REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1998	6/2/98 - 12/31/98

4. TITLE AND SUBTITLE
Accoutability Issues in Multihop Message Communication

5. FUNDING NUMBERS
C DAKF11-98-P-0304

6. AUTHORS
Dr. Sourav Bhattacharya
Dr. Raymond Paul

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Arizona Board of Regents
Arizona State University, Office of Research and Creative Activities
P.O. Box 871603
Tempe, AZ 85287-1603

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Army Research Laboratory
Georgia Institute of Technology
115 O'Keefe Bldg
Atlanta, GA 30332-0862

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

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

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Accountability (aka. Non-repudiation, or NRP) is a key component of information systems security, and it is a stated need in the Orange Book guidelines for security level classifications. This report presents a framework of the "accountability" needs of a message communication system. In . particular, we demonstrate that the traditional approach of Digital Signature (DS) based solutions to the accountability needs of a message k communication system is only one part of the overall problem. In a multihop message delivery system, there can be other aspects of accountability that may not be addressed using DS techniques. We identify a specific problem, namely the Sender's Ambiguity Problem (SAP), that remains to be solved if a comprehensive treatment to accountability could be developed.

The SAP problem is introduced and demonstrated in this report, and its relevance to multihop (where the hops could be physically separated routers, or logically distinct multiple software modules) message communication system is shown. We show that various application domains, including messaging systems and document distribution systems, are vulnerable to the SAP issue, and therefore, this research may have practical significance. The primary focus of this report is to identify the SAP problem (and, hence, raise a point that DS alone cannot completely solve the accountability problem). Then we present an outline of our research in SAP framework. The framework includes NRP categories, NRP types of services, NRP levels of certification. Finally, we present a set of metrics that can potentially be used to asses the SAP problem, and its existence severance, in a networked or distributed system. Follow on research is required to elaborate the SAP framework.

14. SUBJECT TERMS
Accountability, Computer Security, Digital Signature, Multihop Communication, Non-Repudiation

15. NUMBER OF PAGES
17

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-1
298-102

Accountability Issues in Multihop Message Communication

Sourav Bhattacharya
Department of Computer Science & Engineering
Arizona State University
Tempe, AZ 85287 – 5406
sourav@asu.edu

Raymond Paul
Test & Evaluation
DDR&E, OSD
Washington, DC 20301-3110
paulra@acq.osd.mil

Abstract:

Accountability (aka. Non-repudiation, or NRP) is a key component of information systems security, and it is a stated need in the Orange Book guidelines for security level classifications. This report presents a framework of the "accountability" needs of a message communication system. In particular, we demonstrate that the traditional approach of Digital Signature (DS) based solutions to the accountability needs of a message communication system is only one part of the overall problem. In a multihop message delivery system, there can be other aspects of accountability that may not be addressed using DS techniques. We identify a specific problem, namely the Sender's Ambiguity Problem (SAP), that remains to be solved if a comprehensive treatment to accountability could be developed.

The SAP problem is introduced and demonstrated in this report, and its relevance to multihop (where the hops could be physically separated routers, or logically distinct multiple software modules) message communication system is shown. We show that various application domains, including messaging systems and document distribution systems, are vulnerable to the SAP issue, and therefore, this research may have practical significance. The primary focus of this report is to identify the SAP problem (and, hence, raise a point that DS alone cannot completely solve the accountability problem). Then we present an outline of our research in SAP framework. The framework includes NRP categories, NRP types of services, NRP levels of certification. Finally, we present a set of metrics that can potentially be used to asses the SAP problem, and its existence severance, in a networked or distributed system. Follow on research is required to elaborate the SAP framework.

Key Words: Accountability, Computer Security, Digital Signature, Multihop Communication.

1. Introduction

Information security is a critical concern in computing and communication systems. As the usage of computers, and in particular inter-networked computers, grow in our everyday lives, the security and privacy issues underlying the usage of these computing and communication platforms become key issues. The explosive growth in internet and intra-net based systems to date has further aggravated the needs of securing sensitive communication across the network. Issues in internet security, and in general security of a distributed system have received significant R&D focus, and requires little elaboration.

Confidentiality, *integrity*, *authentication*, *non-interference* and *accountability* are some of the well-known issues in information security [1]. The focus of this report is on *accountability.* A definition and framework of *accountability* may be found in [2]. The term "accountability" broadly implies that the transacting parties in a secure system should be *made liable* to what they (each, individually) did do, as well as did not do. For example, if a user A did send a particular message, then A (or, some other user in the networked system) should be able to prove (subsequently, if someone else refutes A's claim) that the message was indeed sent. Likewise, if A did not send a particular message, then A (or, some other user in the networked system) should be able to prove that no such message was ever sent by the user A.

