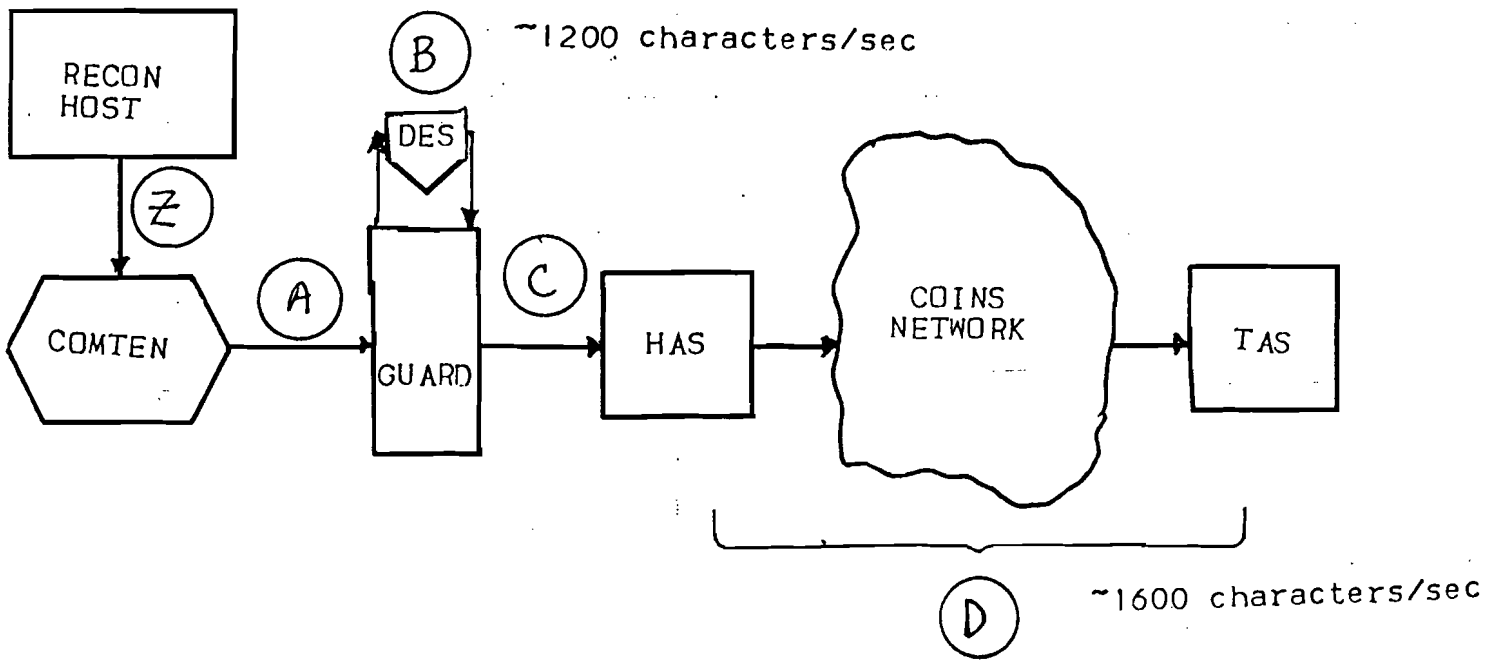


FIGURE 1

Transmission Segments Involved in Covert Signaling

messages could be stored in the Guard and 'called' by the RECON application in a RECON-to-Guard message that says 'Send generic error message N to user M'. By involving the Guard in the handling of error messages, it would then be possible for the Guard to recognize error messages (from ordinary query results) and thus be in a position to detect if/when error messages are being sent more frequently than an (as yet to be determined) allowable rate.

Reference

1. On the Feasibility of Connecting RECON to External Networks, J.P. Anderson Co., March 16, 1981.