Overall, *accountability* implies being responsible for what the user did, and not be charged for what the user did not do. The accusations, claims and rebuttals aspect of *accountability* imply that legal notions can get involved, since a falsified claim of message transfer or a denial of a legitimate message delivery can eventually lead to a court of law, where the criteria of "beyond reasonable doubt" would have to be satisfied before a penalty charge can be asserted. *Accountability,* aka. non-repudiation (NRP), for a message communication system includes the following issues:

- Non-repudiation of Origin (NRO)
- Non-repudiation of Receipt (NRR)
- Non-repudiation of Submission (NRS)
- Non-repudiation of Delivery (NRD)

NRO and NRR indicate the accountability aspects at the source and destination nodes, respectively. While, NRS and NRD indicate the same across the message delivery system, viz. the communication channel. The focus of this report, which concentrates on the *accountability* aspects of multihop message communication, includes the NRS and NRD aspects across a set of intermediate hops, i.e., the routers and gateways, that constitute the message path from the source to the destination. In some cases, the *multihop* communication path between a human-source and a human-destination can include multiple software modules as well.

1.1 Overview of the SAP Problem

The traditional NRP model is based on a direct communication path, i.e., a physical channel, between the source and destination[1]. In such a model, if the source node could prove that it transmitted a message towards to the destination node, then it follows that the destination node must have received the message. This is because there is no other entity between the source and

[1] In certain cases, a trusted third party (TTP) [3] was assumed to exist, which could directly communicate with both the source and the destination nodes. The TTP model is built on a centralized, and 1-hop communication model, which is extended in this report to multihop message delivery.

destination nodes but a bare, non-devious and most likely non-intelligent physical media. Hence, if one can prove that the source node did transmit the message, then the delivery of that message to the destination node can be proved.

Multihop Systems:
However, the problem becomes complex if multiple intermediate nodes get involved. Suppose, a source node S transmits a message to a destination node D, and two intermediate nodes x, and y constitute the message path. S can prove that it did transmit the message, and yet D can refute having received the message at all. D, when asked later to clarify how this could happen, i.e., how, despite a proven message transmittal from S, the message did not reach destination — D could point the source of failure at the intermediate nodes x or y. In the world of internet, and intra-net (or, extra-net) based traffic delivery, the intermediate nodes (routers and gateways) are seldom trusted, and almost never equipped with detail traffic logging capabilities (refer Section 3, and particularly 3.6 for a discussion on the multihop communication can also be paralleled in a multi-module software component). Thus, if the destination node, D, refutes a claim of having received the message, and points the blame to the intermediate routers, then such an explanation would have to be accepted as a legitimate possibility. In reality, however, D might have received the message, and may have had vested (covert) interest in denying having received the message. This constitutes a failure of the system accountability[2].

This, and the problems of similar nature, are termed as the SAP (Sender's Ambiguity Problem) issues that can contribute to the failure in accountability in the communication system. The classical solution to accountability needs, namely the digital signature (DS), cannot eliminate the SAP problem. DS allows a message receiver (D) to prove that the source node (S) must have had sent it, since, no other node could have signed the message with the private key of S. However, if D claims not to have received the message, then S cannot prove that D indeed might have received the message – which is the focus of the SAP problem. SAP, in a sense, is dual to the role of Digital Signatures. A comprehensive treatment to accountability requires to address both DS and SAP issues.

1.2 Contribution

Digital signatures have been viewed as the solution to the *accountability* needs of secure systems. A proof of message delivery has been traditionally addressed using a return message acknowledgement. For a message from S to D, the following proofs could be made using digital signatures:

- If D receives the message, then D can prove that S only could have sent it.
- If D receives the message, and sends an (signed) Acknowledgement back to S, and if S receives the Acknowledgement message, then S can prove that D must have received the message.

However, the following items are also essence of *accountability* proofs, which may not be addressed using digital signatures.

- If D denies (perhaps, falsely) having received the message, then S cannot prove that D did actually receive the message and making a false denial.
- If S denies (perhaps, falsely) having received the Acknowledgement message, then D cannot prove that S did actually receive the Acknowledgement and making a false denial.

[2] Refer Section 3.6 regarding whether this problem could exist with Fortezza cards [4].

We term the above as Sender's Ambiguity Problem (SAP). A key contribution of this report is to demonstrate that Digital Signature (DS) is not a complete solution for *accountability*[3], and solutions using the DS techniques (e.g., the Fortezza cards [4]) cannot offer full accountability (refer Section 3.6 for details). The Fortezza cards must be equipped with additional capabilities to solve the SAP issue, which is the focus of this research. SAP issues also relate to reliability/availability, and denial-of-service factors. We show the impact of the SAP issues in real-life applications, e.g., the distributed messaging systems, or distributed document preparation and reporting systems. Finally, a framework for SAP, and an early identification of a suite of SAP metrics are presented.

2. Accountability Framework: Problem Formulation

Accountability refers to being able to prove what a particular user did do, as well as did notdo. In critical business applications, e.g., financial transactions, that are transacted directly in person, i.e., face to face, a large degree of the accountability problem gets solved using handwritten signatures, and double signatures (from both the transacting parties) on hard copies. An implicit assumption in such cases is that both the transacting parties could avail a common meeting place, and carry out the signatures on each others' presence, and/or in presence of a trusted third party (e.g., a notary). However, in electronic transactions, it is not always possible for the message sender and receiver to meet at a common point. In fact, the very need of not having to directly meet in person, leads to the usage and popularity of the electronic transaction forums. But, a price paid is in the potential of fraud and lack of accountability, as formulated below.

2.1 Forward and Reverse SAP

In a business transaction between two or more parties, different parties may have vested (i.e., covert) interests to denial of either sending a message, or receiving a message. Consider, for example, a user S placing a stock purchase request (over the web) at a certain time. Subsequently, if the stock prices fell down, the user S may have good reasons to deny having placed the purchase request. In this case, the (digital) signature of S with the purchase request message will prove that S had indeed placed the purchase request. A dual problem may also occur, where the stock broker (D) realizes that the stock prices went up, and it is convenient ☺ business not to have purchased the stocks before the price increase. So, D may deny having received the purchase request from S. In this case, S cannot prove having placed the purchase request, unless it did receive an acknowledgement from D. D may point the blame to un-trusted intermediate routers, and get away with the illegitimate business approach. We term this phenomenon as the *forward SAP*. In essence, *forward SAP,* refers to a situation where the sender cannot conclusively prove whether or not the destination node did actually receive the message, or was it a un-intentional message transmission failure.

The dual situation, *reverse SAP*, may also occur. Suppose, S sends a dated (e.g., 48 hours validity) contract to D, and immediately after sending the (signed) contract S realizes that a better business could be to award the contract to a third party, D2. However, since S has already initiated the contract request to D, it cannot be revoked, unless D declines to accept the contract, or a time-out occurs. S chooses to adopt this "time-out" option to nullify the contract as follows: D receives the message, accepts the contract, and sends back a (signed) Acknowledgement message towards S. S receives the signed Acknowledgement message, and realizes that it could be convenient business to not accept the Acknowledgement message. Thus, S awaits the passage of 48 hours, and afterwards terminates its previous "contract invitation" to D, and re-awards a contract to D2. User D, when notified of this

[3] Thus, digital signature is necessary, but not sufficient for *accountability.*

decision, claims to have signed the initial contract and having sent a signed Acknowledgement message. User S then refutes the claim, and points the blame to an untrustworthy intermediate node.

Note that if an Acknowledgement (from S to D) of the Acknowledgement (of D to S) is required to be sent, then a recurring version of the *reverse SAP* problem could re-appear. Thus, multiple rounds of Acknowledgement messages is not a solution for the SAP problem.

2.2 "Prove Beyond Reasonable Doubt" Requirement in Accountability

In the event of a breach of accountability, where claims, charges and denials (i.e., counter-charges) are going around, eventually the matter (if it is serious enough) may lead to the legal system. Now, a court of law requires to prove "beyond reasonable doubt" before impounding a penalty judgement to a faulty party. If a situation occurs, where it is clear that either a destination node D is making a false denial of message reception, or an intermediate node X could have had a message failure – then, although it is clear that the fault lies with one of the two (D, or X), a penalty judgement can be given to neither. The presumption of penalty judgement is based on that unless proven guilty, one is innocent. Thus, although it may be rare (i.e., a small probability) that the intermediate node X would fail to deliver a message, and while it may be quite obvious using common sense that the destination node D is making a false denial, it cannot be proved in a court of law.

Such is the need for a truly accountable system, in the sense that accountability denials must be resolved in a proven, conclusive and unambiguous fashion. The lack of conclusiveness, and/or unambiguity, if any, should be at an extremely low probability level, e.g., 0.0001% or even lower.

2.3 Timing Attributes of Accountability

The need for accountability stems from critical business transactions, or transactions which are valued for other reasons (e.g., sensitivity, social, legal, political aspects). In such transactions, often the timing aspects of the transaction is also important. For example, with a stock purchase request that has been subsequently denied by the broker, the timing of the transaction is also important. If the message sender cannot prove that (s)he sent the message at a particular time frame, then the motivation for the entire transaction, and refusal thereof (by the receiver), may loose significance. Likewise, in the "48 hour timeout" example stated in Section 2.1 above, establishment of the times of the different actions is a critical step.

Therefore, the four aspects of Accountability, or NRP, namely:

- Non-repudiation of Origin (NRO)
- Non-repudiation of Receipt (NRR)
- Non-repudiation of Submission (NRS)
- Non-repudiation of Delivery (NRD)

also includes those at specific time intervals. The transacting parties should be able to establish the NRO, NRR, NRS and NRD properties at or around given time instants. Finally, for a multihop communication across *n* intermediate hops (or, equivalently *n* intermediate software modules), the NRO, NRR, NRS and NRD properties must hold for each intermediate node. The accountability framework, thus, needs addressing the following needs:

Underlying techniques in packet encryption are also used in the creation of digital signatures. Digital signatures can provide a certain degree of non-repudiation, however, cannot solve the SAP issue. The distinction is in between the cases of "message has reached", or "has not". If the (signed) message has reached the destination, then the destination is empowered to prove that the source alone could have sent it – since, none other than the source node could have used the private signature key of the source[5]. Likewise, if the return acknowledgement message has reached the source, then the source (which is, in this case, receiver of the acknowledgement message) can prove that the destination alone could have sent the acknowledgement message. Note that, in this case, the return acknowledgement is treated as a message sent from the destination to the source.

However, if the message has not reached (or, claimed to have not been received by the destination), then the respective source nodes cannot prove anything – which is the focus of the SAP problem. The sender of the message is left with ambiguity whether the message truly did not reach the destination, or if the destination node is playing fowl and making a false denial. This ambiguity stays both for the message, and its return acknowledgment. We term these two cases as the *forward SAP,* and *reverse SAP*, respectively.

Depending on the terminology used, the lack of accountability, or ambiguity issue discussed here could also be referred to as a "traceability" aspect. The lack of proof on the sender's behalf regarding whether the message truly did not reach the destination, or if the destination node is playing fowl and making a false denial – it is also a case of missing traceability. The fact that the multihop message communication system couldn't provide adequate capabilities to track down, and pin point exactly where the message got lost (if at all), or whether the message was indeed delivered – it creates the ambiguity and lack of conclusiveness in accountability proofs.

> **Digital signatures are necessary, but not sufficient for solutions to the "accountability" problem. SAP, for example, is a problem that cannot be solved using the digital signatures. Additional capabilities, beyond the digital signatures, are required.**

3.2 Role of Signed Receipt

Traditionally, the issue of how the sender node could prove about the message delivery and reception at the destination – has been addressed using the return (signed) acknowledgement issue. The argument is as follows – if the destination node successfully receives the message, un-signs it, and acknowledges the receipt using a return acknowledgement message using its own signature – then upon receipt of the signed acknowledgement, the source node could prove that the destination node must have received the message. This seems to indicate that the ambiguity problem for the sender is solved using the return signed acknowledgement message. In other words, the SAP problem really does not exist so long as a return signed acknowledgement can be sent.

Such is not the case, on a closer look. The return (signed) acknowledgement message is nothing but a new message transmitted in the opposite direction, e.g., from the destination to the source. While, the return acknowledgement message can prove the delivery of the forward message, the delivery of the acknowledgement message itself becomes an open issue. The return acknowledgement can

[5] There is a small possibility that a third party user could decipher the private signature key of the sender, which has been the traditional focus of digital signature R&D (viz. how to make the key progressively more difficult to break). However, this is an orthogonal issue to our research. We assume that the digital signatures are trustworthy, i.e., hard to break.

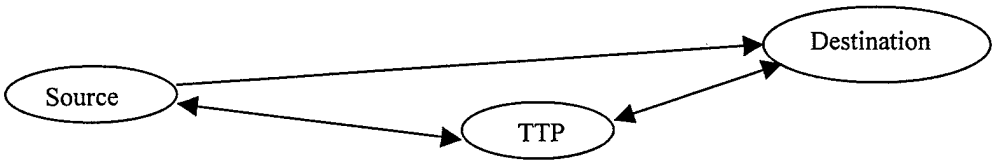
solve the SAP issue of the forward message, but it re-creates the SAP issue for the acknowledgement message itself. Refer to the example in Section 2.1 (defining *reverse SAP*) for business motivations to make denials to the acknowledgement message itself.

3.3 Traffic Log Capabilities

An accountable multihop message communication system has a close analogy to the accountable mail delivery system of the US Mail. Registered mail delivery in the latter is based on a detail logging capability, at each one of the intermediate mail router stations. There are a set of basic assumptions, or premise in the accountability of such a system, as follows.

- First, each intermediate mail router is equipped with a mail logging capability, a process that documents for each registered mail when it was received, by whom, and who sent it out to the next delivery station. In the wake of a refusal, or a false claim afterwards, the system can track down exactly where the packet got lost, or delivered.
- Second, the trustworthiness of the mail router stations, and that of the US Mail service personnel are assumed. The handwritten signature of the service personnel, and the fact that such personnel have no conflict of interest issue with either the sender or the receiver, lead to the accountability proof.
- Third, the reliability of the mail router stations, and that of the service personnel, are assumed. If one particular service person is unavailable on a particular date, another person can replace, and their respective signatures provide the tracking capability as to who logged the registered mails.

Thus, the solution to the SAP problem, if we create an analogy to the US Mail system, seems to indicate the need for a traffic logging capability. It is not a simple solution, however, as keeping a log of the billions of packets flowing through the internet to date, and that too in a timely fashion (i.e., fast enough) is not an easy task. More importantly, the internet gateways and traffic routers are existing commercial systems, and they may not agree to insert such capabilities into their existing products. Unless motivated from a commercial and business standpoint, the traffic logging approach may not be a feasible solution for the SAP problem.



Trusted Third Party in Accountable Communication.

3.4 Role of Trusted Third Party

Accountable and trustworthy transactions in business and finance, to date, often rely on a trusted third party concept (e.g., a notary). The objective is to provide an independent witness to the transaction, so that a subsequent refusal can be tracked down to the independent witness in a court of law. The same can be extended to accountable electronic transactions, as noted in [3] and shown above.

However, the trusted third party (TTP) concept is primarily based on a centralized system, where both the source and the destination nodes could directly communicate with the TTP. A centralized

system solution has inherent difficulties of the lack of scalability, and congestion – for example, if a large number of (source, destination)-pair messages are transmitted around the network, then the TTP will be loaded and may become a hot spot (unless, multiple TTPs are invoked). Furthermore, in a multihop communication environment the centralized assumption is no longer true, and the TTP may not be able to witness the transacting activities of the sender (or, the receiver) without having to worry about the trustworthiness of the intermediate routers and nodes. Thus, the TTP approach is essentially for a centralized system, and may not scale well to multihop environments. Besides, it does not solve the SAP problem, simply because it cannot scale to multiple hops.

A particular design of accountable message delivery system using TTP needs to be discussed to show the existence of the SAP problem. In this design, the source node (S) sends a signed message to the destination node (D) – but the key to un-sign the message is held at the TTP. The destination node (D) cannot pre-view the message unless D sends a request to the TTP to obtain the (unique) key to un-sign the signed message. (This key, held at the TTP, could vary with time, or message sequence number, or any such changing parameter, such that D could not use a key obtained during previous message exchange to un-sign a subsequent message.) Thus, if D makes the key-request to the TTP, then the TTP (which can be a trusted entity, regulated and monitored by trusted agency or individuals) will have a record of it, and can prove in a court of law later that D indeed had received the message from S – otherwise, why would D ask for the key to un-sign the message.

> However, even in this case, D can create the SAP issue – as follows. D can request for the key from the TTP, and TTP can very well keep a record of such a request. Next, D brings the key back to the destination node, and un-signs the message – previews it, and wants to make a denial of "seeing" the message. D argues as follows – while, it is true that the key was requested at the TTP, it was lost during transmission from the TTP to D, and hence, the key quite never reached D. Hence, D was unable to un-sign the message, and never received the "content" of the message. In this case, S will not be able to un-ambiguously prove that D did receive the key, and unlocked (i.e., un-signed) the message, but making a false denial.
>
> The main accountability denial stems from the fact that the path from the TTP to D is multiple hops (either physical hop / router, or multiple software modules).

3.5 Fairness Aspects of NRP

In providing an accountable message communication solution, the relative effort to be taken by the sender and the receiver may become an issue. If the entire performance overhead, of providing accountability, is upon the message sender, then it is unfair to the sender side. Likewise, if the entire performance overhead is at the receiver side, then it is unfair to the other extreme. Fairness aspects of the NRP protocols have been studied [3], where a uniform distribution of the performance overhead to both the parties is an objective. This research is orthogonal to our focus in this report, however, we note that when a SAP solutions approach is proposed the fairness aspect should be considered. For example, the accountability related performance overhead should ideally be split uniformly between the source, destination and the intermediate nodes.

3.6 SAP Issue in Fortezza Cards

The Fortezza cards [4] have been developed as a solution to the accountability, and many other security needs in secure communication systems. Defense systems of information generation, distribution, and dissemination often rely on Fortezza cards. Now, the SAP problem may also exist for Fortezza card based systems – described as follows.

Consider two cases: 1) where the Fortezza cards are tamperproof, and 2) where they are not. For case 2, the existence of the SAP issue is simpler to illustrate – thus, we address the case 1 first. In case 1, since it is a tamperproof system, the Fortezza card (hereafter referred as the Fc) is not likely to participate with the human-destination-user in creating accountability denials. In other words, the possibility where the human-destination-user might have received (and, pre-viewed) a message but does not want to acknowledge having received it (for vested or covert interests) – really does not exist at the Fc level. The Fc cannot be programmed, or configured to make a false denial by the human-destination-user. The Fc at the destination user's computer will acknowledge having received the message, and will most likely send an acknowledgement message back at the sender node's Fc. (This argument will also relate to the validity of the SAP issue existence in secure communication protocols, such as SSL, or MISSI.) Therefore, one might question – does the SAP problem really exist for systems equipped with Fc. The answer, we argue, is in affirmative, as discussed below. (Note that, in case 2, i.e., where the Fc is not tamperproof, then the SAP problem can of course exist, due to the active involvement of the Fc itself in the false accountability denial.)

To analyze this case further we need to distinguish between the human-destination-user, and the Fc-at-destination. While, the latter is not likely to create a false accountability denial, the former (i.e., the human) can create a false accountability denial. The human-destination-user again can be of two types: human destination user in person, or a trojan horse operating in the human destination user's computer system. There are also other players in the system, namely the Operating System (OS) which is most likely a commercial-off-the-shelf (COTS) software, and software application tools (e.g., Microsoft Mail, or Netscape Mail). Therefore, the message flow at the destination computers can be as follows:

Fc-destination => OS-destination => Mail-Tool-Destination => human-user-dstn

Or,

Fc-destination => OS-destination => Trojan Horse Destination

Logically, each one of the components in the message path, albeit all physically within the destination node's computer, can be paralleled with the multihop message communication path discussed in Section 2.1. It is true that the Fc-destination would possibly not make a false denial of a message receipt (since it is tamperproof), but the human-user-destination can do so. Now, when subsequently, a court of law establishes that the message was indeed received at the Fc-destination, and questions how could it not have reached the human-user-destination – then the (falsely denying) human-user-destination could point the blame to a not-100%-trusted OS, or not-100%-tested-and-trusted Mail Tool. Since, these software components are often COTS (for economy, ease and familiarity of usage etc reasons), they cannot be proven to not have created an accidental message loss. The message loss could have occurred for various reasons – as the human-destination-user would argue – such as, buffer overflow, time-out in scheduling, or even an incorrectly designed COTS software. Since, the possibility of a COTS OS encountering a buffer overflow is not 1 in trillion range – such an excuse will have to be accepted as a possible one, and hence the human-destination-user will get away under "innocent unless proven guilty" argument.

The notion of trojan horses, and the possibility of their existence, adds to the severity of the above problem. Although, computing systems may be claimed to be tamperproof, and/or free of virus / trojan horse – such is not practical reality. In fact, much of the research

motivation in covert channels is based on the distinction between human-user and trojan-horse-user. Identical argument, in this case, would drag the possibility of a remotely-inserted (e.g., Java applet) miscreant into the destination node's computer. The human-destination-user, under a vested (covert) interest plan, may deny having received the message, and get away with it by pointing the blame to a possible trojan horse, which might only have existed for a short period of time, and deleted afterwards (to explain why the trojan horse could not be located at a later time).

The next case is where the Fortezza-card is not tamperproof. In this case, the SAP issue existence is rather straightforward. The "multiple" hops, or steps of communication will not only include the OS, COTS software (as discussed above), but will also include the untrusted physical routers and gateways across the internet, intra-net and extra-net.

4. SAP Framework

We propose a framework for SAP, which includes categorization of the underlying R&D issues, and identification of the different types of messaging services and levels of certification that are deemed necessary for an overall accountable message communication system. The framework is followed by a suite of SAP metrics, much of which is left as a future area of research.

4.1 NRP Categories

The question is how many different types of non-repudiation issues that can occur in a system. Section 2.1 identified the *forward SAP* and *reverse SAP* issues, which are two distinct types of accountability denials, and prevention requirement, thereof. We detail, in the following, some of the other types of accountability denials. These are called NRP Categories.

- *Forward SAP*: This refers to a case where the destination node (perhaps falsely) claims to not have received the message, and if the source node claims to have sent the message – then the destination node points the blame to an un-trustworthy intermediate node. The accountability provider mechanism, must in such cases, be able to uniquely identify whether the message was indeed lost in transmission (i.e., non-intentional), or being falsely denied upon to have been received.

- *Reverse SAP*: This refers to a case where the destination node does indeed acknowledge having received the message, and returns a signed acknowledgement message – but the sender node does not declare to have received the acknowledgement message. It could happen in two ways: either the return acknowledgement message gets lost in transmission, or, the sender node makes a false denial. The accountability provider mechanism, must in such cases, be able to uniquely identify whether the return acknowledgement message was indeed lost in transmission, or being falsely denied upon to have reached the sender node.

- *Iterative SAP:* This refers to the iterative SAP between *forward* and *reverse* SAP. The return acknowledgement provides accountability for the forward message, but creates a new lack of accountability for the return message itself. Thus, an acknowledgement for the acknowledgement message, i.e., level-2 acknowledgement, becomes necessary. Next, a level-3 acknowledgement becomes necessary for providing accountability for level-2 acknowledgement. This process can continue forever, unless a termination or arbitration mechanism is in place.

- *Boolean or Fuzzy:* This refers to an issue whether the accountability proof is deterministic, or probabilistic. In deterministic accountability, the system is either completely accountable, or not. In probabilistic accountability, the degree of accountability is denoted at a percentage level.

- *Granularity (per Packet, per Message):* This refers to an issue whether the accountability proof is on a per packet basis or per message basis. Suppose, the traffic logging capability is provided with to track down any accountability denial. The question is then whether the traffic log is generated on a per packet basis, or per message basis. If a message consists of multiple packets, then the packet-based traffic log would create more performance overhead than message-based traffic log.

- *Temporal:* It refers to the accountability denial, and proof in response, at a particular time instant or interval. Certain accountability denial is regardless of any time instant, e.g., "a particular message was never received". On the other hand, certain other accountability denial is specific to time intervals, e.g., "a particular stock purchase request was not placed before close of business of a given date". In this case, the accountability denial is specific to a time interval. Since, many critical business transactions are deadline specific, *temporal accountability* is a serious concern. In temporal accountability establishment, the notion of a global clock (and related research, e.g., clock synchronization) is essential.

- *Failure Proof:* The accountability provider mechanism should be able to distinguish between intentional (i.e., false claims) denials and non-intentional message failures. This is a key distinction, as in one case the conniving party must be penalized, while in the other the loss must be accounted to a non-ideal message transmission medium. In this regard, the accountability provider mechanism would relate to the reliability of message transmission.

4.2 NRP Types of Services

In follow up with the US Mail delivery system, we propose three primary types of NRP services. These are:

- Registered Mail
- Certified Mail
- Special Delivery Mail

Registered Mail is the most accountable message delivery mechanism, where at every intermediate router a log entry is created to track the flow of the message. The sender, and every intermediate node are required to have a trusted, reliable, and timely log book creation mechanism, for every transacted message or packet. This type of service is likely to be slow, and require both CPU processing and buffer overhead at the intermediate nodes.

Certified Mail is the next level of accountability, where the sender can provide the proof of sending the document, but cannot prove if the destination node actually received the message. Here, the intermediate nodes may not have as much overhead of tracking all the messages and packets, leading to a higher performance. However, the SAP problem may still exist, as the receiver node may falsely deny having received the message, and no audit can be enacted to prove whose fault it is (the intermediate nodes', or the destination node).

Special Delivery Mail is a special case of the registered mail service, where a particular trusted agent traverses the multihop network along with the message, and carries out the task of traffic log creation at each intermediate node. The sequence of steps to be carried out at each intermediate

node remains identical as to those with registered mail, however, here these tasks are carried out by the special (mobile) agent, which achieves both speed and efficiency. This type of service is analogous to a trusted human agent carrying a message in person.

Priority Services:

The above three types of services can be offered in any combination, and particularly at specific priority levels. Thus, one can have priority "high" registered mail, or priority "low" registered mail, and so on. A system wide set of priorities can be assigned, e.g., levels "high", "medium", and "low", or a localized (specific to each node) set of priority values can be adopted. There are well-known tradeoffs [5] for global priority vs. local priority values in real-time scheduling area, and similar tradeoff can be observed in prioritized NRP message transmission also. As an alternative, the priority values can be based on per user basis, e.g., some user deemed more important than some other users. The detailing of these priority services and performance issues thereof remain as a future research.

Billing:

Once the different types of non-repudiation services have been established, it remains as an added task to establish the billing options, e.g., rates for the different types of services. This is an issue to deal with economics of telecommunication services [6], and is beyond the scope of our current research.

4.3 NRP Levels of Certification

We propose a multi-stage quality model to develop different levels of certification for the NRP levels of a particular system under consideration. The idea is to propose a suite of SAP Metrics (refer Section 4.4), and either A) a set of weight or relative importance factors to combine these metrics into a single parameter, or B) a set of rules (i.e., logic flow) to unify the measured values into a SAP level. An weighted combination of the measurements (per the SAP Metrics), and the weights lead to a composite NRP capability of the system. Next, the composite NRP capability of the system can be compared against a threshold, and a certification level can be arrived. Thus, our approach is based on the following ideas:

- Suite of SAP Metrics, including some binary and some multi-level metrics. The metrics include both network-centric, and security specific attributes.

- **Weight-Based (Learning Approach):**
  - A set of weights, indicating the presumed relative importance of the SAP metrics. The set of weights can be application domain dependent, and can learn or adapt with time or with different domain.
  - An aggregation mechanism that combines the SAP metric based measurements and weights into a composite index. This composite index denotes the level of (NRP) capability of the particular system.
  - Threshold based computation of the NRP certification level. The set of threshold values, one for each certification level, can adapt with time and/or application domain.

- **Logic Based:**
  - A set of rules to unify the SAP metrics' measurements, and combine them into a single value or single measurement.

- The rules themselves may update (i.e., learn) with time. However, we initiate with a fixed set of rules, and continue with the measurement.

4.4 SAP Metrics

The SAP Metrics are grouped into two categories, one that relate to the measurements around the intermediate nodes, and the other that concerns only about the source and destination nodes.

SAP Metrics for the Intermediate Nodes: For each intermediate node, and/or router, the following metrics apply. Note that each intermediate "node" can also be intermediate software modules in a multi-module software chain (refer the discussion in Section 3.6).

- Logbook availability (binary metric)
- Buffer Space of Logbook (multi-level metric)
- Time-Stamp Capability (binary metric)
- Error-Detecting and Correcting Capability (multi-level metric)
- Re-Transmission Capability (including time-out ranges, multi-level metric)
- SNMP Level Handshake Capability Between Adjacent Hops-Pair (multi-level metric)
- Personalized Digital Signature Capability (binary metric)
- Signature Key Lengths (multi-level metric)

SAP Metrics for the Sender / Receiver Nodes: For either the sender, or the receiver node, the following metrics apply.

- Capability to transfer a secure message enclave
- Capability to inquire the (trust)-status to each intermediate node
- Capability to demand for "receipt" from each intermediate node(s)
- Capability interact with trusted 3rd party nodes, if any
- Capability to install, remotely, a trusted computing base (TCB) at one or more of the intermediate nodes
- Capability to install, remotely, a trust-monitor at each intermediate node, which can watch out for abnormal events, including intrusions.

Identification of the SAP metrics, in detail, and in particular those for multiple software modules (instead of multiple network nodes) is left aside as a future research.

5. Summary and Conclusion

This report discusses a notion that digital signatures alone cannot offer a comprehensive solution to the accountability needs of a secure multihop communication system. We demonstrate that while digital signatures are necessary, but they are not sufficient for the accountability needs, and a class of accountability denial problems (namely, "Sender's Ambiguity Problem, or SAP), demonstrated in this report, need to be resolved. The approach of using return (signed) acknowledgement messages to offer round-trip accountability may be inadequate, as the acknowledgement for the acknowledgement message becomes a necessity in an iterative fashion.

We discuss related R&D issues, e.g., message logging, packet encryption, fairness aspects of non-repudiation, and trusted 3rd party models are addressed. Finally, we propose a framework for SAP, including different types of non-repudiation categories, various services for highly accountable

message delivery, and an approach for the non-repudiation (NRP) level of certification. The latter, i.e., NRP level of certification, leads to a suite of SAP metrics, identified in this report. Further details on the SAP problem, which will include both measurement & metrics (i.e., Test and Evaluation) and algorithms to solve the SAP problem, are left as future research. Authors are currently conducting research in proposing a framework to the SAP problem, both for multi-node hops and multi-module software components.

References:

[1] Bell D.E., La Padula L.J., "Security Computer Systems : Mathematical Foundations", Technical Report, Hanscom AFB. Bedford. MA. Rep. FSD-TR-73-278. ESD/AFSC, Vol. 1, 1973. (Also available as a Technical Report from MITRE Corp., Bedford, MA, 1974.)

[2] Zhou J., Dieter G., "Evidence and Non-Repudiation", Journal of Network and Computer Applications, July 1997, vol 20, no 3, pp 267 – 281.

Also in:

Kailar R., "Accountability in Electronic Commerce Protocols", IEEE Transactions on Software Engineering, vol. 22, no. 5, May 1995.

Zhang N., Shi Q., "Achieving Non-Repudiation of Receipt", The Computer Journal, vol. 39, no. 10, 1996, pp 844 – 853.

Coffey T., Puneet S., "Non-Repudiation with Mandatory Proof of Receipt", Computer Communications Review, Jan 1996, vol. 26, no. 1, pp 6.

[3] Zhou J., Dieter G., "A Fair Non-Repudiation Protocol", IEEE Symposium on Security and Privacy, 1996, Oakland, CA, pp 55 – 61.

Also in:

Zhou J., Dieter G., "An Efficient Non-Repudiation Protocol", Proceeding of the 10th IEEE Computer Security Foundations Workshop, Rockport, MA, June 1997, pp 126 –132.

[4] Fortezza Cards: FAQ. http://www.armadillo.huntsville.al.us/general/faq.html

[5] Chen M.I., Lin K.J., "A Priority Ceiling Protocol for Multiple-Instance Resources Proceedings of the IEEE Computer Society Real-Time Systems Symposium, Dec 1991, pp. 140-149.

[6] Noll Micheal A., "Internet Pricing vs. Reality", Communications of the ACM, vol. 40, no. 8, Aug 1997, pp 118 – 121.

Appendix A: *Potential R&D Tasks, Schedule and Cost*

We identify potential research tasks that may be conducted to address the SAP issues for an accountable multihop communication environment. The proposed research is in two parts: A) measurement approach for the severity of the SAP issue, and B) a framework for solution of the SAP issue to design an accountable multihop system. Summary of these R&D tasks are listed below, and further detail can be provided as needed.

Part A: SAP Measurement Tasks

Task 1.1: Identification of the SAP Metrics for a Generalized Distributed System
Task 1.2: Identification of SAP Metrics for a Domain Specific Application (e.g., JCALS, or DMS)
Task 1.3: Development of SAP Certification Level using Weighted-Value Model
Task 1.4: Development of SAP Certification Level using Logic and Rule-Based Model

Part B: SAP Solution Algorithms Tasks

Task 2.1: Development of a Direct-Exchange (D-E) Model for SAP Solution
Task 2.2: Development of Algorithms for the D-E Model Across Un-Trusted Internet Routers
Task 2.3: Development of Algorithms for the D-E Model Across Trusted Networks (including DISN, and/or Fortezza Card Equipped Systems)
Task 2.4: Development of Algorithms for the D-E Model Across COTS Operating Systems and Applications
Task 2.5: Implementation Algorithms for the D-E Model for Specific Environments
- Java, and Web-Browser Enabled Platforms
- Unix, and Remote Procedure Enabled Platforms

Schedule:

Tasks in parts A and B are inter-related, and thus an ideal combination would be to commence both groups of tasks. However, in principle they are independent, making the independent execution of tasks in parts A and B feasible, although not preferred.

- Task 1.1 must precede the other tasks in Part A. Tasks 1.2, 1.3, and 1.4 can be executed in parallel.

- Likewise, Task 2.1 must precede the other tasks in Part B. Tasks 2.2, 2.3, and 2.4 can be executed in parallel. Task 2.5 is an implementation task, oriented towards demonstration with real systems.

Cost: