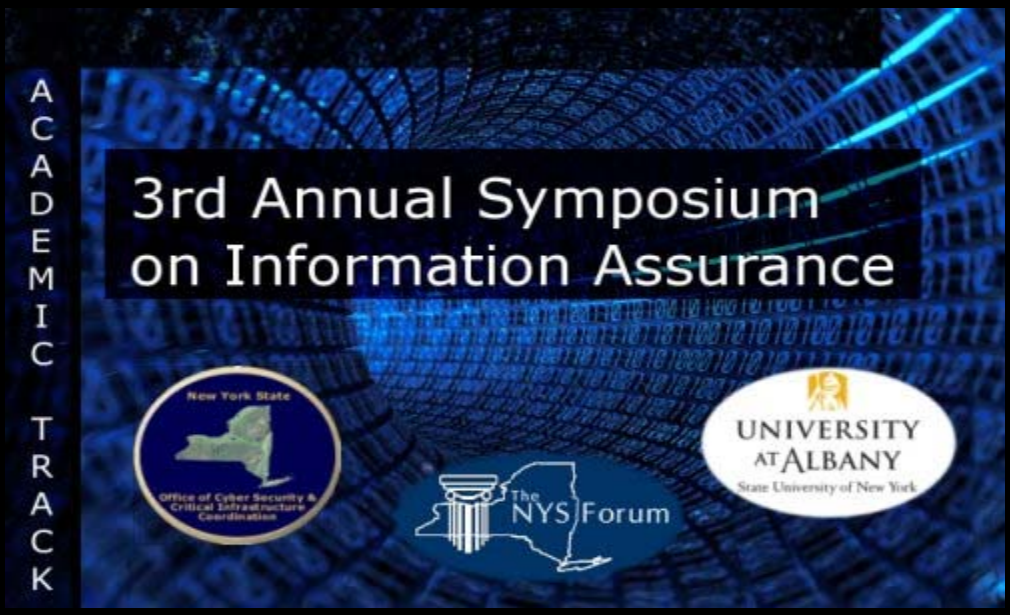


conference proceedings

3rd Annual Symposium on Information Assurance (ASIA '08)

Symposium Chair:
Sanjay Goel
Information Technology Management, School of Business
University at Albany, State University of New York

Academic Track of 11th Annual NYS Cyber Security Conference
Empire State Plaza Albany, NY, USA
June 4-5, 2008

Proceedings of the 3rd Annual Symposium on Information Assurance
Academic track of the 11th Annual 2008 NYS Cyber Security Conference
June 4-5, 2008, New York, USA.

Symposium Chairs

Sanjay Goel, Chair
Director of Research, NYS Center for Information Forensics and Assurance (CIFA)
Associate Professor, Information Technology Management, School of Business, University at Albany, SUNY

Laura Iwan, Co-Chair
State ISO, NYS Office of Cyber Security and Critical Infrastructure Coordination (CSCIC)

Program Committee

Anil B. Somayaji, Careleton University
Anna C. Squicciarini, Purdue University / Penn State
Arun Lakhotia, University of Louisiana at Lafayette
Bolek Szymanski, Rensselaer Polytechnic Institute
Daniel O. Rice, Loyola College in Maryland
Fred B. Schneider, Cornell University
George Berg, University at Albany, SUNY
Gurpreet Dhillon, Virginia Commonwealth University
Hong C. Li, Intel Corporation
Mark H. Linderman, Rome Labs
Martin Loeb, University of Maryland
Michael Sobolewski, Texas Tech University
Mohammed Zaki, Rensselaer Polytechnic Institute
Nasir Memon, Brooklyn Polytechnic
Paliath Narendran, Univeristy at Albany, SUNY
R. Sekar, Stony Brook University, SUNY
Raghu T. Santanam, Arizona State University
Rahul Singh, University of North Carolina, Greensboro
Raj Sharman, University at Buffalo, SUNY
Robert Ghanea-Hercock, BTexact Technologies
Ronald Dodge, USMA West Point
Shambhu Upadhyaya, University at Buffalo, SUNY
Shiu-Kai Chin, Syracuse University
Stelios Sidiroglou, Columbia University
Stephen F. Bush, GE Global Research Center

External Reviewers

Adnan Baykal, NYS Office of Cyber Security and Critical Infrastructure Coordination
Madhusudhanan Chandrasekaran, University at Buffalo, SUNY
Nicholas C. Weaver, International Computer Science Institute
Saeed Abu-Nimeh, Southern Methodist University

Submissions Chair

Damira Pon, University at Albany, SUNY

Note of Thanks

We would like to express our appreciation to all of the sponsors which supported the symposium.

CONFERENCE KILOBYTE SPONSORS

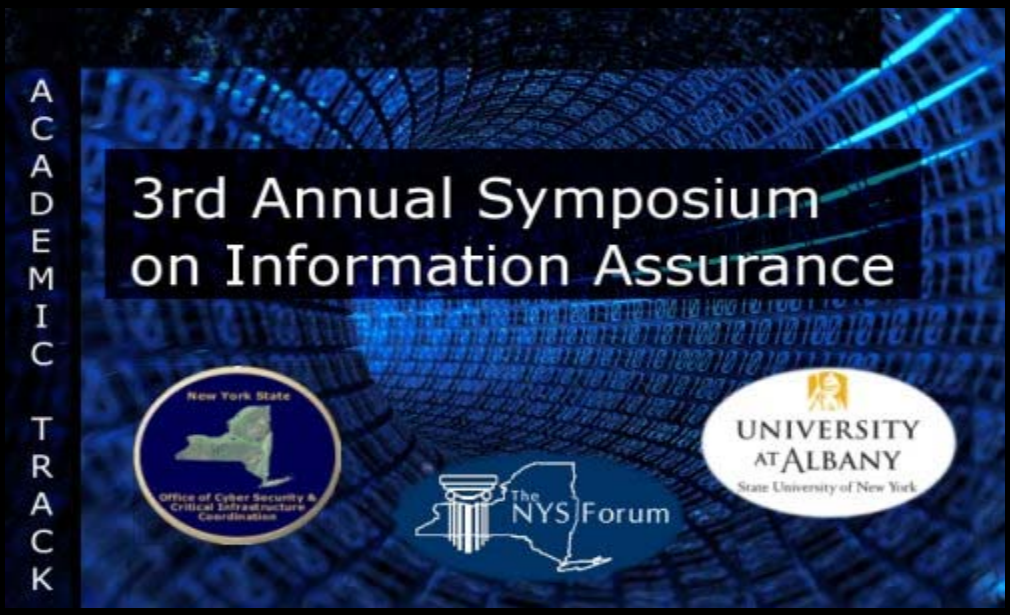
BearingPoint
FORTIFY SOFTWARE
Microsoft
ORACLE
serverware CORPORATION
symantec.

CONFERENCE MEGABYTE SPONSOR

D&D CONSULTING

SYMPOSIUM SPONSORS

UNIVERSITY AT ALBANY State University of New York
SCHOOL OF BUSINESS & COLLEGE OF COMPUTING AND INFORMATION



conference proceedings

3rd Annual Symposium on Information Assurance (ASIA '08)

Symposium Chair:
Sanjay Goel
Information Technology Management, School of Business
University at Albany, State University of New York

Academic Track of 11th Annual NYS Cyber Security Conference
Empire State Plaza Albany, NY, USA
June 4-5, 2008

MESSAGE FROM SYMPOSIUM CHAIRS

Welcome to the 3rd Annual Symposium on Information Assurance (ASIA'08)! This symposium complements the NYS Cyber Security Conference as its academic track with a goal of increasing interaction among practitioners and researchers to foster infusion of academic research into practice. For the last two years, this symposium has been a great success with excellent papers and participation from academia, industry, and government and highly attended sessions. This year, we again have an excellent set of papers, invited talks, and keynote addresses.

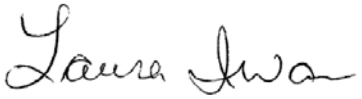
Our keynote speakers this year are Billy Rios, a phishing expert from Microsoft Corporation and John Crain, the Chief Technical Officer from ICANN (Internet Corporation for Assigned Names and Numbers). The symposium has papers in multiple areas of security, including web/email security, distributed security management, nanosensor security, security governance, application security, and internet security. We hope to include selected papers from this symposium in a journal special issue. We have also added a roundtable discussion on forensics training and education to the program this year based on strong participant interest.

We would like to thank the talented program committee that has supported the review process of the symposium. In most cases, the papers were assigned to at least three reviewers who were either members of the program committee or experts outside the committee. We ensured that there was no conflict of interest and that each program committee member was not assigned to review more than two papers. We also personally read the papers and the reviews and concurred with the assessment of the reviewers. For this year's symposium, we have 8 refereed papers and two invited papers. Our goal is to keep the quality of submissions high as the symposium matures. The program committee serves a critical role in the success of the symposium and we are thankful for the participation of each member.

We were fortunate to have extremely dedicated partners in the NYS Office for Cyber Security and Critical Infrastructure Coordination (CSCIC), the NYS Forum, and the University at Albany, State University of New York (UAlbany). Our partners have managed the logistics for the conference, allowing us to focus on the program management. We would like to thank the University at Albany's School of Business and College of Computing and Information for providing financial support for the symposium.

We hope that you enjoy the symposium and continue to participate in the future. In each of the subsequent years, we plan to have different themes in security-related areas. Next year, the symposium will be held on June 3-4, 2009. If you would like to propose a track, please let us know. The call for papers for next year's symposium will be distributed in the fall and we hope to see you again in 2009.

Sanjay Goel
Director of Research, CIFA
Associate Professor, School of Business



Laura Iwan
NYS ISO, Office of Cyber Security and Critical Infrastructure Coordination (CSCIC)

SYMPOSIUM ON INFORMATION ASSURANCE AGENDA

DAY 1: Wednesday, June 4 2008

REGISTRATION - Base of the Egg and VISIT EXHIBITORS – Convention Hall (7:00am – 5:30pm)

SYMPOSIUM SESSION 1: Opening & Keynote - Mtg. Rm. 7 (8:00am – 9:15am)

Introduction to Symposium: Sanjay Goel, *Symposium Chair*
Opening Remarks: Paul Leonard, *Dean, School of Business, University at Albany, SUNY*
Keynote Address (D-1): Phishing Underground: A New Look at an Old Problem
Billy Rios, *Microsoft Corporation, Redmond, WA*

MORNING BREAK & VISIT EXHIBITORS - Convention Hall (9:15am - 9:30am)

CONFERENCE MORNING SESSION – Egg Swyer Theater (9:30am – 10:15am)

Live Hacking Demo: Sanjay Goel, *School of Business and NYS Center for Information Forensics and Assurance at the University at Albany, SUNY*

CONFERENCE OPENING TALKS – Egg Swyer Theater (10:15am – 11:45am)

Welcome: Will Pelgrin, *Director, CSCIC*
Introduction: Peter Bloniarz, *Dean College of Computing and Information, University at Albany, SUNY*
Keynote: Patrick Gray, *Cisco Systems, Inc.*

LUNCH ON YOUR OWN (11:45am – 12:45pm)
VISIT EXHIBITORS – Convention Hall

SYMPOSIUM SESSION 2: Web/Email Security (12:45pm – 2:00pm)
Chair: Bolek Szymanski, *Rensselaer Polytechnic Institute*

When Elephants Dance, Mice Must be Careful: Content Provider Conflict on the Modern Web
Terri Oda, Anil Somayaji, and Tony White, *Carleton University, Canada*

CUSP: Customizable and Usable Spam Filters for Detecting Phishing Emails
Madhusudhanan Chandrasekaran, Vidyaraman Sankaranarayanan, and Shambhu Upadhyaya, *State University of New York at Buffalo*

VISIT THE EXHIBITORS – Convention Hall (2:00pm – 2:45pm)

SYMPOSIUM SESSION 3: Distributed Security Management (2:45 – 4:00)
Chair: Raj Sharman, *State University of New York at Buffalo*

Secure Space Computing with Exertions
Daniel Kerr and Michael Sobolewski, *Texas Tech. University*

Peer-to-Peer Simulation for Network Security
Daniel O. Rice and George Wright, *Loyola College in Maryland*

ATTENDEE RECEPTION – Convention Hall (4:00 pm – 5:30pm)

SYMPOSIUM ON INFORMATION ASSURANCE AGENDA, CONT'D.

DAY 2: Thursday, June 5 2008

REGISTRATION - Base of the Egg and VISIT EXHIBITORS – Convention Hall (7:30am – 3:45 pm)

SYMPOSIUM SESSION 4: Award & Keynote – Mtg. Rm. 7 (8:00am – 9:15am)

Best Paper Award Presentation: Laura Iwan, *Symposium Co-Chair*

Keynote Address (D-2): Securing the Internet Infrastructure: Issues and Problems
John Crain, *Chief Technical Officer, ICANN, Marina Del Rey, CA*

ROUNDTABLE: FORENSICS TRAINING & EDUCATION (9:15 am – 10:00am)
Moderator: Sanjay Goel, *University at Albany, SUNY*

Panelists: Fabio Auffant, *NYS Police Computer Crime Laboratory*, Christian Balan, *Champlain College in Burlington*, and Sean Smith, *New York Prosecutors Training Institute*

MORNING BREAK – Convention Hall (10:00 am – 10:15am)

SYMPOSIUM SESSION 5: Invited Talks (10:15am – 11:30am)
Chair: Anil Somayaji, *Carleton University*

On Information Assurance in Nanoscale Networks
Stephen F. Bush, *General Electric Global Research*

An Analysis of Information Security Governance Structures: The Case of Société Générale Ifeoma Udeh and Gurpreet Dhillon, *Virginia Commonwealth University*

LUNCH ON YOUR OWN (11:30am – 12:30pm)
VISIT EXHIBITORS – Convention Hall

SYMPOSIUM SESSION 6: Application Security (12:30pm – 1:45pm)
Chair: George Berg, *University at Albany, SUNY*

Content-sensitive, Temporally Adaptive Metadata
Brendan J. Gilbert, Raj Sharman, Manish Gupta, H.R. Rao, Shambhu Upadhyaya, and Kenneth P. Mortensen, Esq., *University at Buffalo, SUNY*

Recursive Data Mining for Author and Role Identification
Vineet Chaoji, Apirak Hoonlor, & Boleslaw Szymanski, *Rensselaer Polytechnic Institute*

VISIT THE EXHIBITORS – Convention Hall (1:45pm – 2:30pm)

SYMPOSIUM SESSION 7: Internet Security (2:30pm – 3:45pm)
Chair: Shobha Chengalur-Smith, *University at Albany, SUNY*

Enhancing the Non-Repudiation Properties of EMV Payment Cards
David J. Boyd, *Information Security Group, Royal Holloway, University of London*

Formal Methods for Intrusion Detection of Windows NT Attacks
Sahika Genc, *Sensor Informatics Technology Laboratory, GE Global Research*

CLOSING REMARKS (3:45pm – 4:00pm)
Sanjay Goel, *Symposium Chair*

TABLE OF CONTENTS

Keynote (D-1): Phishing Underground: A New Look at an Old Problem

Billy (BK) Rios, Security Engineer
Microsoft Corporation, Redmond, WA

This talk will expose the tools and tactics used by the phishing underground. It's really a new look at an old problem. Follow us as we track real life phishers hiding in the shadiest corners of the Internet, analyze the tools used by phishers, see how phishers "phish" other phishers, and discover the sites where real life identities are being bought and sold. The specific topics covered by this talk will include: how phishers set up a phishing site, a look at the backdoors used by phishers, determining how phishers get identity information, a thorough look at the tools used by phishers, and a detailed look at the sites used to buy and sell stolen identities.

Content Provider Conflict on the Modern Web

Terri Oda, Anil Somayaji, Tony White
School of Computer Science, Carleton University
Ottawa, Ontario, Canada
{toda,soma,arpwhite}@scs.carleton.ca

***Abstract*—Today many web pages include externally sourced content. Advertisements, video, blog "trackbacks," search—these and other features of the modern web are provided by third-party servers. Such external content is so popular that content is often incorporated from more than one source. In this paper we argue that such multiple inclusions are a significant security risk because of the potential for conflict between included elements. In particular, the use of JavaScript to provide external content means that providers can observe and interfere with each other. Financial incentives and competitive advantage provide motivation for such conflicts, both for criminals and for legitimate enterprises. To prevent users and web content providers from becoming collateral damage, we must develop and deploy practical techniques for isolating externally provided web content. This paper outlines the security threat posed by combining content from different providers and describes requirements for a solution.**

I. Introduction

From the beginning of the World Wide Web, HTML pages have been composite documents, incorporating elements from multiple sources. Early pages mostly used text and images from a single web server; modern web pages, however, include content from multiple organizations. Some of these inclusions provide functional enhancements such as search services, blog "trackback" links, and video players; others supply the advertisements that are the economic foundation of much of the web.

Standard HTML4 mechanisms for incorporating external content (such as the `img` and `embed` tags) restrict them to a portion of a page: they can only be displayed within a box within the page, they cannot observe the rest of the page, and they can only receive user input when mouse or keyboard events are directed to them. Many web developers, however, have found these mechanisms to be too restrictive for dynamic content, and so they have turned to another Web technology: JavaScript.

The most common mechanisms for including external content today require the web page author to incorporate a small fragment of boilerplate JavaScript code. This code will typically load more code from a third party server; this additional code is what provides the actual functionality. Unlike HTML-based inclusion mechanisms, included JavaScript has full access to a page: all of the content and all of the events.

Many have recognized that such inclusions could represent a security threat, particularly if the external JavaScript code is compromised (e.g., [3]). Others have recognized the particular dangers of web mashups—web applications that combine together ("mash up") two or more existing web applications or pages [12], [10], [7]. What has not been appreciated, however, is that the common case—inclusion of content from multiple providers in an "ordinary" web page—itself constitutes a security risk. The risk comes from the opportunities and incentives for conflicting code.

Specifically, the commercial agendas of external content providers seldom align; however, all JavaScript code is considered trusted within the confines of a page: each piece of JavaScript can access all of a page's code and data. Thus, it is possible for one content provider to manipulate the code and data included from another. This manipulation can be used to degrade service, divert advertising revenue, and conduct click fraud.

While current proposals for securing JavaScript in web mashups can help secure included content in certain circumstances, they break important uses such as context-sensitive advertisements (such as Google Adwords) while introducing usability issues for unsophisticated web developers. Thus, we believe that new solutions are needed for securing external web content.

This paper has two key contributions. The first is identifying the security threat caused inclusion of content from multiple content providers. This threat is made more dangerous by the assumption that content providers will interact only in safe ways (or not at all), as well as the assumption that most web pages are not in need of protections currently reserved for more complex web applications. Such assumptions can lead to inappropriate security decisions or design of systems which do not easily address the full scope of the problem. Because of the risks involved in such assumptions, the second contribution here is in outlining the requirements for a solution to this problem.

The rest of this paper proceeds as follows. In Section II, we explain in more detail how external content is included in web pages. Section III describes the standard security restrictions on JavaScript and their limitations. We explore the idea of content providers being adversaries in Section IV, including specific attack scenarios. Some requirements for a solution are discussed in Section V. In Section VI, we present related work in web security including work on web mashups. Section VII discusses the opportunities and challenges for better JavaScript isolation mechanisms and Section VIII concludes.

II. Web Page Composition Using Multiple Sources

Most webpages are constructed using information from several sources. Sites that have content they want others to

```
1 <object width="425" height="355">
2 <param name="movie" value="http://www.youtube.com/v/FiARsQSlzDc">
3 </param>
4 <param name="wmode" value="transparent"></param>
5 <embed src="http://www.youtube.com/v/FiARsQSlzDc"
6          type="application/x-shockwave-flash" wmode="transparent"
7          width="425" height="355">
8 </embed></object>
```

Listing 1. Code for including a video on a web page, as generated by YouTube. Note that the information about the URL for the video is repeated both as a parameter within the object tag (line 2) and inside the embed tag (line 5). This is to ensure compatibility with more browsers, as some use the object tag and others use embed.

Figure 1. Inclusion of an image into an HTML document results in a predictable webpage

include will often give fragments of HTML code that users can put in their page. Although browsers still vary in how they render a page, this is the easiest way to assure that anyone who wants to can include this content, be it an image, a video, or something else.

A. Including Static Content

In the most basic of HTML, there are many ways to include static content that will be the same every time the content is viewed. For example, video site YouTube generates code for people to embed video objects in their pages, as shown in Listing 1.

Here, the pertinent part is the `object` or `embed` tag which includes a flash video from YouTube into the page. Both tags are provided because some browsers only understand one or the other. The web browser reads the HTML, goes to get the video, and inserts this video into the page. It only inserts the video in where this tag was found. Images inserted with the `img` tag work the same way.

The path to content inclusion is shown in Figure 1. Here, you can see that the web page, combined with simple content such as the image shown, behaves in a predictable way, inserting the image where expected on the page.

B. Including JavaScript content

JavaScript is often used to generate content dynamically. Advertisements are a good example of this. Consider the code provided by Google for inserting an AdSense advertisement onto a web page, as described in Listing 2. Here, we have a small piece of JavaScript code which contains a few settings, followed by a link to more JavaScript code. This code then actually produces the advertisement which is to be placed on the page.

Note that there is no indication of where the advertisement should be placed. The very act of including this code allows it read and write access to the entire page. The included JavaScript code can choose to place the advertisement anywhere it deems suitable. In practice, it will place the content where the web page creator included the code, since this is the way things usually work with static content as described in Section II-A. However, this placement is not guaranteed—it is merely a convention.

Scripts are included as source, and often multiple scripts are included in the same page. Script sources are evaluated in the same context as the main page: the expectation is that the code of included scripts will not interfere with each other. Multiple inclusions work because developers respect conventions; the browser enforces no separation.

Figure 2 gives a visual representation of what could happen when JavaScript is included into a page. Unlike Figure 1, the result of this action is unpredictable. Figure 2a shows what one might expect the code to look like given JavaScript code from an advertiser: the code only adds an advertisement image into the box provided. However, we can see in Figure 2b that the JavaScript could be used to add content to a page, say to insert multiple advertisements. Finally, Figure 2c shows that JavaScript code could also be used to delete the contents of entire page. In practice, typical JavaScript from external parties does not make drastic changes to a page's appearance; however, as part of providing services such as visitor statistics and context-sensitive advertisements, included code commonly accesses virtually all parts of the including web page.

While there are clearly issues with giving external entities this level of control, there are limits placed on the functionality of JavaScript that address many security concerns. We discuss these features below.

```
<script type="text/javascript"><!--
google_ad_client = "pub-6828282629126141";
/* 728x90, created 3/6/08 */
google_ad_slot = "5248526188";
google_ad_width = 728;
google_ad_height = 90;
//-->
</script>
<script type="text/javascript"
        src="http://pagead2.googlesyndication.com/pagead/show_ads.js">
</script>
```

Listing 2. A sample advertisement inclusion (Google AdSense).

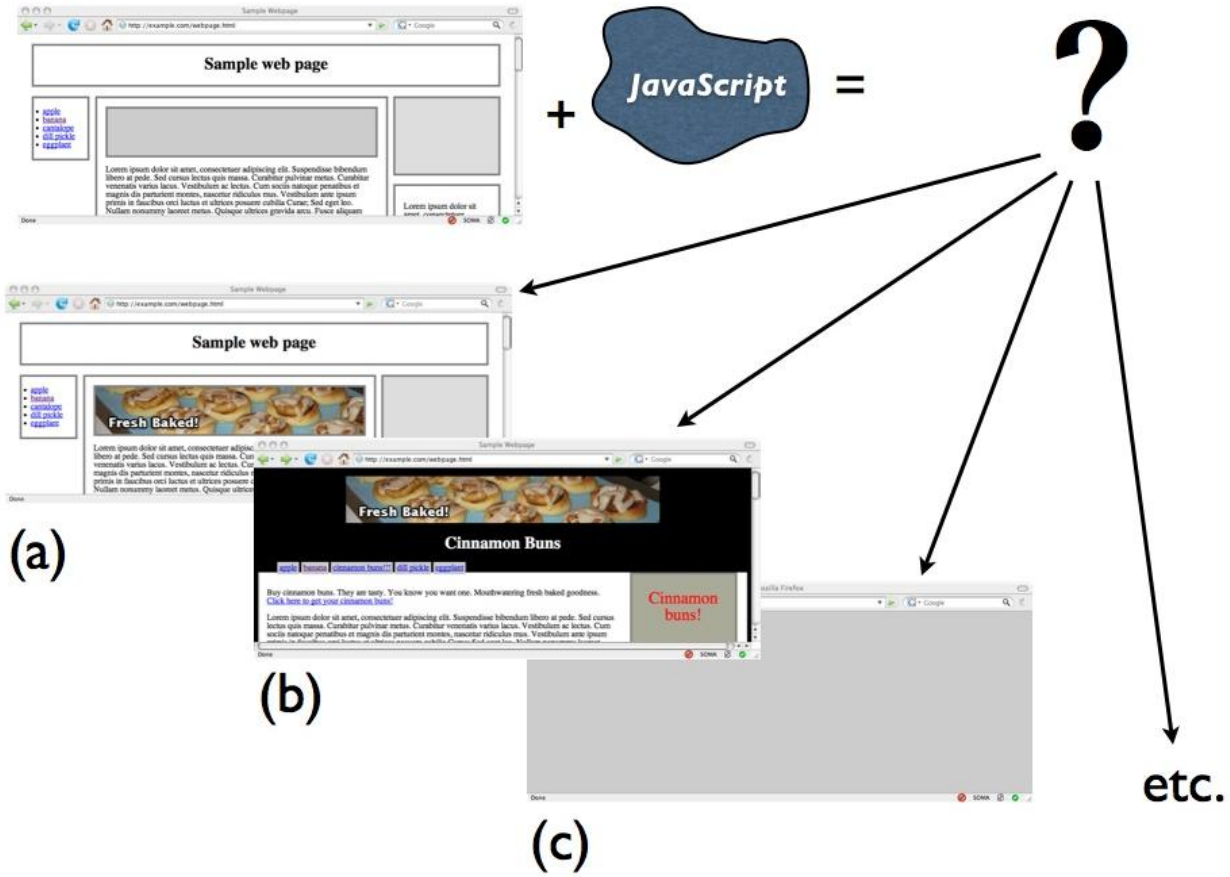


Figure 2. Inclusion of JavaScript in an HTML document leads to unpredictable results. (a) looks as one might expect given code from an advertiser: the code places an image advertisement in the box provided for the advertisement. (b) shows another possibility where the advertiser decides to modify the existing page, deleting segments, changing others to be more favorable to their advertisement. (c) shows a case where the JavaScript has replaced the page with a simple blank one.

III. JavaScript Security

JavaScript was designed to be a programming language for the web. Since web pages are provided by untrusted sources the goal was to create an environment such that code on a web page should not be able to harm web users, their computers, or other Internet hosts. To provide these guarantees, JavaScript enforces an execution sandbox. This sandbox is designed to isolate programs from each other and from the underlying operating system. Like Java applets (the first Web programming framework to employ a sandbox), the JavaScript sandbox prevents programs from accessing raw memory, the contents of other loaded web pages, and local files [4]. While elements can be incorporated from any remote resource that can be described by a URL, this content is also isolated to the including web page, thus preventing many forms of attack.

Elements from the same site residing on different pages or frames, however, often need to interact to exchange information. Thus, the JavaScript sandbox is relaxed in the case of documents originating from the same domain. Thus, JavaScript code in one document can manipulate the state of another document with the same origin. This same origin exception is routinely used to implement multi-pane interfaces, pop-up windows, and more complex AJAX-based sites such as Google Maps. While it is possible for the JavaScript sandbox to be subverted by exploiting browser flaws [1], to a large extent it succeeds in accomplishing its design goals. Our concern, however, is that malicious interactions can occur within a given page's sandbox.

The JavaScript supports powerful mechanisms for code and data separation, but these are undermined by unfettered access to the global environment and the Document Object Model (DOM). The DOM contains references to all document text, top-level functions, global variables and objects—essentially everything that is in a web page. DOM objects may be accessed through a number of global variables such as `document`. Any named node can be accessed using a call to `document.getElementByID()`. Each specific node may have its properties inspected, modified, or deleted.

All JavaScript code, including imported code, can read from and write to the global environment and, by extension, to all parts of the Document Object Model (DOM). Variables can be overwritten, functions substituted, and page elements can be read and changed arbitrarily. For example, a named link could have its color inspected, thus exposing whether or not that link had been previously visited by the user. Alternately, a node could be removed entirely from the document tree entirely, thereby compromising the document displayed.

The reason why imported code can access all global variables and functions is that all imported code and data is included in page's global JavaScript context (environment); there is so separation between imported elements and code embedded in the page (see Figure 3. This lack of separation is what allows content providers to come into conflict. For example, code from content-A.com can change attributes associated with `document` and these affect what is seen by

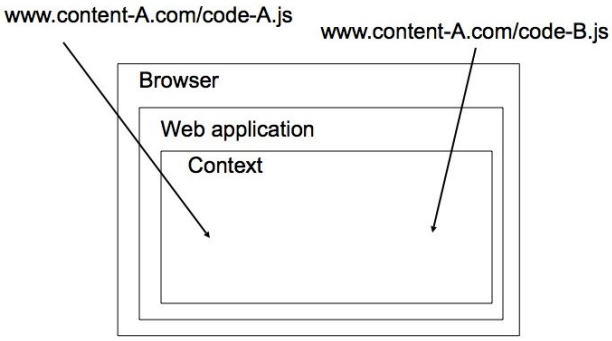


Figure 3. JavaScript included from external sources share the including application's global execution context (environment).

code included from content-B.com. Furthermore, a function named `foo` loaded from content-A.com will be overwritten by a function of the same name when content is loaded from content-B.com. This conflict arises because the top-level namespace is shared. To see the full extent of the problem, refer again to Figure 2b in which the outcome of code inclusion is undecidable. While it is expected that advertisers will only write in the box where the advertisement is intended to be displayed, there is nothing stopping them from doing other things to the page—including targeting other content providers.

To summarize, the current model for web document creation allows code and data to be included from several sources. The assumption is that all code is equally trusted and should be integrated with the same rights and privileges as the document that caused it to be included. Controlled interaction between sources—other than the same origin policy—is not provided. This inclusion of code and data can be thought of as a type of code mobility, a form of distributed computing. As with other code mobility systems, inappropriate trust relationships lead to a number of potential problems. We discuss these problems in the context of content provider conflict below.

IV. Content Providers as Adversaries

Much attention has been focused on the problem of cross site scripting (XSS), an attack in which someone injects malicious code, usually JavaScript, into a page [14]. XSS is usually accomplished by taking advantage of a bug in the input checking of a web application. Attackers who are trusted content providers, however, do not need to exploit software bugs to inject malicious JavaScript—they already have the required access.

However, why would one content providers want to interfere with another? Consider that many content providers are competitors. For example, they may both provide advertising services, they could both serve up video content, or they might both provide fast servers for accelerating delivery of web content. This inclusion of code from competitors happens on real pages. For example, CNN's website includes advertisements from advertisement.com (an AOL subsidiary), yet its search functionality is provided by Google. Salon includes

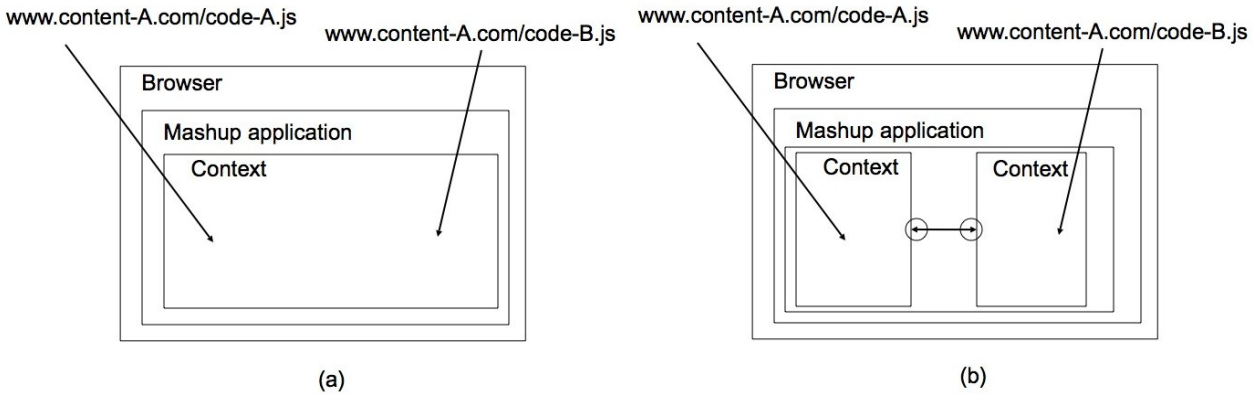


Figure 4. Current (a) and proposed (b) architectures for web mashup applications.

JavaScript code from Yahoo's Overture, Google Analytics, and other smaller advertisers and media providers. Indeed, it seems that virtually every major website that isn't owned by Google, Microsoft, or Yahoo includes content from competing organizations.

While it is likely that virtually all current external content providers are playing "nicely" with each other, we should not expect that those relationships will remain so civilized. Indeed, as the Sony rootkit debacle showed [20], major corporations are perfectly capable of using malicious software to further their interests.

Broadly speaking, there are three fundamental goals that can be achieved by manipulating JavaScript behavior within the sandbox: observation, content manipulation, and server manipulation. Here we expand on these attack strategies by exploring how a content provider A could target another provider B when they both have code resident on a page X.

A. Observation

One key task for any organization is to monitor its competition. By manipulating JavaScript, it is possible for one content provider to observe what the other is doing. For example, if A wanted to know what text advertisements B was presenting on a page, A could copy those advertisements into a separate variable (by walking the DOM tree) and then send them to another server using an HTTP GET or POST command.

Note that these are the same mechanisms used to generate customized advertisements and to gather statistics on site visitors. The only difference here is the subject of observation.

B. Content Manipulation

It would also be possible for provider A to directly manipulate B's code and data. Since there are no protection boundaries and every part of the global environment is accessible by all JavaScript code, all A needs to do is overwrite or override B's variables. One simple attack would be to delete B's content. This would deny B advertising revenue; however, it would also deny X's owner revenue as well. Another attack would be to rewrite displayed advertisements to make them less appealing, thus reducing B's click-through rate. A could also replace B's advertisements with A's.

A more sophisticated attack would be for A to adjust the attack based upon what code B has included. By walking the DOM, A's JavaScript can analyze any of B's code that was part the document. While imported JavaScript cannot be directly analyzed in this way, A's server can grab B's JavaScript file, analyze it, and return instructions on what variables to change to A's JavaScript in X—all by using a single GET request.

Of course, since there is no restriction on A's access to the page, A could choose to modify other parts of X. A could cause B to display inappropriate advertisements by inserting (hidden) content into the page. More sophisticated content-based attacks are also possible such as content censorship; to avoid detection, however, they would need to be very constrained and targeted.

C. Server Manipulation

Rather than manipulating B's JavaScript, A could instead send messages to B's servers using the state present in the web page. Note that A's JavaScript can do anything that B's JavaScript can do; thus, A can impersonate B's code to B's servers. A could do this to give false information about page X (e.g., misreport the content of X) to B, or to generate false clicks on B's advertisements (i.e., perpetrate click fraud).

V. Requirements

This section attempts to create a set of goals for web security solutions addressing the problem of content provider conflict. There are 4 goals: ease of use for all web page creators, clear adoption path, isolation between content providers, and flexibility for future innovation in web applications.

A. Ease of use for all web page creators

Few people would choose to design security solutions which are unusable, but sometimes it can be unclear who the intended users are and what skills they have. Early web mashup solutions such as Subspace [10] rely heavily upon skilled web programmers who could produce secure web pages. This makes sense when you are considering securing complex web applications, which are often created by skilled programmers. However, we have seen that there is risk any time code from

multiple providers is included on a page. Thus, any blog that contains cut and pasted ad code, videos, etc. could be at risk, and many people who use the simple cut and pasted code are not programmers at all.

The ideal solution to content provider conflict must take this common use-case into account. For example, one could provide secured code which can be cut and pasted with the code fragments currently in use, or perhaps encourage tools which automatically generate more secure code. Or, perhaps, the solution lies in involving the page creator as little as possible since their skill set and preferred tools cannot be predicted.

B. Clear adoption path

Deployment of any solution is important, and a clear, feasible adoption path is needed to go past the research sphere into actual use on the web. The web is a distributed and heterogeneous environment, and it is the diversity that presents many problems to adoption. There are various types of web server and browser in use, controlled by many different individuals and organizations. Changing all of them at once is infeasible. Similarly, caution must be used when making changes to JavaScript or other web languages, and at least partial backwards compatibility or a way to deal with older websites could potential ease the pain of adoptions. The ideal solution would put some thought into these issues and examine ways in which deployment could be achieved.

C. Isolation between content providers

If the problem is that content providers have too much access to other content providers' code and content, then the solution is to limit this access by providing isolation between components. The ideal solution will block the three attacks described previously: observation, content manipulation, and server manipulation.

D. Flexibility for future innovation

Although it is difficult to predict the future, solutions should try to plan for it by giving flexibility and allowing for innovation in web applications. The current model, while it now seems overly permissive, has given us the ability to make applications that were unheard of when the web was created. The ideal solution would solve current use-cases without limiting itself to only those known cases.

VI. Related Work

Although concerns about the security of JavaScript are as old as the language itself, only more recently have security researchers begun really exploring the kinds of attacks that JavaScript makes possible. While web security issues such as drive-by downloads [16], [15], cross-site scripting [13], and cross-site request forgery [11] do not require JavaScript, JavaScript does make these and other attacks more potent and easier to execute. Many security practitioners recommend that users disable JavaScript in their browsers entirely [2] or on a per-domain basis [8]; because so many pages require JavaScript to render correctly, such "solutions" are not practical for most users. Thus, even though such a solution would successfully block attacks, it can hardly be considered to be deployable.

JavaScript itself already has powerful mechanisms for code and data separation, as it has an environment-based lexical scoping model for variable and function binding. This model supports full closures, and thus provides very powerful mechanisms for code and data separation. These built-in abilities may be very helpful when it comes to finding a solution, but at the moment they are seldom-used. This is perhaps due to the fact that code is often written by those who have little understanding or interest in security. If things could be arranged so that using these mechanisms were the easiest route to writing JavaScript, it is possible that they would be utilized more effectively.

Some have argued for a more comprehensive approach to JavaScript security [18]; others have focused on scanning web pages for dangerous forms of JavaScript [19], [13], [11], [6], [21]. One limitation of code approaches is that they cannot restrict the regular behavior of external content providers such as ad servers even when they are potentially dangerous. Indeed some solutions must explicitly whitelist ad servers in order to achieve acceptable performance [21], [8]. As such, many techniques are unable to solve the problems of conflicting trusted content providers since they assume that all providers are trusted.

Currently there is an ongoing battle between advertisers and criminals. Advertisers regularly lose money to click fraud schemes in which criminals fake legitimate user behavior [9], and research indicates that some click fraud strategies can be extremely subtle and hard to detect [3]. Criminals are also taking advantage of the access ad servers have to regular users; indeed, even major companies such as DoubleClick [5] and Microsoft [17] have been tricked into distributing advertisements containing malware. In this environment it is clear that users have reason to be wary of ad servers. What we have argued here is that content providers such as ad servers need to be wary of each other as well.

A web mashup is an application that combines code and data from more than one source into a single integrated tool. In order to make a mashup application today, as shown in Figure 4a, JavaScript from multiple sites must be imported into a single environment. As explained in the previous section, this construction raises significant security issues.

Several researchers have proposed secure mechanisms for building web mashups. The goal of such systems is to achieve something equivalent to Figure 4b. Here, code and data from different sources are loaded into separate contexts; they neither share the same DOM objects, nor do they share the same namespace. Using the example from Section III, a function `foo` would be defined twice, once in the context from A and once in the context from B. For A and B to communicate, they must use well-defined communication channels using mutually agreed-upon protocols. Note that this architecture is virtually identical to that used by most distributed computing platforms.

Current secure web mashup proposals achieve separation and message passing either by requiring significant architectural changes to web applications [12], [10] or by using JavaScript language extensions that must be implemented in web browsers [7]. By themselves, these requirements make current secure mashup solutions unappealing for regular web pages. Even worse, these proposals would restrict the ability of content providers to access the body of the including page, thus breaking most popular site statistics services and context-sensitive advertisements. Thus, while secure web mashup mechanisms could be used to prevent content provider conflict, usability and functionality limitations would have to be addressed before they could be widely deployed.

VII. Discussion

When looking at the potential for conflict between external content providers, we do not mean to imply that we expect most providers to engage in open warfare on the web; instead, we are simply pointing out that there are conflicting agendas, and there is nothing to prevent those conflicts from manifesting as real attacks. While one type of content provider conflict, click fraud, is a significant problem, currently it is only perpetrated by known criminals. What happens, though, when legitimate businesses go bad?

The problem is somewhat analogous to that faced by companies that outsource to multiple partner companies. Each partner must be given access to the company's IT infrastructure; that access must be limited, however, in order to minimize the risks to the company. On the web, page authors outsource key parts of their "business" to outside parties, but do so without restricting their behavior. Social norms and legal measures can deter bad behavior to some degree; unfortunately, the global nature of the Internet means that we cannot rely upon individual societies or governments to enforce those norms. In operating systems, we long ago realized that memory protection made for more robust and secure systems. The question is, how do we bring analogous protections into the JavaScript sandbox?

As with most security problems, a point solution is not the answer; rather, layers of defense are required. Some defenses, such as code obfuscation and tamper resistance, could be deployed by the content providers themselves. Such measures, however, are partial at best; if conflict were to become common, an arms race would follow that would, at a minimum, degrade the experience of regular web users through broken and slowly executing web pages.

Work on more secure web mashups is an important step forward. However, the requirements for mashups and for content providers are different: where mashups require channels of communication between content in different frames, many content providers advertisements need to be able to analyze or even change the contents of a page in order to provide their services. We believe that protections from interactions between external content providers is an important area for future work.

One approach to providing such protections would be to adopt programming patterns and mechanisms to expose necessary content to external JavaScript without permitting unfettered access to the DOM. Another approach would be to enhance the browser such that it can automatically recognize and enforce appropriate boundaries between included JavaScript. Whatever the approach, the challenge is always to get the necessary buy-in from content providers, tool providers, web developers, and web users. Given the potential for problems, however, there may be sufficient motivation for a major change in the way web content is created and interpreted.

VIII. Conclusion

Including JavaScript code from multiple content providers is a potential source of security vulnerabilities. Such JavaScript can interact to allow surreptitious observation, content manipulation, and server manipulation. Although generally not a problem today, these interactions place external content providers at risk from both criminals and competitors. While there exist techniques for protecting against cross-site scripting, cross-site request forgery, and other web attacks, we lack mature methods for regulating interactions within the JavaScript sandbox. While work on protections for web mashups are an important step forward, further work is needed to find solutions that handle some very common use cases for the web, including context-sensitive ads and pages created by people who are simply pasting in code fragments provided by others (as opposed to created by skilled web programmers intending to create a complex web application). When creating these solutions, designers need to consider the needs of a wide range of web page creators, produce a clear adoption path so that their solution is not prohibitive to deploy, achieve separation between content providers who should not interact, and keep in mind not only current use cases, but also future innovation.

Acknowledgments

We thank the members of Carleton Computer Security Laboratory and the anonymous reviewers for their suggestions.

This work was supported by the Canada's National Sciences and Engineering Research Council (NSERC) through their Postgraduate Scholarship program (TO) and Discovery Grant program (AS & TW). In addition, Research in Motion (RIM) has provided support for our research in Web security.

References

[1] "Symantec internet security threat report," Symantec, Tech. Rep. XII, September 2007.

[2] CERT® Coordination Center, "Frequently asked questions about malicious web scripts redirected by web sites," CERT, Tech. Rep., 2004. [Online]. Available: http://www.cert.org/tech_tips/malicious_code_FAQ.html

[3] M. Gandhi, M. Jakobsson, and J. Ratkiewicz, "Badvertisements: Stealthy click-fraud with unwitting accessories," vol. 1, no. 2. Taylor & Francis, 2006, pp. 131–142.

[4] L. Gong, M. Mueller, H. Prafullchandra, and R. Schemers, "Going beyond the sandbox: An overview of the new security architecture in the Java Development Kit 1.2," in *USENIX Symposium on Internet Technologies and Systems*, 1997.

[5] D. Goodin, "Doubleclick caught supplying malware-tainted ads," *The Register*, November 13 2007.

[6] O. Hallaraker and G. Vigna, "Detecting malicious javascript code in mozilla," in *Engineering of Complex Computer Systems, 2005. ICECCS 2005. Proceedings. 10th IEEE International Conference on*, 2005, pp. 85– 94.

[7] J. Howell, C. Jackson, H. Wang, and X. Fan, "Mashupos: Operating system abstractions for client mashups," in *Proceedings of the Workshop on Hot Topics in Operating Systems*, May 2007.

[8] InformAction, "Noscript." [Online]. Available: http://noscript.net/

[9] N. Ives, "Web marketers fearful of fraud in pay-per-click," *The New York Times*, March 3 2005.

[10] C. Jackson and H. J. Wang, "Subspace: Secure cross-domain communication for web mashups," in *Proceedings of the 16th International World Wide Web Conference (WWW2007)*, Banff, Alberta, May 8-12 2007.

[11] N. Jovanovic, E. Kirda, and C. Kruegel, "Preventing cross site request forgery attacks," in *2nd IEEE Communications Society International Conference on Security and Privacy in Communication Networks (SecureComm)*. Baltimore, MD: IEEE Computer Society Press, August 2006.

[12] F. D. Keukelaere, S. Bhola, M. Steiner, S. Chari, and S. Yoshihama, "Smash: Secure cross-domain mashups on unmodified browsers," IBM Research, Tokyo Research Laboratory, IBM Japan, Ltd., Tech. Rep. RT0742, June 11 2007.

[13] E. Kirda, C. Kruegel, G. Vigna, and N. Jovanovic, "Noxes: A client-side solution for mitigating cross site scripting attacks," in *The 21st ACM Symposium on Applied Computing (SAC 2006), Security Track*, Dijon, France, April 2006.

[14] T. Oda, G. Wurster, P. V. Oorschot, and A. Somayaji, "SOMA: Mutual approval for included content in web pages," School of Computer Science, Carleton University, Tech. Rep. TR-08-07, 2008.

[15] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The ghost in the browser: Analysis of web-based malware," *Workshop on Hot Topics in Understanding Botnets (HotBots), April*, vol. 10, 2007.

[16] N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose, "All your iframes point to us," Google, Tech. Rep. provos-2008a, February 4 2008.

[17] J. Reimer, "Microsoft apologizes for serving malware," *ars technica*, February 21 2007.

[18] C. Reis, S. Gribble, and H. Levy, "Architectural principles for safe web programs," in *Sixth Workshop on Hot Topics in Networks (HotNets) 2007*, 2007.

[19] C. Reis, J. Dunagan, H. J. Wang, O. Dubrovsky, and S. Esmeir, "Browsershield: Vulnerability-driven filtering of dynamic html," in *IEEE Symposium on Security and Privacy*, Oakland, CA, May 2006.

[20] B. Schneier, "Real story of the rogue rootkit," *Wired*, 2005.

[21] P. Vogt, F. Nentwich, N. Jovanovic, C. Kruegel, E. Kirda, and G. Vigna, "Cross site scripting prevention with dynamic data tainting and static analysis," in *14th Annual Network and Distributed System Security Symposium (NDSS 2007)*, San Diego, CA, February 2007.

CUSP: Customizable and Usable Spam Filters for Detecting Phishing Emails

Madhusudhanan Chandrasekaran, Vidyaraman Sankaranarayanan, Shambhu Upadhyaya
Department of Computer Science and Engineering,
University at Buffalo, State University of New York, Buffalo, NY 14260
{mc79, vs28, shambhu}@cse.buffalo.edu

***Abstract*–Phishing attack continues to be a significant threat to the Internet users and commercial organizations worldwide causing billions of dollars in damage. A successful phishing attack depends on the inability of an end user to accurately tell legitimate and spoofed emails apart. However, unlike their legitimate counterpart, as spoofed emails are composed in bulk, they do not contain any user specific data, which relates users with their accounts. In this paper, as a first step, we propose a customizable spam filter that allows the users to store this user specific data on a per organization basis, and then use the stored data to discriminate against fraudulent emails. As a next step, we propose a NLP based technique to generate context sensitive warnings that would help in educating users about the dangers of phishing attack. Lastly, we test and validate our framework on existing phishing corpus and live emails.**

***Index Terms*–Context-sensitive warnings, Email Fraud, FrameNet, Natural Language Processing (NLP), Phishing, WordNet**

I. INTRODUCTION

Phishing is a Web based attack where attackers trick users into revealing confidential information such as password, credit card number, social security number (SSN), or bank account numbers in fraudulent Websites that mimic the look-and-feel of their legitimate counterpart. This redirection of users into fraudulent Websites can be achieved either through social engineering attacks or forcefully using malware. Even though a variety of threat vectors such as instant message spamming (spimming), public forum spamming and DNS redirection (pharming), can be used for phishing, due to its widespread adoption and ability to be easily spoofed, email continues to be the preferred vehicle to launch such attacks. Recent studies estimate that in 2007 alone, more than 25,000 unique phishing emails hijacking 150 different brands were sent out on a per month basis, resulting over $3 billion dollars in damage worldwide [9].

Given the significant impact on global economy, several anti-phishing efforts have been undertaken in both academia and industry to detect and mitigate phishing attacks. Most of these approaches are implemented as browser add-ons and third-party toolbars that operate on every URL visited by a user to determine their authenticity. Despite their initial success in protecting the users from divulging confidential information across fake Websites, they have several drawbacks. First of all, most of these toolbars adopt decisions based on blacklists (or whitelists), which are propagated to them by some centralized servers. However, as phishing sites are ephemeral, the probability that these lists reach the clients in time is very low. Second, a recent study on 10 popular anti-phishing toolbars conducted by Zhang et al. [15] showed that the toolbars when tested on live phishing data, exhibit poor performance having an overall accuracy of less than 60%. Lastly, as these toolbars operate close to the source of the attack (i.e., on Websites rather than emails), any misclassification error on their part would imply that their users are left defenseless. An alternative to detect phishing attack is to filter out spoofed emails before they reach user's mailbox. Traditionally, anti-spam mechanisms were used for this purpose. Although phishing emails can be regarded as *unsolicited junk*, they do not share the same characteristics as spam emails, thereby requiring specialized filters for classification. In this context, a few specialized efforts have been undertaken that attempt to classify phishing emails based on the features intrinsic to them: such features include, but are limited to, the content type of the message (Plain text/HTML), nature of the contained URLs (dotted IP/encoded format), credibility of the referred domains, words that frequently appear in the phishing email content, etc. However, due to the instant availability of automated tools, it has become possible for the phishers to fabricate phishing emails just by using a reduced subset of these features that can evade even the sophisticated email filters. Moreover, as the features used by these approaches (e.g., frequently occurring words, different visible and referred-to URLs, number of URLs, and email MIME type) are also present in the emails sent by the legitimate institutions, it has become extremely hard to build generic classifiers that can accurately detect phishing emails. To overcome these limitations, we adopt a proactive approach, which attempts to detect phishing emails based on the user specific data contained in them. The main assumption here is that as phishing emails are composed in bulk, they lack in any data that can relate the users to their personal information; on the contrary, legitimate financial institutions send out directed emails to customers using personalized data that are not known publicly (transaction identifiers (tids), abbreviated version of their account number, full name, date-of-birth, address, etc.) This private data, in turn, can act as shared

authentication secret used to validate the sending domain's legitimacy.

In this paper, as our first contribution, we propose CUSP, a **C**ustomizable and **U**sable **S**pam filter to detect **P**hishing attacks, which allows users to store private data on a per organization/account basis. Subsequently, every incoming email that purports to originate from the stored organization is examined to see if it contains the previously stored data. If there is a mismatch (or the data is absent), the email is deemed as suspicious. The notion of verifying the sender's domain for detecting spoofed emails is not new; there exists mechanisms like SPF (Sender Policy Framework), Sender ID, DKIM (Domain Key Identified Mail) that validate the sending domain using IP addresses or digital signatures. Although these mechanisms can vouch for the sending domain's reputation, they still fail to stop users from falling prey to phishing attacks. Furthermore, unlike CUSP, these mechanisms are heavyweight – each of them adopts a different protocol that requires changes to the existing email infrastructure. For the next step, we focus on generating context-sensitive warnings that help users in identifying phishing attacks. As phishing is a social engineering attack, their emails falsely impose an implied sense of urgency and threat (account suspension) or lure and cajole (reward for completing a survey) to trick the users into visiting fake Websites. As our second contribution, we propose a novel technique that relies on context-sensitive text categorization as a means to detect the "tone" or the implied message of the email. We consider this identification a critical factor towards not only identifying the phishing emails, but also communicating the import of the email to the end user. Consider the phishing emails that get past the standard phishing filter: if our framework can provide a meaningful communication to the user regarding the intentions of the email sender, it would not only be an effective methodology to defeat the attack, but could also educate the naïve user against the potential harmful effects, which, after all, is the key to defeating these attacks. We implement CUSP as a plug-in to Microsoft Outlook – a popular email client used by both home and corporate users. Lastly, we evaluate CUSP against existing corpus and report our findings.

The rest of this paper is organized as follows. We begin Section 2 by discussing related work. Section 3 presents a short survey of user specific data contained in financial emails from institutions that are vulnerable to phishing attacks. Section 4 presents the threat model that we are targeting to address in this paper. Section 5 presents an overview of CUSP and its implementation details. The generation of context-sensitive warnings is also discussed here. Section 6 presents the results of our evaluation and discusses the limitations of CUSP. In Section 7, we conclude the paper.

II. RELATED WORK

In this section, in order to bring out the efficacy of our approach, we briefly compare and contrast our work with other related approaches.

A. Browser Plug-ins and Anti-Phishing Toolbars

Since most of the phishing attacks rely on the inability of users to discern legitimate and fake emails apart [17], several commercial and open source toolbars have been proposed to assist the users in determining the validity of the visited Websites. Spoofstick [14] is one such browser add-on which displays the IP address of the spoofed URLs in its toolbar. As it requires the end user to discriminate between the fake and the real Websites, it does not provide an automated solution to detect phishing attacks. NetCraft antiphishing toolbar [12] is another monitoring tool that employs client-server architecture. Each toolbar subscriber acts as a client and is responsible for reporting suspicious Websites to a central server. The server then processes every incoming request by checking the domain age, hosted location and URLs, and then puts the reported Websites either into a whitelist or a blacklist. These lists are propagated to other clients to assist them with their decision making. A disadvantage with such an approach is that as phishing Websites are ephemeral, it might not be possible to propagate the generated blacklists to the clients in time. SpoofGaurd [6] is a browser plug-in that examines the visited Website using stateful and stateless evaluation. The stateless evaluation includes check for invalid links, URL obfuscation attacks, valid https connection and authenticity of SSL/TLS certificates. It also checks to determine whether the images present in the legitimate sites are imported by unknown suspicious domains. Unlike stateless evaluation, stateful page evaluation monitors every outgoing data using site specific salts so that a user does not provide his username and password into a site he has never visited before. In most cases, the final result of these toolbars is either binary (phishing or safe) or ternary process (where a score/color is displayed on the toolbar to warn users about the sites' suspiciousness.) Despite their advantages, a recent study [15] experimented with 10 popular anti-phishing toolbars revealed that the toolbars failed to identify 15% of the phishing Websites used for testing. Also, as discussed earlier, these toolbars depend on the validity of IP as an important detection criterion and fail to protect from attacks launched from the legitimate Websites. Lastly, these toolbars ignore the weak human factor and require users to make the final decision, i.e., to trust a suspicious Website or not.

B. Digital Signing and PKI Based Schemes

Digital signing and trust propagation schemes have been proposed to make email secure and reliable. These schemes employ publicly available standards such as S/MIME, PGP and GPG to encrypt, decrypt and validate email messages. Spam protection framework (SPF), Certified Sender Validation (CSV) and DomainKeys have

also been proposed as an alternative mechanism to authenticate emails based on their sender's domain name. DomainKeys uses digital signatures to authenticate domain name and the entire content of a message, whereas SPF and CSV look at the email headers to identify forgery. Even though these schemes act as an effective anti-spoofing solution, they suffer from several disadvantages. First, adoption of these techniques necessitates steep learning curve which might be elusive to everyday users. Second, these techniques require installation of additional software to support S/MIME, PGP, GPG, etc. These provisions are not readily available in most of the popular Web based email clients such as Yahoo Mail, Hotmail and Gmail. Finally, these techniques suffer from key distribution problems, where a trusted medium is needed to exchange keys needed to sign and encrypt/decrypt messages. In the case of PGP/GPG schemes, as there is no central authority server, a phisher can infiltrate the Web of trust by digitally signing his emails. Another drawback of this PKI-based and authentication based approaches is that both the sender and the receiver need to have the same signing and verification mechanisms.

C. Content based Phishing Attack Detection

Several research efforts employing machine learning and pattern recognition techniques have been proposed to classify phishing emails. Most of the earlier algorithms were tailored to detect spam emails and did not perform well when applied in the context of phishing attacks. These approaches were naïve in the sense that they essentially focus on detecting the presence of uncommon words that appear in the spam emails [5]. As phishing emails closely imitate their legitimate counterpart, unlike spam, they do not contain such random and junk words. In order to classify phishing emails, Fette et al. [8] employ a set of 16 different machine learning algorithms operating on a predefined feature set. The feature set consists of structural elements that indicate presence of illegitimate hyperlinks, IP based URLs, non-matching URLs and other characteristics intrinsic to phishing emails. CANTINA [16] is another tool which uses term frequency and inverse document frequency (tf-idf) to identify commonly appearing words in phishing Web pages. These words along with other structural elements are used as features for classification. As opposed to these heavy-weight approaches, CUSP assists the users to filter out phishing emails based on the user-specific data contained in them. Also, based on the tone of the phishing email, context-sensitive warnings are generated to let the user know the working of phishing attacks.

III. USER SPECIFIC DATA IN THE EMAILS FROM LEGITIMIATE INSTITUTIONS: A BRIEF SURVEY

In order to demonstrate the feasibility of our approach, we present a brief survey on the user specific data contained in the emails from legitimate institutions. This would also allow us to identify the private data that need to be stored in CUSP so that accurate prediction of phishing emails is possible. For this survey, we consider the top 20 most phished brands in 2007, as reported by Phishtank. Phishtank, a collaborative undertaking of academia and industry, operates by assimilating and publicizing phishing email feeds, which are then verified by the interested subscribers. Out of these 20 brands, 17 are online banks and credit card institutions. The remaining three are popular Internet portals that support e-business. The summary of our findings are presented in the form of a table (see Appendix A). All the 20 brands claim that they do not send emails to the customers requesting their personal credentials. Furthermore, the banks' Websites clearly state that any email carrying such information on their behalf is a fraudulent one. Majority of the banks also claim to send out personalized emails to the customers (i.e., having information such as their last/full name, last four digits of their account number, and occasionally their home address.) However, there were mixed response on whether such data can be used for validation purposes. While most of the banks advised the customers to use this data as one of the "visual indicators" to identify spoofed emails, one bank cautioned otherwise citing "spear phishing" as the example. It is important to note that even though it may be possible for an attacker to launch targeted phishing attacks (spear phishing) by using the recipients' private data obtained through other means, they are usually rare due to the difficulty involved. A recent study involving real human subjects shows that the users place implied trust on personalized emails [10]. Although the underlying intention was correct, the subjects were not able to make a clear distinction on whether the personalized data is actually the private data (i.e., not publicly known). For example, the subjects incorrectly trusted the emails that contained first four digits of the credit card number, even though first four digits are not random and are dependent on the card issuer.

IV. THREAT MODEL

In this paper, we restrict our focus to sifting legitimate emails and phishing emails based on user specific data contained in them. Even though it might be possible for an attacker to acquire the user specific data through "dumpster-diving" or illegitimately accessing the user's mailbox, nevertheless such targeted approaches are not scalable from the attacker's standpoint. Phishing attacks can be enforced via different mediums such as "chat", "phone", or by using malware. The defenses against these threats, however, are beyond the scope of our paper. As is, the main shortcoming of our approach is that it cannot be used in cases where the institution under consideration does not include personalized content in their emails. Popular email services like Yahoo, Gmail cannot address users with any personal information. Even in companies like Amazon.com and PayPal, as there is no concept of user account number, the only identifying data available is the user's name. Obtaining the user's name is relatively easy, when compared to other private data such as last four digits of the account number. Hence, in such cases, it may be possible for an attacker to

evade CUSP by using spear phishing attacks with emails comprising of this publicly available user information.

V. OVERVIEW OF CUSP

In this section, we describe how CUSP can be used to detect fraudulent emails from known institutions by giving out its working details. CUSP has been developed as a plug-in for Microsoft Outlook in C# using Visual Studio Tools for Office 2003 (VSTO). CUSP attaches itself to the email client, and is bootstrapped with a list of popular institutions that are prone to phishing attacks. At the time of installation, if a user is subscribed with any of the preloaded institutions contained in CUSP, then he is required to specify the corresponding user specific data that are to be included in legitimate emails from them. Figure 1 shows a form in CUSP requesting such information from the user. In case if an institution of user's choice is not present in CUSP, then he can add a custom tag to include it. Similarly, as there is no common consensus among institutions on what user specific data are to be included in their emails, the user is also provided with an option to add/modify the existing tags representing different fields such as his address, product key, date-of-birth, etc. We also understand that it might be difficult for a naïve user to figure out beforehand the data to be included in CUSP corresponding to a given institution. Ideally, such information needs to be updated by the software provider, as opposed to the user. The user specific data are hashed and stored in CUSP similar to the way in which values for auto-completion fields are stored in a browser. This also ensures that any compromise of CUSP does not easily give away the user information.

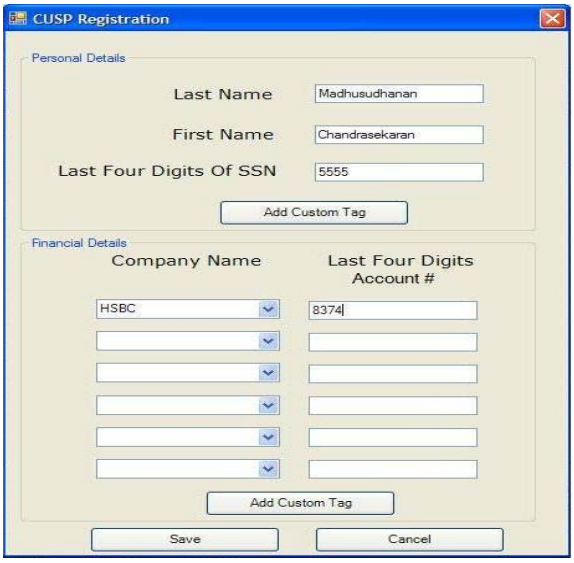


Fig. 1. CUSP requesting the user to enter private data corresponding to the subscribed institution

Any email that purports to originate from the preloaded organizations is examined to see if it contains the relevant user specific data. If it does, then the email is tagged "safe" and sent to the user's mailbox. On the other hand, if the user specific data is missing or is incorrect, the email is tagged as "phishing." If a user fails to enter the required information at the time of installation, a dialog box is prompted asking for relevant information as shown in Figure 2. The user, also, has an option of regarding the email message as not a financial institution. Using this option indiscriminately exposes risk of the user falling prey to phishing attack. If an email is tagged as phishing, then the content of the email is analyzed to extract the "tone" conveyed in the email so that appropriate context-sensitive warning messages can be generated. The process of generating appropriate context-sensitive warnings is discussed in the next section. Once the warnings are generated, they are communicated back to the user in a text box. Figure 3 shows the text box indicating warning messages for a HSBC phishing email which threatens users to disclose sensitive information by using account revocation as an argument. For a user who is still not convinced by the warning, an option of forwarding the email to the provided security department is provided so that accurate response about the validity of the email can be obtained. Disallowing the user to respond to the email until proper clearance is obtained can also be done in a forceful manner as shown in [3].

Fig. 2. A modal box interrupting the user to enter the required user specific data before opening the email

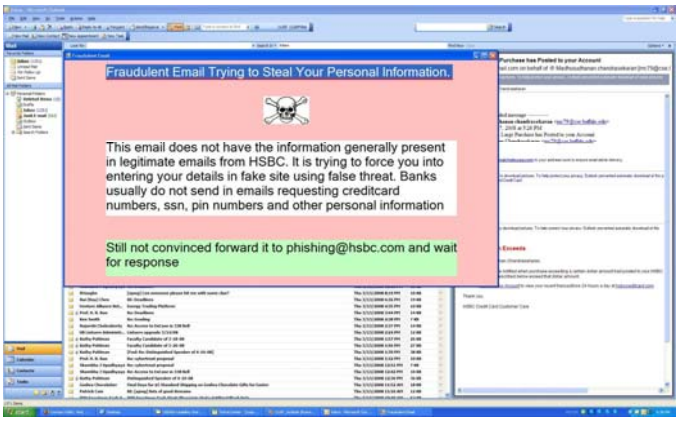


Fig. 3. Context-sensitive warning explaining the tone of the phishing email to the user

A. *Context Sensitive Warning Generation*

The process of generating context-sensitive warnings is two-fold: (i) First, the suspect email's content needs to be parsed and analyzed to extract underlying tone; (ii) then, the extracted tone must be communicated in an effective manner so that the user does not respond to the email. For the first step, during the training stage, each phishing email is broken down into a sequence of words $W = \{w_1, w_2, \ldots, w_n\}$. The goal is then reduced to finding an optimal term set, $W_{spoof} \subset W \times W$, such that the "tone" conveyed in the spoofed email is accurately captured. At a broader level, even though W_{spoof} can be extracted using "bag-of-words" approaches that enumerate frequently appearing words in the email text, they, however, do not take into account the context (presence or absence) of a word with regard to other words in the text. As independent words (1-gram words such as account, credit card, user, name of the institution) that appear frequently in phishing emails also appear in emails sent from legitimate institutions, these 1-gram based approaches fail to scale well. In addition, these approaches do not account for grammatical relevance/context of the words appearing in the phishing email. In order to capture the tone conveyed in phishing emails, during the time of tokenization, we tag each word with its part-of-speech (POS), such as noun, verb, adjective, etc. We use Stanford log-linear part-of-speech tagger for this purpose, which is trained based on Penn Treebank English POS tag set [1]. As a result, each term in W_{spoof} contains a set of related words that appear in the phishing email along with their POS. Subsequently, insignificant words such as articles, conjunctions, prepositions and pronouns, which do not play any role in characterizing phishing emails are eliminated through a stop-word list. Once the insignificant words are removed, remaining words are normalized by converting them into their linguistic roots or "stems". Stemming is a process in which morphological variants of words with similar semantic interpretation are transformed into their equivalent root. For example, in the context of phishing, the words submitting, submitted, submit appearing across different phishing emails are truncated into their root submit. The process of tagging the words with their POS is done prior to stemming to retain the context under which the words appear, as transforming the similar words into their root can alter the underlying POS. Porter's algorithm, a popular stemming algorithm, is used for this purpose [13]. These transformed words are run through a standard thesaurus so that different stems with same meaning can also be normalized. Then the extracted words are passed through WordNet [7], an online resource where the nouns, verbs, adjectives and adverbs are grouped into a set of cognitive synonyms (synsets), which expresses a distinct concept. For example, the synsets for word immediate consists of words/phrases instantly, straightaway, straight off, directly, now, right away, at once, forthwith, like a shot. Each synset is also provided with a short summary (gloss), which provide more descriptive definitions or example sentences. The gloss for the synset of word immediate as provided by WordNet is without hesitation or delay; with no time intervening. Even though the glosses provide description for synset, they do not annotate their underlying semantic roles, which is crucial in identifying the tone conveyed by the email. Using the lexical knowledge base FrameNet [2], synsets are mapped into one or more pre-annotated semantic frames depending on the type of event or state and the participants associated with them. These semantic frames are representations of situations involving various participants, properties, and other conceptual roles. Each frame has a set of associated words (lexical units), and is descriptive of a specific context/situation. Currently FrameNet consists of around 850 semantic frames with 135,000 annotated sentences. Also, each frame can be derived or related to many other frames. For the sake of illustration, the relationship between the *Taking_Time* frame, which denotes time critical contexts and other similar frames in FrameNet is shown in Figure 4.

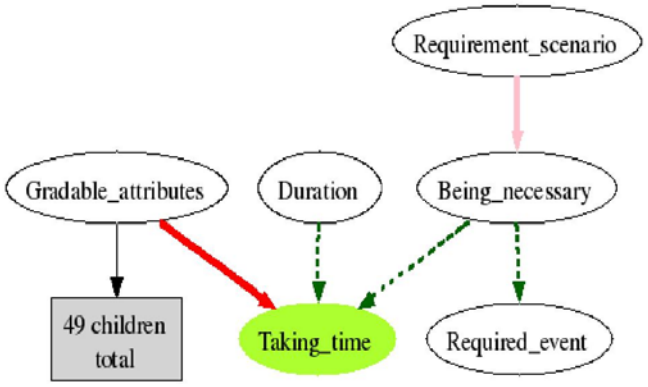


Fig 4. Relationship between the frame Taking_Time and other abstract frames representing semantic roles that specify time boundedness in FrameNet

We adopt the algorithm given in [4] to convert the synsets into equivalent FrameNet frames. The algorithm, essentially, is a two step process: (i) In the first step, all candidate frames, whose lexical unit comprises of words in the synset or their variants (hypernyms/antonyms) are chosen. Then, for words that are not listed in the lexical unit of any frame, the names of the frames are checked to determine if they contain the words in synsets. For a frame to match there has to be at least 50% overlap between the word and the frame name. Finally, in order to select the best set of frames, all candidate frames are weighted depending on selection. The boost factor places more weight on the frames chosen for having the synset words in the lexical unit, as opposed to the frames which contains the words in their frame names. The weights for each frame are computed as shown in Algorithm 1.

input : WordNet Synsets for each word extracted from the email
output: A set of FrameNet frames with corresponding weights indicating the overall relevance

for *each word* w_s *in the synset* **do**
    search_words = set of related hypernyms, antonyms corresponding to w_s from WordNet;
end
evoked_by_lexical_unit = ϕ;
evoked_by_name_match = ϕ;
for *each frame* f *in FrameNet* **do**
    for *each word* w *in search_words* **do**
        if w *is in lexical unit of* f **then**
            evoked_by_lexical_unit(f) = evoked_by_lexical_unit(f) $\cup$ w;
            spreading_factor(w)+ = 1;
        else if *(w has 50% match with f's name)* **then**
            evoked_by_name_match(f) = evoked_by_name_match(f) $\cup$ w;
            spreading_factor(w)+ = 1;
    end
end
for *each word* w *in search_words* **do**
    for *each frame* f *in FrameNet* **do**
        $\text{Weight}(f) = \sum_{w \in \text{evoked\_by\_lexical\_unit}(f)} \frac{\text{similarity}(w_s, w) * \text{boostfactor}}{\text{spreading\_factor}(w)}$;
        $\text{Weight}(f) \mathrel{+}= \sum_{w \in \text{evoked\_by\_name\_match}(f)} \frac{\text{similarity}(w_s, w)}{\text{spreading\_factor}(w)}$;
    end
end

Algorithm 1. Mining set of weighted FrameNet frames for WordNet synsets

In order to extract the tone of the email, a conglomerate set of frames is created by aggregating all the frames returned for each word in the phishing email message. In the training phase, we group these conglomerate sets into five categories, namely, justification, Penalty, Urgent Action, Reward and Concern, depending on the corresponding returned constituent frames and their weights. Also, appropriate context-sensitive warnings are assigned to the conglomerate sets, which describe the intent of the phisher to the recipients in an effective manner. In the testing phase, each email tagged as "suspicious" by CUSP is analyzed to see if it matches exactly/partially with one or more tagged conglomerate frames. Then depending on the match, the generated context-sensitive warnings are communicated to the user

VI. EVALUATION

A. Dataset

For our experiments, we consider a publicly available phishing corpus [11], which contains 434 phishing messages collected in a period of five months. Preprocessing is done to eliminate ill-formed emails that were not composed in English. Also, for the sake of brevity, messages with significant amount of spam (junk words) were discarded. The final list thus formed contained a total of 362 phishing emails. Almost all of the emails did not contain any (even random) user-specific data, barring a few exceptions. These set of emails were detected promptly by CUSP without any misclassification errors. A small fraction of emails (<2%) that impersonated eBay correctly had the user's full name along with the user id. As these data can be obtained easily, it may be possible for an attacker to evade CUSP by using more focused attacks. Moreover, a few emails had fake transaction ids to fool the users into believing that they are sent by legitimate institution's security department.

B. Experiences with Context-Sensitive Warning Generation

The process of generating context-sensitive warning messages is two-fold: (i) In the first phase, we take a set of 200 email messages as training data and run it through the CUSP engine. The set of WordNet synsets returned are then passed into FrameNet to obtain the set of relevant frames. Each frame bears a part of the semantic structure of the tone conveyed in the email. For example, emails that impose a sense of urgency in the users (i.e., belonging to the Urgent Action category), have frames with names such as Response, Communication response, Requesting, Activity pause, Submitting documents compliance, etc. Similarly, emails that express security concern as an argument (i.e., belonging to Concern category) to deceive the users have frames with names such as Assistance, Personal relationship, Cause to start, Becoming aware, Evidence, Request, Education teaching, etc. Phishing emails that extort private data from users by threatening account revocation as a reason (i.e., frames falling into Penalty category), have frame names such as Inhibit movement, Thwarting, Compliance, Scrutiny, Attempt, Persuasion, Telling, etc. Similarly, the names of the frames corresponding to emails that give away incentives to users (i.e., frames belonging to the Reward category) for disclosing their account details are Telling, Personal relationship, Compatibility etc. Lastly, frames corresponding to Justification category include Waking up topic, Questioning, Leadership Request, Execute plan using, Protecting, etc. Then, we tag each conglomerate set (email) to only one of the five categories, even though they may have different overlapping individual frames. There were a total of 126 frames returned by FrameNet, which were formed by combining one or more of the 850 pre-annotated frames. Also, roughly on an average each email message in the training data returned a total of 15 different frames. The time taken to process each email message in the training phase is roughly in an order of few seconds; (ii) Once the training phase is completed, in the testing phase, each message is processed and depending upon the category they fall in appropriate context-sensitive message is conveyed to the user.

C. Limitations of CUSP

There are three main limitations with CUSP: (i) First, as of now, the list of institutions that are vulnerable to phishing attacks is directly hard-coded in CUSP. Even though users are provided with an option to add their own custom tags, to make it more scalable, it is essential that these tags are managed remotely by a centralized system; (ii) Second, as CUSP operates only on text messages, it is still possible for a phisher to evade detection by encoding spoofed emails as images. To address such cases, we can provide warnings to users instructing them not to give away confidential information in response to such emails; (iii) Third, at this stage we only target phishing emails that are composed in English. However, as FrameNet like systems exist for other languages, porting CUSP to these languages is relatively easy.

VII. CONCLUSION AND FUTURE WORK

In this paper, we introduce CUSP, a customizable filter to separate phishing and legitimate emails based on user specific data contained in them. At the time of deployment, CUSP is bootstrapped with a list of known institutions that are vulnerable to phishing attacks. A user has the option of storing his personal information in the filter corresponding to subscribed institutions. Subsequently, every incoming email that purports to originate from the bootstrapped institutions is analyzed to see if it contains the corresponding user specific data. In case, if the data is absent, the email is tagged as phishing, and depending on the "tone" of the email, context-sensitive warnings are generated. We believe that these context-sensitive-warnings would help educate users about the potential hazards of phishing attacks. As a part of our future work, we would like to evaluate CUSP on emails sent over from legitimate organization and conduct a user study to test out the efficacy of context-sensitive warnings. Such a field test would help in fine tuning the working of CUSP so that better results could be achieved.

REFERENCES

[1] *Stanford log-linear part-of-speech tagger*. http://nlp.stanford.edu/software/tagger.shtml, March 2008.

[2] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet project," *In COLING/ACL-98*, pages 86–90, 1998.

[3] J. C. Brustoloni and R. Villamari´n-Salomo´n, "*Improving security decisions with polymorphic and audited dialogs*," in SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security, pages 76–85, New York, NY, USA, 2007. ACM.

[4] Burchardt, K. Erk, and A. Frank, "A WordNet detour to FrameNet," in *Sprachtechnologie, mobile Kommunikation und linguistische Resourcen,* B. Fisseni, H.-C. Schmitz, B. Schrder, and P. Wagner, Eds, page 16, Frankfurt am Main, 2005. Lang, Peter.

[5] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing email detection based on structural properties," in *New York State Cyber Security Conference (NYS)*, Albany, NY, 2006.

[6] N. Chou, R. Ledesma, Y. Teraguchi, and J. Mitchell, "Client-side defense against web-based identity theft," in *11th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, 2004.

[7] Cognitive Science Laboratory, Princeton University. *WordNet - a lexical database for the english language*. http://WordNet.princeton.edu/, 2008.

[8] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *16th international conference on World Wide Web (WWW)*, pages 649–656, Banff, Alberta, Canada, 2007. ACM Press.

[9] Gartner Press Releases, "*Gartner survey shows phishing attacks escalated in 2007; more than $3 billion lost to these attacks*." http://www.gartner.com/it/page.jsp?id=565125.

[10] M. Jakobsson, "The human factor in phishing," in *Privacy & Security of Consumer Information*, 2007.

[11] J. Nazario, *phishingcorpus homepage*, march 2008. http://monkey.org/~jose/wiki/doku. php?id=PhishingCorpus.

[12] NetCraft, *Netcraft anti-phishing toolbar*, 2004.

[13] M. F. Porter, "An algorithm for suffix stripping pages," in *Readings in information retrieval*, pages 313– 316, 1997.

[14] Spoofstick, *Spoofstick toolbar*, 2004.

[15] Y. Zhang, S. Egelman, L. Cranor, and J. Hong, "Phinding phish: An evaluation of anti-phishing toolbars," in *Proceedings of Network & Distributed System Security Symposium (NDSS)*, 2007.

[16] Y. Zhang, J. Hong, and L. Cranor. Cantina: a content-based approach to detecting phishing web sites. In 16th international conference on World Wide Web (WWW), pages 639– 648, Banff, Alberta, Canada, 2007.

[17] R. Dhamija, J.D. Tygar, and M. Hearst, "Cantina: a content-based approach to detecting phishing web site," in *16th international conference on World Wide Web (WWW)*, pages 639– 648, Banff, Alberta, Canada, 2007.

Appendix: A

TABLE I
SURVEY OF TOP 20 PHISHED BRANDS' SECURITY POLICY.

Name of the Bank	Token Identifier	Source	Website Link Specifying Privacy Policy
Amazon.com	Nil	S	http://www.amazon.com/gp/help/customer/display.html?nodeId=15835501
Bank of America Corporation	U	W	https://www.bankofamerica.com/privacy/Control.do?body=privacysecur_email_fraud
Barclays Bank	FN, ADR	W	http://www.personal.barclays.co.uk/BRC1/jsp/brccontrol?task=homefreevi2\&value=9117\&target=_blank\&site=pfs
Branch Banking and Trust Comp	FN, LF	W	http://www.bbt.com/bbt/about/privacyandsecurity/emailcommunication.html
Capital One	FN, LF	S	http://capitalone.com/fraud/prevention/phishing.php?linkid=WWW_Z_Z_Z_FRD_C1_01_T_FPRV1
Citibank	FN, LF	S, W	https://www.citicards.com/cards/wv/detail.do?screenID=607
eBay	FN, UID	S, W	http://pages.ebay.com/education/spooftutorial/
Fifth Third Bank	FN	W	https://www.53.com/wps/portal/privacy/?New_WCM_Context=/wps/wcm/connect/FifthThirdSite/Global+Utilities/Privacy\%20\%26\%20Security/#
HSBC Bank	FN, LF	S, W	http://www.us.hsbc.com/1/2/3/personal/inside/securitysite/your-responsibility
HSBC Credit Card	FN, LF	S, W	http://www.us.hsbc.com/1/2/3/personal/inside/securitysite/your-responsibility
JP Morgan Chase and Co	FN, LF	S, W	http://www.chase.com/ccp/index.jsp?pg_name= ccpmapp/shared/assets/page/Report_Fraud\#5
National City	U	W	http://www.nationalcity.com/about/privacy/identity/default.asp
PayPal	FN	W	http://www.paypal.com/cgi-bin/webscr?cmd=p/gen/fraud-prevention-outside
Poste Italine	U	W	http://www.poste.it/online/phishing.shtm
Regions Bank	U	R	http://www.regions.com/about_regions/email_fraud.rf
US Bank	FN, LF	W	https://www4.usbank.com/internetBanking/en_us/info/BrowserRequirementsOut.jsp
Volksbanken Raiffeisenbankeni	U	W	http://www.vr-networld.de/c132/default.html
Wachovia	FN, LF	R	http://www.wachovia.com/securityplus/page/0,,10957_10970,00.html
Wells Fargo	U	W	https://www.wellsfargo.com/privacy_security/fraud/report/fraud?_requestid=394409
Western Union	U	W	http://www.westernunion.com/info/fraudProtectYourself.asp

- **All the companies indicate that they do not send emails requesting confidential information.**
- Token identifiers indicate what user specific data is included in the companies' email to the customers
  - FN - Full Name, UID - Username, LF - Last four digits of Account Number, NA - No private data, U - Unverified/Not known
- Source indicates where the information about company's security policy was obtained (W - Website, S - Sample email.)

Secure Space Computing with Exertions

Daniel Kerr and Michael Sobolewski
Texas Tech University, SORCER Research Group
sobol@cs.ttu.edu

***Abstract*–Exertion-oriented space computing is a valuable advance in distributed and parallel computing seeing as it abstracts out several major problems in distributed computing, such as load balancing and mutual exclusion. The main problem with space computing is that of security due to the fact that exertion spaces are inherently public and ad hoc, thus making it difficult to implement secure groups. The location independent group key interactive management framework presents a federated methodology and protocol for group management that is secure, scalable, and modifiable for the metacomputing exertion-oriented space computing environment. The framework does so through the use of a group establishment protocol, authorization and authentication services, high level cryptography, and persistent group information storage. The SORCER computing grid is used as a validation case for the framework and is presented in this paper.**

I. INTRODUCTION

Security in networking has been a serious issue since the dawn of networked systems. With the increasing trust in computers for the use of storing sensitive data (account numbers, social security numbers, criminal records, health history, etc) security has drastically increased in order to coordinate with the importance of the information being exchanged. Along with the growth in computer networking is the growth in distributed computing.

A rather new concept in distributed computing is the concept of space computing. Space computing helps to solve several of the significant problems with distributed computing, but still leaves room for improvement. The space computing environment starts by being completely public so that all services can access objects in the publicly shared space. The space broker is also oblivious to who is a member of the space environment.

Space computing, being based on the idea of the tuple space from Linda programming language, utilizes three major functions to manipulate tuples (or objects) from the tuple space (or object space). In Linda they are `in` – the removal of a tuple from the tuple space for reading, `out` – the writing of a tuple to the tuple space, and `rd` – the copying of a tuple from the tuple space for reading [[11]]. These three functions are mimicked in other implementations such as JavaSpaces [[3]] where the `in` function is called `take`, the `out` function called `write`, and the `rd` function called `read`.

The question at hand is whether or not a secure encrypted exertion oriented programming model that implements group creation and maintenance services can be implemented in a space computing environment.

The paper is organized as follows. Section II provides a brief description of the SORCER environment; Section III and IV describes basics of exertion-oriented programming; Section V describes three types of collaborations including space-based collaborations; Section VI presents the required cryptography and key agreement; Section VII describes a framework for the creation and management of groups within the exertion-oriented space computing environment, and Section 8 provides concluding remarks.

II. SORCER

SORCER (Service Oriented Computing EnviRonment) [[9]] is a federated service-to-service (S2S) metacomputing environment that treats service providers as network objects with well-defined semantics of a federated service object-oriented architecture. It is based on Jini semantics of services in the network and Jini programming model with explicit leases, distributed events, transactions, and discovery/join protocols [[6]]. While Jini focuses on service management in a networked environment, SORCER focuses on exertion-oriented programming and the execution environment for exertions. SORCER uses Jini discovery/join protocols to implement its *exertion-oriented architecture* (EOA) using *federated method invocation* [[8]], but hides all the low-level programming details of the Jini programming model.

In EOA, a service provider is an object that accepts remote messages from service requestors to execute collaboration. These messages are called service exertions and describe *service data*, *operations* and provider's *control strategy*. An *exertion task* (or simply a *task*) is an elementary service request, a kind of elementary remote instruction executed by a single service provider or a small-scale federation for the same service data. A composite exertion called an *exertion job* (or simply a *job*) is defined hierarchically in terms of tasks and other jobs, a kind of network procedure executed by a large-scale federation. The executing exertion is dynamically bound to all required and currently available service providers on the network. This collection of providers identified in runtime is called an *exertion federation*. The federation provides the implementation for the collaboration as specified by its exertion. When the federation is formed, each exertion's operation has its corresponding method (code) available on the network. Thus, the network *exerts* the collaboration with

the help of the dynamically formed service federation. In other words, we send the request onto the network implicitly, not to a particular service provider explicitly.

The overlay network of service providers is called the *service grid* and an exertion federation is in fact a *virtual metacomputer*. The metainstruction set of the metacomputer consists of all operations offered by all service providers in the grid. Thus, an exertion-oriented (EO) program is composed of *metainstructions* with its own *control strategy* and a *service context* representing the metaprogram data. The service context describes the data that tasks and jobs work on. Each service provider offers services to other service peers on the object-oriented overlay network. These services are exposed *indirectly* by operations in well-known public remote interfaces and considered to be elementary (tasks) or compound (jobs) activities in EOA. Indirectly means here, that you cannot invoke any operation defined in provider's interface directly. These operations can be specified in a requestor's exertion only, and the exertion can be passed on to any service provider via the top-level `Servicer` interface implemented by all service providers called *servicers*—service peers. Thus all service providers in EOA implement the `service(Exertion, Transaction):Exertion` operation of the `Servicer` interface. When the servicer accepts its received exertion, then the exertion's operations can be invoked by the servicer itself, if the requestor is authorized to do so. Servicers do not have mutual associations prior to the execution of an exertion; they come together dynamically (federate) for a collaboration as defined by its exertion. In EOA requestors do not have to lookup for any network provider at all, they can submit an exertion, onto the network by calling `Exertion.exert(Transaction) :Exertion` on the exertion. The `exert` operation will create a required federation that will run the collaboration as specified in the EO program and return the resulting exertion back to the exerting requestor. Since an exertion encapsulates everything needed (data, operations, and control strategy) for the collaboration, all results of the execution can be found in the returned exertion's service contexts.

Domain specific servicers within the federation, or task peers (*taskers*), execute task exertions. *Rendezvous* peers (jobbers and spacers) coordinate execution of job exertions. Providers of the `Tasker`, `Jobber`, and `Spacer` type are three of SORCER main infrastructure servicers, see

Fig. 1.

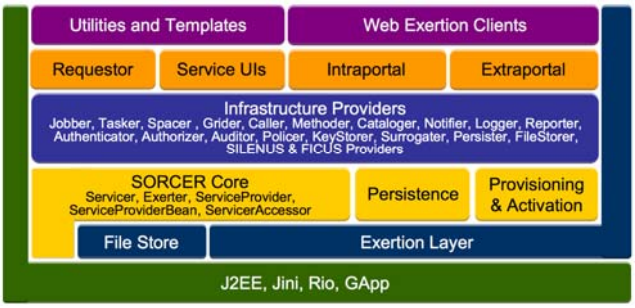


Fig. 1. The SORCER layered functional architecture.

In view of the P2P architecture defined by the `Servicer` interface, a job can be sent to any servicer. A peer that is not a `Jobber` type is responsible for forwarding the job to one of available *rendezvous* peers in the SORCER environment and returning results to the requestor.

Thus implicitly, any peer can handle any job or task. Once the exertion execution is complete, the federation dissolves and the providers disperse to seek other collaborations to join. Also, SORCER supports a traditional approach to grid computing similar to those found, for example in Condor [[10]]. Here, instead of exertions being executed by services providing business logic for invoked exertions, the business logic comes from the service requestor's executable codes that seek compute resources on the network.

Grid-based services in the SORCER environment include `Grider` services collaborating with `Jobber` and `Spacer` services for traditional grid job submission. `Caller` and `Methoder` services are used for task execution. `Callers` execute conventional programs via a system call as described in the service context of submitted task. `Methoders` can download required Java code (task method) from requestors to process any submitted context accordingly with the code downloaded. In either case, the business logic comes from requestors; it is a conventional executable code invoked by `Callers` with the standard `Caller`'s service context, or mobile Java code executed by `Methoders` with a matching service context provided by the requestor.

III. EXERTION-ORIENTED PROGRAMMING

Each programming language provides a specific computing abstraction. Procedural languages are abstractions of assembly languages. Object-oriented languages abstract entities in the problem domain that refer to "objects", communicating via message passing, as their representation in the corresponding solution domain. However, we cannot just take an object-oriented program developed without distribution in mind, and make it a distributed system, ignoring the unpredictable network behavior. The EO programming is a form of object-oriented

distributed programming that allows us to describe the distributed problem in terms of the intrinsic unpredictable network domain instead of in terms of distributed objects hiding the notion of the network domain that in reality cannot be hidden.

What intrinsic distributed abstractions are defined in SORCER? Well, service providers are "objects", but they are specific objects—they are network objects with a network state, network behavior, and network type(s). Service providers act also as network peers (servicers)*;* they are replicated and dynamically provisioned for reliability to compensate for network failures. Servicers can be found transparently in runtime by type(s) they implement. They can federate for an exertion submitted onto the network and participate in the collaboration outlined by the exertion. The exertion encapsulates service *data*, *operations*, and *control strategy* used by the collaboration. The component exertions may need to share context data of ancestor exertions, and the top-level exertion is complete only if all nested exertions are successful. Thus, a collaboration is a *process*, an exertion is the *specification* of collaboration, and a dynamic federation of peers is the *implementation* of a collaboration.

Let's first look at the EO approach to see how it works. Exertion-oriented programs consist of *exertion* objects called tasks and jobs. An exertion *task* corresponds to an individual network request to be executed on a service provider. An exertion *job* consists of a structured collection of tasks and other jobs. The data upon which to execute a task or job is called a s*ervice context*. Tasks are analogous to executing a single program or command on a computer, and the service context would be the input and output streams that the program or command uses. A job is analogous to a batch script that can contain various commands and calls to other scripts. Pipelining Unix commands allows us to perform complex activities without writing complex programs. As an example, consider a script `sort.sh` connecting simple processes in a pipeline as follows:

```
cat hello.txt | sort | uniq > bye.txt
```

The script is similar to an exertion job in that it consists of individual tasks that are organized in a particular fashion. Also, other scripts can call the script `sort.sh`. An exertion job can consist of tasks and other jobs, much like a script can contain calls to commands and other scripts.

Each of the individual commands, such as `cat`, `sort`, and `uniq`, would be analogous to a task. Each task works with a particular service context. The input context for the `cat` "task" would be the file `hello.txt`, and the "task" would return an output context consisting of the contents of `hello.txt`. This output context can then be used as the input context for another task, namely the `sort` command. Again the output context for `sort` could be used as the input context for the `uniq` task, which would in turn give an output service context in the form of `bye.txt`.

To further clarify what an exertion is, an exertion consists mainly of three parts: a set of *service signatures*, which is a description of operations in collaboration, the associated *service context* upon which to execute the exertion, and control strategy (default provided) that defines how signatures are applied in the collaboration. A *service signature* specifies at least the provider's interface that the service requestor would like to use and a selected operation to run within that interface. There are four types of signatures that can be used for an exertion: `PREPROCESS`, `PROCESS`, `POSTPROCESS`, and `APPEND`. An exertion must have one and only one `PROCESS` signature that specifies what the exertion should do and who works on it. An exertion can optionally have multiple `PREPROCESS`, `POSTPROCESS`, and `APPEND` signatures that are primarily used for formatting the data within the associated service context. A *service context* consists of several data nodes used for input, output, or both. A task may work with only a single service context, while a job may work with multiple service contexts since it can contain multiple tasks. The programmer can define a control strategy as needed for the underlying exertion by choosing relevant exertion types and configuring attributes of service signatures [[7]]. A reader interested in EO programming detail can review two simple EO programs for the `sort.sh` in [[7]].

If we use the Tenex C shell (`tcsh`), invoking the UNIX script is equivalent to: `"tcsh sort.sh"`, i.e., passing the script `sort.sh` on to `tcsh`. Similarly, to invoke the exertion `sortJob`, we call `"sortJob.exert()"`. Thus, the exertion is the program and the network shell at the same time, which might first come as a surprise, but close evaluation of this fact shows it to be consistent with the meaning of object-oriented distributed programming. Here, the virtual metacomputer is an ad hoc federation that does not exist when the exertion is created. Thus, the notion of the virtual metacomputer is encapsulated in the exertion (specification) that creates the required federation on-the-fly (implementation) to execute the collaboration (process).

IV. SERVICE MESSAGING AND EXERTIONS

In object-oriented terminology, a message is the single means of passing control to an object. If the object responds to the message, it has an operation and its implementation (method) for that message. Because object data is encapsulated and not directly accessible, a message is the only way to send data from one object to another. Each message specifies the name (identifier) of the receiving object, the name of operation to be invoked, and its parameters. In the unreliable network of objects; the receiving object might not be present or can go away at any time. Thus, we should postpone receiving object

identification as late as possible. Grouping related messages per one request for the same data set makes a lot of sense due to network invocation latency and common errors in handling. These observations lead us to service-oriented messages called exertions. An exertion encapsulates multiple *service signatures* that define operations, a *service context* that defines data, and a *control strategy* that defines how signature operations flow in collaboration. Different types of control exertions (`IfExertion, ForExertion, and WhileExertion`) [[7]] can be used to define flow of control that can also be configured additionally with adequate signature attributes (*flow type* and *access type*—see Section V).

An exertion can be invoked by calling exertion's `exert` operation: `Exertion.exert(Transaction) :Exertion`, where a parameter of the `Transaction` type is required when the transactional semantics is needed for all participating nested exertions within the parent one, otherwise can be `null`. Thus, EO programming allows us to submit an exertion onto the network and to perform executions of exertion's signatures on various service providers indirectly, but where does the service-to-service communication come into play? How do these services communicate with one another if they are all different? Top-level communication between services, or the sending of service requests (exertions), is done through the use of the generic `Servicer` interface and the operation `service` that all SORCER services are required to provide—`Servicer.service(Exertion, Transaction)`. This top-level service operation takes an exertion as an argument and gives back an exertion as the return value. How this operation is used in the federated method invocation framework is described in detail in [8].

So why are exertions used rather than directly calling on a provider's method and passing service contexts? There are two basic answers to this. First, passing exertions helps to aid with the network-centric messaging. A service requestor can send an exertion out onto the network—`Exertion.exert()`—and any servicer can pick it up. The servicer can then look at the interface and `PROCESS` operation requested within the exertion, and if it doesn't implement the desired interface or provide the desired operation, it can continue forwarding it to another provider who can service it. Second, passing exertions helps with fault detection and recovery. Each exertion has its own completion state associated with it to specify if it has yet to run, has already completed, or has failed. Since full exertions are both passed and returned, the requestor can view the failed exertion composition to see what method was being called as well as what was used in the service context input nodes that may have caused the problem. Since exertions provide all the information needed to execute a task including its control strategy, a requestor would be able to pause a job between tasks, analyze it and make needed updates. To figure out where to resume a job, a rendezvous service would simply have to look at the task's completion states and resume the first one that wasn't completed yet.

V. PUSH AND PULL COLLABORATIONS

SORCER also extends exertion execution abilities through the use of a rendezvous service implementing the `Spacer` interface. The `Spacer` service can drop exertions into a shared object space, implemented using JavaSpaces [2], in which collaborating servicers can retrieve matching exertions, execute them, and return the resulting exertions back to the object space. When the attribute *access type* of a `PROCESS` signature is set to `PULL` then the associated exertion is passed onto a `Spacer`, otherwise (access type is `PUSH`) the exertion is passed directly on to the servicer specified by the signature. Another signature attribute—*flow type*, manages the flow of control (`SEQUENTIAL`, `PARALLEL`, or `CONCURRENT`) for all component exertions at the same level.

In Fig. 2, four use cases are presented to illustrate push vs. pull exertion processing with either `PUSH` or `PULL` access types. We assume here that an exertion is a job with two component exertions executed in parallel (sequence numbers with a and b), i.e., the job's signature flow type is `PARALLEL`. The job can be submitted directly to either `Jobber` (use cases: 1—access is `PUSH`, and 2—access is `PULL`) or `Spacer` (use cases: 3 —access is `PUSH`, and 4—access is `PULL`) depending on the interface defined in its `PROCES` signature. Thus, in cases 1 and 2 the signature's interface is `Jobber` and in cases 3 and 4 the signature's interface is `Spacer` as shown in Fig. 2. The exertion's `ServicerAccessor` delivers the right service proxy dynamically, either for a `Jobber` or `Spacer`. If the access type of the parent exertion is `PUSH`, then all the component exertions are directly passed on to servicers matching their `PROCESS` signatures (case 1 and 3), otherwise they are written into the exertion space by a `Spacer` (case 2 and 4). In the both cases 2 and 4, the component exertions are pulled from the exertion space by servicers matching their signatures as soon as they are available. Thus, `Spacers` provide efficient load balancing for processing the exertion space. The fastest available servicer gets an exertion from the space before other overloaded or slower servicers can do so. When an exertion consists of component jobs with different access and flow types, then we have a *hybrid* case when the collaboration potentially executes concurrently with multiple *pull* and *push* sub collaborations at the same time.

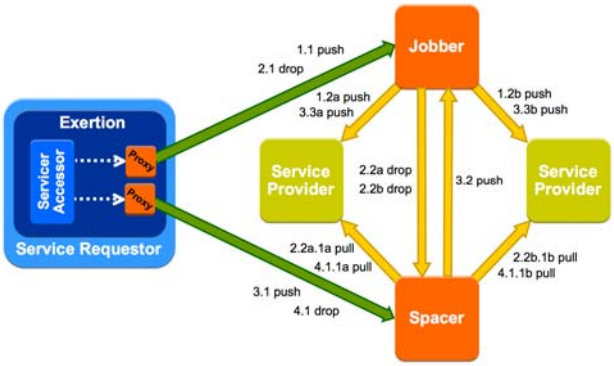


Fig. 2. Push vs. pull exertion processing

VI. CRYPTOGRAPHY AND KEY AGREEMENT

The major topic in cryptography necessary to understand the location independent group key interactive management (LOKI) framework is that of key exchange protocols. Key exchange protocol (also known as key agreement) is an agreement for how parties agree upon on a key. They require that each member must impact the key, that no previously exchanged information can be assumed, that it must prevent eavesdropping, and external parties should not be able calculate the key based on any publicly shared information.

When talking about SORCER `PULL` collaborations it is obvious that a flexible key exchange protocol is necessary in order to accommodate the creation of groups of unknown size. It is this requirement that prompted the creation of the Multi Diffie Hellman Key Exchange Protocol, which scales the standard two party Diffie Hellman Key Exchange Protocol into an N-ary key exchange protocol. It does so by linking the standard Java `KeyAgreement` object of each party member with all other party member's `KeyAgreement` in order to create a *Complementary Compound Key* (CCK) for each party member, which acts as a new public key for each specific member [[4]]. When a member's specific CCK is combined (`doPhase`) with its local Java standard `PrivateKey,` it will generate the group wide *Shared Secret Key* (SSK), which can then be used for both encryption and decryption, of shared objects. An example of this is shown in Fig. 3, and demonstrates a five party group.

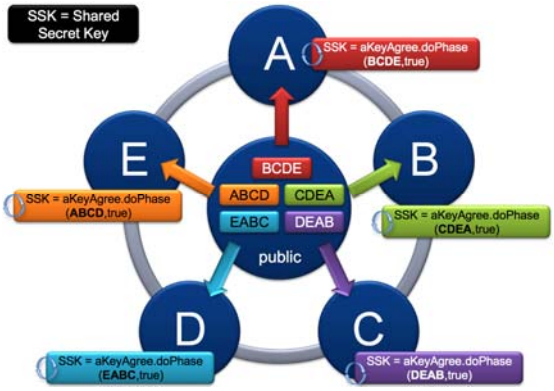


Fig. 3. Multi Diffie-Hellman Key Exchange Protocol

As you can see in Fig. 3, each party member's CCK consists of a chain of all other party members in the group. The main restriction on the Multi Diffie Hellman process is that it is more time consuming for groups of large size (greater than 1000). This is shown in Fig. 4.

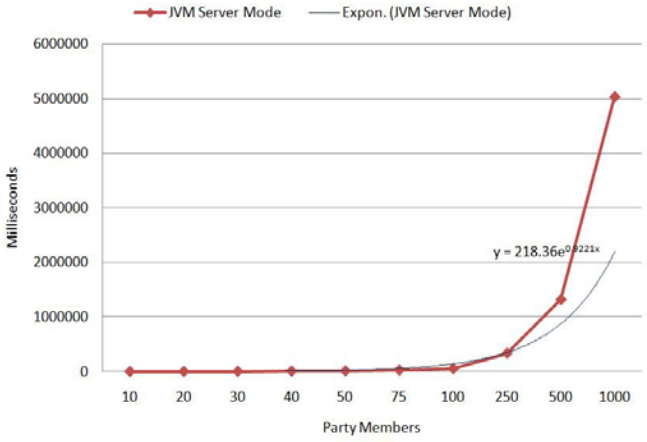


Fig. 4. Multi Diffie-Hellman Running Time Curve

Fig. 4, shows both the test data, in red, and the estimated line of best fit, in grey. The line of best fit shows that the Multi Diffie Hellman process is exponentially complex. Even though this is not ideal, this complexity is acceptable for the for the test case of LOKI implemented in the SORCER compute Grid (cGrid) application [[9]], due to the fact that groups are typically smaller than twenty five party members.

VII. LOKI FRAMEWORK

The LOKI framework establishes a protocol for group creation and management within `PULL` collaborations (see Section 5). There are several assumptions that are made about groups within the framework, the first being that there is a requestor, or a service desiring a group to be created. It is this requestor who will, by default be the administrator to the group that results from the creation request. The second

assumption is that the requestor knows what type of services or who it would like to invite to the group. From these two assumptions the LOKI framework handles the management of all subsequent services previously described in the exertion-oriented model.

To start we will look at what characteristics and attributes each service in LOKI has within the framework (see Fig. 5). Upon creation, each service will create a Java `KeyPair`, `KeyAgreement`, encryption `Cipher`, decryption `Cipher`, unique identifier, and status bits. This member information is utilized in order to perform LOKI secured read, write, and take operations to the exertion space. When a member performs a LOKI secured read, write, or take operation, it will first utilize discovery lookup to locate a `LokiGroupManagement` provider. The LokiGroupManagement provider is a service running within the space environment that is responsible for storing all group activity in a persisten data storage.

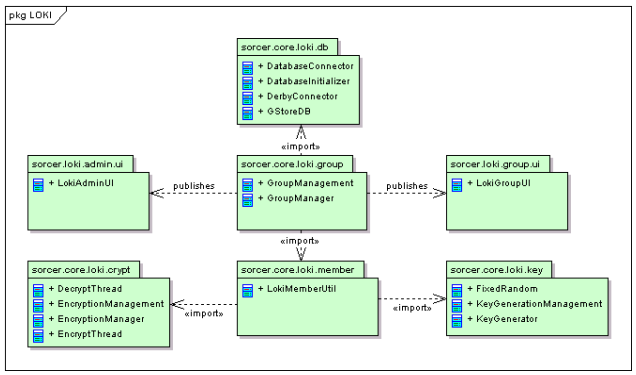


Fig. 5. LOKI package model

The use of persistent data storage allows for the framework to do continuity checking, ensuring that all data and group information is valid throughout group activity. If at anytime the group information is conflicting or faulty, the group is dissolved; all exertions are taken from the space and stored in the persistent data storage only to be accessed by the group administrator.

To understand the protocol of group creation in the simplest form it is best to start looking at the cGrid application without group implementation, then compare it to the implementation of LOKI groups in the LOKI cGrid (lcGrid) application.

A. SORCER Grid Collaborations

The `Grider` provider publishes a user interface which is used to enter the specifications for the job to execute. This information is stored into individual tasks, one for every block of data to be executed. These tasks are stored in a job and passed to the `Spacer` provider. The `Spacer` then breaks them down and dispatches them to the exertion space ("in" exertions in Fig. 6).

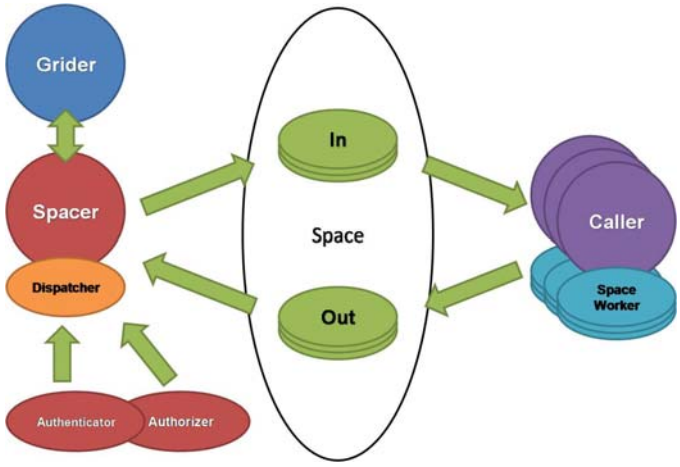


Fig. 6. cGrid Activity

The `Caller` service, upon startup, launches several `SpaceWorker`s that constantly listen to the exertion space for exertions with the `Caller` service type. At this point the `SpaceWorker`s start the collaboration between the `Spacer` and the `Caller`. The `SpaceWorker`s find the dropped exertion from the `Spacer` provider, take it from the space and call their own `exert` method for the retrieved matching exertion. Once execution is complete, the `Caller` service drops the results back to the space ("out" exertions in Fig. 6). The `Spacer`'s results collection thread, waits for these results and takes them from the exertion space. It is at this point that the `Spacer-Caller` collaboration is complete and the resulting information is processed. The results are then passed back to the `Grider` via the previously established proxy, where they are either presented to the user or persisted in the SORCER federated file system [[1]].

B. LOKI Enabled Collaborations

Execution of the cGrid application with LOKI starts in much the same way as it does without LOKI. The `Grider` provider publishes a user agent, which is used to enter the specifications for the job to execute. This information is stored into individual tasks, one for every block of data to be executed. These tasks are stored in a job and passed to the `Spacer` provider via `ServicerAccesor`. It is here where the first of two collaborations start between the `Spacer` and the `Caller (see` Fig. 7`)`. The `Spacer` takes the `PULL` collaboration attributes and sends out invitations or *Creator KeyPair* (CKP) exertions to the services required to execute the `PULL` collaboration, via the exertion space. Each invitation contains the `Spacer`'s public key, so that responses can be encrypted specifically for the `Spacer` to decrypt. The `Caller`'s `SpaceWorker` picks up this object, sees that it is an invitation, extracts the `Spacer`'s public key, encrypts its key pair with the `Spacer`'s public key, adds its own public key object, and drops this response

back to the exertion space (KP). The `Spacer` waits to receive a response for each invitation, then utilizes the `Authorizer` service [[2]] to validate that the responses are legitimate. It is here that the first collaboration is complete.

The `Spacer` service initiates the second collaboration in the LOKI protocol by combining the decrypted key pairs in order to calculate a CCK for each member of the newly created group. These CCKs are packaged into a CCK exertion, which is then dropped to the space. After this is complete, the `Spacer` begins to encrypt the initial job with the shared secret key, calculated by the `Spacer`'s private key and respective CCK, and drops the encrypted job exertion to the exertion space. The `Caller`'s `SpaceWorker` retrieves its respective CCK from the retrieved CCK exertion. The `Caller`'s `SpaceWorker` then retrieves the encrypted exertion from the exertion space and decrypts it with the generated shared key. The `SpaceWorker` passes it up to the `Caller`, where it's `exert` method is called, and results are generated. These results are then encrypted with the shared secret key, and dropped to the exertion space. The `Spacer` picks up the component results from the exertion space, decrypts them with the group shared key, and then encrypts the resulting top level exertion with the `Grider`'s public key. Once this is complete they are passed to the `Grider`, via the established proxy, where they are decrypted and either displayed to the user or persisted in the federated file system.

As described in the LOKI protocol there are two major collaborations between the `Spacer` and the `Caller` (see Fig. 7). The first is for the actual establishment of the group and exchange of member information. The second is for the processing of the initial job created by the `Grider`. The only difference between the LOKI execution collaboration and the standard execution collaboration (Section A) is the encrypting and decrypting of the component exertions in the job itself.

will walk through Fig. 7, step by step in order to visualize the execution process:

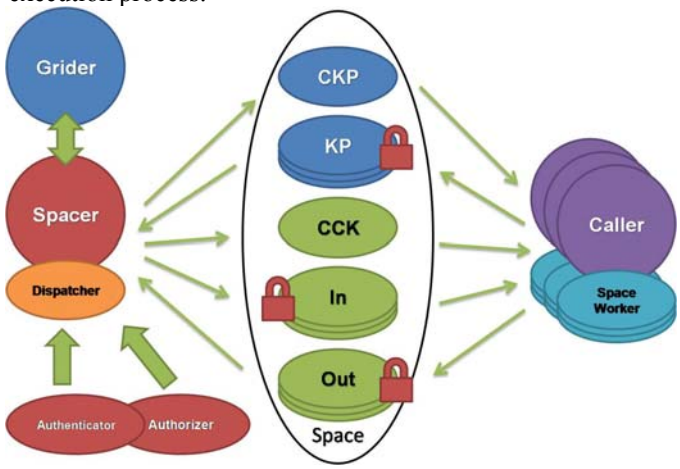


Fig. 7. LOKI cGrid collaboration

Step 1: Grider creates job and passes it to Spacer

Step 2: Spacer analyzes job, and writes CKP for every prospective group member

Step 3: Prospective member's SpaceWorker writes its KeyPair encrypted by group creator's PublicKey

Step 4: Spacer takes all KP exertions, computes CCKs, packs CCKs in CCK exertion and writes CCK exertion

Step 5: Spacer drops group wide encrypted job

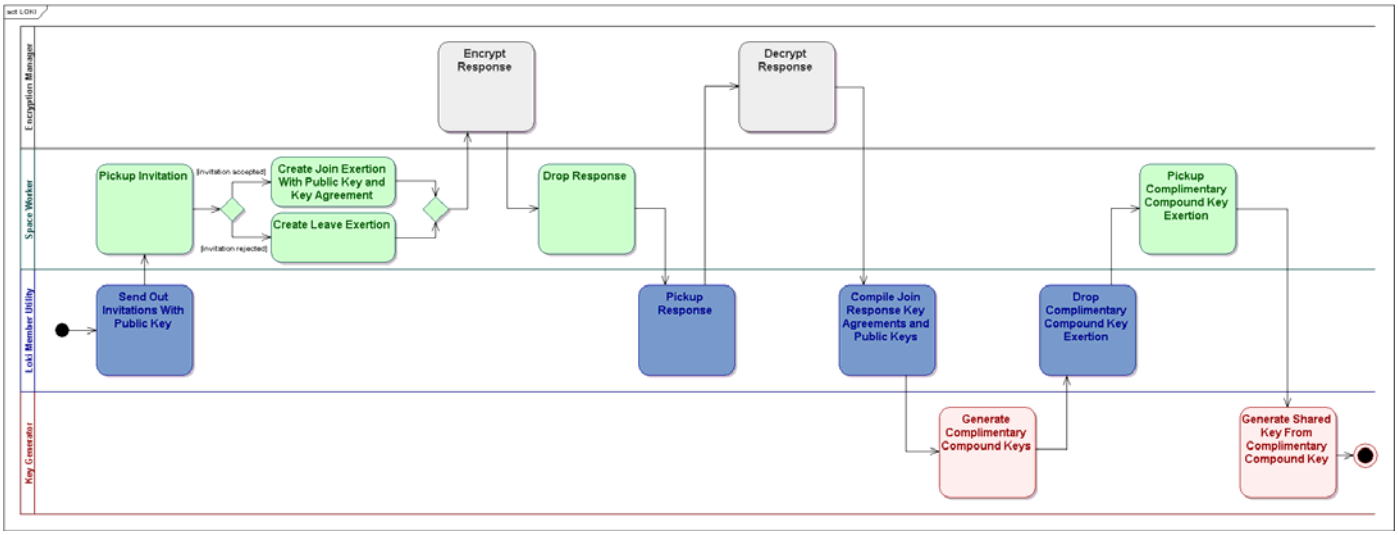
Step 6: Member's SpaceWorker reads CCK exertion

Step 7: Member's SpaceWorker takes encrypted job and decrypts with group wide shared key

Step 8: Member's Space Worker computes results, encrypts results and write to space.

Step 9: Spacer takes results and decrypt them with the group wide shared key

Step 10: Spacer passes the results to the Grider



C. *Security Flow Control*

Vital to the credibility of the LOKI framework, security needs to be maintained from start to finish. In order to verify that all links of the communication are secured in the LOKI implementation we must refer to Fig. 7.

The lcGrid job execution starts with `Grider` service, which then communicates with the `Spacer` service with the help of `ServicerAccesor`. The `Grider` to `Spacer` communication is secured with a two party Diffie Hellman key agreement. The Diffie Hellman agreement utilizes the `Grider`'s `KeyPair` and the `Spacer`'s `KeyPair` to individually calculate a common shared secret key. The shared secret key is then used to encrypt exertions exchanged between the two services.

The next link in the communication of job execution is that of the `Spacer` to `Caller` link, which passes over the exertion space. The `Spacer` to `Caller` communication is secured using the LOKI group key protocol (Fig. 8), and scalable key agreement. The LOKI protocol outlines the procedure for group creation, as well as the modified space interaction methods. These methods utilize the group's shared secret key to encrypt exertions passed between services over the exertion space.

The last possible gap in the job execution communication is in the concept of the invitation sent out by the group requestor. In order to ensure security, the validity of the response to the invitation needs to be authenticated. In order to do this within SORCER, the `Authenticator` service is utilized [[2]]. The `Authenticator` service guarantees that the service that has responded is the service that was invited.

VIII. CONCLUSION

The first objective was to secure the inherently public space computing environment. Through the implementation it has been shown that the LOKI framework successfully secures the lcGrid execution environment. The encryption and decryption of exertions with a securely created group wide shared secret key, before any exertion reaches the public space, ensures that exertions stay secure.

The second objective is to maintain security in the ad hoc space computing environment. The creation of the group key, accounts for the ad hoc nature of the system. The key is created and members who belong to the group can come and go, but will not be given access to the group if they establish new group characteristics (i.e. unique identifier, `KeyPair`, and `KeyAgreement`). The key agreement that is created in the framework is scalable to any number of party members, which satisfies the third objective—to abstract the complexity of the N-ary key policy.

As well the LOKI framework provides persistent data storage, management of group services, and is inherently easy to use. The solution also maintains several important architectural characteristics such as availability through extensive fault maintenance, modifiability through the use of interfaces, performance through the use of Java, security which has been described previously, testability through the modular nature of the framework, and usability through complexity abstraction.

Although this is the case, the solution grows incrementally more complex with increasingly large groups. The framework still holds but the time complexity for groups over 1000 members will grow faster than the benefits of the framework provide. It is for this reason that we must conclude that the framework is complete and practical, but for groups of large numbers of members (greater than 1000), alternate formations should be explored. Such alternatives may be the concept of sub groups, where sub group's shared key is the member key in a parent group. This would dissolve the issue of increasing time complexity and allow for groups of sizes much larger than 1000 party members. Other alternatives may include the CCK component calculation through the creation of specialized fast CCK calculation service within the LOKI framework.

In conclusion the Loki framework successfully secures the ad hoc space computing environments, with management of group services with little to no restrictions.

REFERENCES

[1] M. Berger, and M. Sobolewski, "Lessons Learned from the SILENUS Federated File System," *Complex Systems Concurrent Engineering*, Loureiro, G. and Curran, R. (Eds.), Springer Verlag, ISBN: 978-1-84628-975-0, pp. 431-440, 2007.

[2] M. Berger, and M. Sobolewski, "Group-based Security in a Federated File System," *2nd Annual Symposium on Information Assurance*, Albany NY, June 6-7, 2007, pp. 56-63, 2007.

[3] E. Freeman, S. Hupfer, and K. Arnold, *JavaSpaces™ Principles, Patterns, and Practice*, Addison-Wesley, ISBN: 0-201-30955-6, 1999.

[4] J. Garms, D. Somerfield, *Java Security*, Wrox, 2001

[5] M.E. Hellman, "An Overview of Public Key Cryptography," *IEEE Communications Magazine* , pp. 42-49, 2002.

[6] "Jini architecture specification," Version 2.1. http://www.sun.com/software/jini/specs/jini1.2html/jini-title.html.

[7] M. Sobolewski, "Service-oriented Programming, SORCER Technical Report SL-TR-13," 2008. Available at: http://sorcer.cs.ttu.edu/publications/papers/2008/SL-TR-13.pdf.

[8] M. Sobolewski, "Federated Method Invocation with Exertions," *Proceedings of the 2007 IMCSIT Conference*, PTI Press, ISSN 1896-7094, pp. 765-778, 2007. http://sorcer.cs.ttu.edu/publications/papers/96.pdf

[9] M. Sobolewski, "SORCER: Computing and Metacomputing Intergrid," *10th International Conference on Enterprise Information Systems,* Barcelona, Spain, 2008. http://sorcer.cs.ttu.edu/publications/papers/2008/iceis-intergrid-08.pdf

[10] D. Thain, T. Tannenbaum, and M. Livny, "Condor and the Grid," In Fran Berman, Anthony J.G. Hey, and Geoffrey Fox, editors, *Grid Computing: Making The Global Infrastructure a Reality*. John Wiley, 2003.

[11] G. Wells, Coordination Languages: Back to the Future with Linda, Rhodes University.

[12] D.R. Kerr, "Space Computing with Group Key Agreement - Location Independent Group Key Interactive Management (LOKI)," Master's Thesis. http://etd.lib.ttu.edu/theses/available/etd-03132008-163802

Peer-to-peer (P2P) Simulation for Network Security

Daniel O. Rice and George Wright
Loyola College in Maryland
4501 N. Charles Street
Baltimore, MD 21286 USA
{drice2, geo} @loyola.edu

***Abstract*–Peer-to-peer (P2P) networks have become a primary propagation mechanism of malicious code through file sharing applications. Many of the modern day malicious codes are being custom designed and deployed to specifically target P2P networks. These codes are tailored to take advantage of the characteristics of P2P protocols and the resulting P2P networks that typically overlay existing networks like the Internet. This research paper discusses an ongoing investigation into a method for increasing P2P networks' resistance to malicious code propagation. The concept is shown through the simulation of P2P networks and the application of mechanisms that influence the topology of the network. We choose to demonstrate this on P2P networks because they have several important characteristics that may increase these networks' vulnerability to malicious code attacks (these are some of the same characteristics that make P2P so valuable). P2P networks are self-organizing through users' "sharing" choices, they are decentralized with a decreased emphasis on a central coordinating authority; and each node in a P2P network determines its own sharing parameters. This paper focuses on applying a simulation methodology to the creation of P2P networks using several Java classes. The model is validated by comparing simulation results to real-world P2P networks. P2P networks are simulated under two regimes. In the first, network nodes download files based on their experience only; the more traffic one node has with another, the more likely it is to seek downloads node. Under the second regime, a node chooses to do traffic with another depending on a price function. The price function in this work is linked to the Pearson coefficient, a measure that captures an important structural dimension of how the network is connected. The research in this paper concludes that the application of a pricing mechanism can lead to networks that evolve in a significantly different way. Ongoing research will determine if the application of a pricing, or other mechanisms, can create networks more resistant to the propagation of malicious code.**

I. INTRODUCTION

Peer-to-peer (P2P) networks are a member of a much larger family of networks often referred to as distributed transient networks (DTNs). P2P networks exhibit behavior typically leading to three characteristics:

Self-organization through user choices in sharing of resources and services (files, storage, processor);

Decentralization due to the decreased emphasis on a central coordinating authority; and, Autonomy in that each node in a P2P network determines when and how much the node will make use of resources available on other nodes and how available it will make resources that it possesses. [10]

It follows that P2P networks tend to be "more scalable, robust, and adaptive than other forms of distributive systems" and particularly "difficult to study due to their size and the complex interdependencies between users, application, protocol and network." [12]

In the literature, P2P simulators have been developed to test networks and protocols. [12] However, many of these simulators have been designed to test very specific aspects of P2P networks. They do not capture even a fraction of the complexities inherent to these very large, diverse, and complex networks, nor are they amenable to validation. In this research we develop, test, and validate a network simulator that has been designed to capture the salient behavioral features of large complex P2P networks. Our immediate goal for this paper is to study the effect of pricing policies on the way network connectivity evolves. Our ultimate goal is to study the impact of security risks on these networks and the mitigating effects, if any, of pricing policies.

II. P2P NETWORK SECURITY – VIRUSES AND WORMS

In the first half of 2007, Symantec documented 212,101 new malicious code threats, a 185% increase over the previous period and a 318% increase over the first half of 2006. [11] These threats include trojans, worms, viruses, and backdoors. Symantec reports that of the malicious code that propagated in the first half of 2007, 22% did so through file sharing in general P2P networks, 18% in the Kazaa P2P network, 15% in the Morpheus P2P network, 15% in the eDonkey P2P network, and 5% in the Winny P2P network (shown in Fig. 1 below).

Rank	Propagation Mechanism	Percentage of Threats
1	File Transfer/Email Attachment	46%
2	File Transfer/CIFS	24%
3	File Sharing/Peer-to-Peer	22%
4	File Sharing/Executables	22%
5	File Sharing/Peer-to-Peer/Kazaa	18%
6	Remotely Exploitable Vulnerability	18%
7	File Sharing/Peer-to-Peer/Morpheus	15%
8	File Sharing/Peer-to-Peer/eDonkey	15%
9	File Sharing/Peer-to-Peer/Winny	5%
10	Backdoor/Kuang2	3%

Fig. 1. Propagation Mechanisms [11]

Also significant, as noted in the "Top ten new malicious code families" section of the Symantec report, contemporary "malicious code authors seem to be diversifying their propagation mechanisms by combining worms with a viral file-infection component."

As pointed out in [11], the obvious recommendation to contend with this growing malicious code threat is that administrators should look to block this type of activity using more specific port blocking used by these applications at the network boundary and protocol filtering (e.g., block all Kazaa traffic). Additionally, where possible P2P applications are should not be permitted on corporate networks, and these enterprises should take measures to prevent P2P clients from being installed on any computers on the network.

However, while this may be possible (albeit difficult) in some corporate network environments, it does restrict the advantages of using P2P in these networks (namely connectivity and convenience). Moreover, some networks cannot restrict P2P file sharing because it is becoming an important and acceptable use for most users. In these networks administrators should encourage end users who download files from P2P networks to scan all such files with a regularly updated antivirus product in order to mitigate the very real threat of malware.

III. P2P MALICIOUS CODE PROPAGATION AND COMPLEX NETWORKS

The emergent behavior of complex networks impacts network topology. For example, Internet topology is at least partially determined by how nodes in the network choose to connect to each other.

One way to describe evolving network topology is by looking at the connectivity of the individual nodes in the network. For example, the topology of the Internet can be described as having a power law degree distribution. [4] This simply means that if you count all of the links going in and out of each node in the network (also called the "degree" of the node), the distribution of the degree for all the nodes in the network would be described by a power law.

Practically speaking, this means that most nodes in the network have a relatively low degree while a few nodes in the network will be of a relatively high degree. We can depict this graphically by plotting node degree (the count of in-out links for each node) versus frequency (the count of the number of nodes of a given degree). When the degree (x-axis) versus frequency (y-axis) plot is fitted with a curve, that curve is best described by a power law (if we are dealing with a power-law degree distribution network).

Another aspect of network topology is a concept called "assortative mixing." [8] Assortative mixing refers to how nodes in the network preferentially attach to other nodes. If high degree nodes, nodes with a relatively high connectivity compared to other nodes in the network, have a preference to attach to other high degree nodes, then the network shows "assortative mixing." If high degree nodes tend to attach to low degree nodes the network shows "dis-assortative mixing." The "assortativeness" of a network can be calculated, as the Pearson product moment coefficient r, for any network. [8] The Pearson coefficient, *r*, has been empirically calculated for many existing networks, including social networks and the Internet (for which $r \approx -0.189$). [4]

In this research, a power-law degree distribution and Pearson's *r* were used to validate the properties of simulated P2P network topology and the model's behavior.

IV. EPIDEMIOLOGY AND COMPLEX NETWORKS

The epidemiological literature is full of studies that address the spread of the human epidemic process; that is, the spread of biological diseases, in populations through the interactions of individuals. Many of these studies use mathematical models to capture the spread of disease in human social networks. [2][5][9] Networks provide a convenient model to describe the social systems and epidemic process by which biological diseases spread. Most of the epidemic studies assume a heterogeneous or random network model. However, malicious computer codes spread in environments better characterized by scale-free networks. The understanding of scale-free connectivity is essential to understanding how malicious code propagates in computer networks. [9]

Two of the simplest epidemiological models are the susceptible-infected-susceptible (SIS) model and the susceptible-infected-removed (SIR) model. The SIS model models individuals who are either infected or susceptible for infection while the SIR models three states including susceptible, infected, and removed indicating immunity or death leading to an individual's removal from the network. The SIR model allows for individuals to be removed from the network in the case of death or acquired immunization. This fits our scenario, when infected computers are conceptually removed from the network when they either acquire immunity or die (e.g., from an unrecoverable crash). The SIR model will be applied in the simulation model in ongoing research into this area.

V. P2P SECURITY AND THE THEORY OF COMPLEX NETWORKS

The Internet and P2P networks have been shown exhibit approximately power-law degree distributions. These networks have also been shown to be extremely susceptible to virus and worm propagation. P2P viruses, such as the Kazaa, Grokster, iMesh and the Polip virus, are specifically designed to take advantage of P2P networks. [11]

In [6], the authors explore the "robust yet fragile" nature of the Internet. The Internet has evolved into a network that is generally unaffected by random component failures but is extremely vulnerable to targeted attacks.

Most social networks, on the other hand, are more robust and less fragile than the Internet. In [8], it is observed that "in social networks…diseases spread easily…however, this type of network has a small central set of people that the disease actually reach…they support epidemics…but the epidemic is limited in who it can reach."

It appears that social networks, and many other scale-free networks, are able to control epidemics organically through their structure. Interestingly, the network structure of the Internet and P2P render them much more vulnerable to the attacks of malicious code. Specifically, networks that have higher assortativeness, i.e., that have a higher Pearson's *r*, are more resistant to the propagation of infection. Networks with lower assortativeness, i.e., lower Pearson's *r*, are more prone to the propagation of infection. If a way could be found to influence the evolution of P2P networks' assortativeness toward higher Pearson's *r* values, the resistance of P2P networks to malware can be increased. Specifically, networks that have higher assortativeness, i.e., that have a higher Pearson's *r*, are more resistant to the propagation of infection. Networks with lower assortativeness, i.e., lower Pearson's *r*, are more prone to the propagation of infection. If a way could be found to influence the evolution of P2P networks' assortativeness toward higher Pearson's *r* values, the resistance of P2P networks to malware can be increased.

As previously mentioned the Pearson coefficient captures the level of assortative mixing in a network and is easily calculated for most networks. In this research, the Pearson's *r* is used in a pricing functions applied in the simulation to test how pricing in P2P can impact the topology of the network. In ongoing work simulation will be used to test a hypothesis that Pearson coefficient impacts the spread of viruses on the simulated P2P networks.

VI. SIMULATION

P2P network is similar to any other network simulation, except that it explicitly models the specific behavior of P2P networks mentioned above. That is, P2P network simulations model (1) self-organization of the network through users' "sharing" choices, (2) decentralization with a decreased emphasis on a central coordinating authority, (3) and each node in a P2P network determines its own sharing parameters. P2P network simulation must be validated to ensure they accurately model real world P2P networks. Specifically, the degree distributions of simulated P2P networks should be approximately power law. Additionally, the Pearson coefficient should be approximately -0.18 (approximating the Pearson of Internet, the network that P2P most commonly overlays).

The P2P simulator should also reflect the stochastic nature of real world P2P networks. This research presents a useful simulator that models P2P networks using several Java classes. The P2P network simulation model presented here is validated with respect to real-world P2P networks.

A. Approach

First, the P2P network simulation model is created using several Java classes, tested, and validated to ensure that it accurately reflects real-world P2P networks. Next, the simulation is tested under two regimes. In the first case, the network nodes download files based on experience, the count of files and the total number of bytes downloaded from a source. The more experience one node has with a source node, the more likely it is to seek more from that node. Under the second regime, a node chooses to download from a source node depending on a price function.

A Monte Carlo approach is taken where each model is simulated multiple times under both regimes and the results compared. The average Pearson's *r* is significantly different between the two regimes.

B. Model

The Java based P2P network simulator graphic user interface (GUI) is shown in Fig. 2 below.

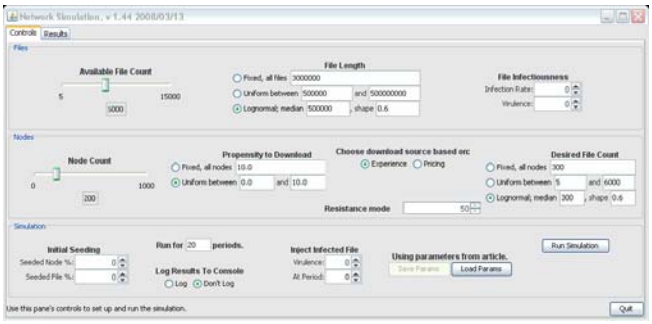


Fig. 2. Network Simulator Graphic User Interface

This figure provides a good basis for explaining the features and capabilities of the P2P network simulator. The first thing one might notice is that the GUI has two tabs, one for simulation controls, and the other for simulation results.

Simulation Controls Tab

This tab allows the user to set parameters and start the simulation. The simulator controls are grouped into three categories: Files, Nodes, and Simulation.

Files

This section of the GUI allows the user to set parameters for the simulated files. The Java class File represents the files that may be downloaded by nodes during a simulation run.

Table 1 shows the major attributes of the File class. Table 1 shows that each file has three major attributes.

TABLE 1
FILE CLASS DIAGRAM

File
infection : Infection
length : long
name : String

Infection

Each file has an Infection (another Java class, which may or may not be null). The major component of the Infection class is virulence. Virulence, which ranges from 0.0 to 100.0, is the percentage of uninfected files that an infected file will infect when it is introduced into a network node's directory. Here we mean "infection" to be an instance of malware that can spread across the network.

Length

This is the file's length or size in bytes. The GUI allows file length to be set in a number of ways. Our default is a lognormally distributed length.

Name

Each file has a unique alphabetic name, ranging from "aaaa" to "zzzz." The user can define the number of unique files available to the simulation run with the "Available File Count" slider. This control allows the user to provide enough distinct files to support a valid simulation while limiting the amount of memory required.

The "File Length" controls allow the user to specify one of three different ways for setting file length in bytes. This control is important because, under one simulation routine, network nodes choose to download file from another source node based on prior experience with the source node, where experience is quantified as files and bytes already downloaded from the source node.

The "File Infectiousness" controls determine the infection rate of the available files (from 0% to 100%) and the virulence of the infections that occur. When the stock of available files is generated at the beginning of the run, these controls affect how many files initially have a malware infection and how likely that infected file is to spread infection to other files in other nodes.

This and other class diagrams below are simplified. Only the attributes necessary for a quick explanation of the network simulator appear here.

Nodes

The middle segment of the control tab of the simulator GUI deals with nodes. A node is either a file-holding computer in a network or a user, depending on context. As a computer on the network, the node uploads or downloads files to or from other network nodes. As a user, the node makes decisions about what files to download and from which other node the files are downloaded. Table 2 shows the six major attributes of the Node class.

TABLE 2
NODE CLASS DIAGRAM

Node
connectedness : int
countFilesDesired : int
directory : Hashtable<String, File>
experienceTable : Hashtable<String, Total>
identifier : String
propensityToDownload : double
resistance : double

Connectedness

Each node keeps track of its participation in a network in terms of the count of edges for which the node is either the beginning or the end. An "edge" is the line between nodes in a network, indicating that one of the nodes has had with the other. Connectedness is updated by a node method (not shown in Table 2) each time a file download occurs.

Count of files desired

On creation, each node is endowed with a desired file count. This simulates the fact that some nodes may want to download more files than other nodes. This defaults to a lognormal distribution.

Directory

Each node maintains a collection of its files. The collection is a Java hash table, keyed on the file's name.

Experience

Since one of the simulation regimes requires nodes to choose download sources based on prior experience, each node keeps knowledge of its experience downloading from other nodes. Experience is kept in a Total object (not discussed here) and keyed on node identifier.

Identifier

Each node has a unique alphabetic identifier, ranging from "Aaa" to "Zzz."

Propensity to download

This is the percentage of the file deficit (difference between files held and files desired) that a node will decide to download in a period. It ranges from 1.0 to 100.0.

Resistance
The probability that a node will notice that an incoming file is infected and disinfect it.

The node class is a threaded class, allowing each node to be simulated on a separately executing Java thread. The first node control is the node count slider. This allows the user to select a count of nodes that provides a valid simulation while conserving memory and limiting run time.

The control for propensity to download the user options for establishing each node's propensity to download files when the node is initially created. The radio buttons for method of choosing download source establish which of two regimes is simulated. Under the "Node experience only" regime, a node downloading a file chooses from among all possible sources of that file based on its historical experience. Experience, in the current implementation, is the product of count of files downloaded from a node and the byte-count downloaded from a node. A downloading node, under this regime, will always download from a node with which it has had the most prior experience. (If a node has had no prior experience with possible sources of a desired file, it will download from a source at random.)

Under the "Network pricing" regime, a node downloading a file chooses to download from the source ordering the desired file at the lowest price. Price, discussed at greater length elsewhere [needs reference], is determined by the equation:

$$price = 200^{(20 - deltaR)} \qquad (1)$$

where deltaR is the change, if any, in Pearson's r that would occur if the downloading and source nodes become connected. This pricing function influences the network to evolve with a greater Pearson's r value.

The "Resistance Mode" spinner control sets resistance of a node to infection, ranging from 0–100%. This is set stochastically from a triangular distribution with a modal value set by the spinner.

The controls for desired file count allow the user to choose from among three methods for establishing, when a node is created, how many files it will want to download. This simulates the fact that some network nodes want to download more files than others. It defaults to a lognormal distribution.

Simulation

The bottom segment of the control tab of the GUI deals with parameters of the simulation itself. The user can seed nodes with an initial complement of files, establish the count of periods to simulate, and choose to inject a single infected file of set virulence in a particular period. There are also controls to save and load a configuration of all user-settable parameters from a disk file, along with buttons to start the simulation and exit the program.

The seeding controls allow the user to start the simulation with a set percentage of nodes already holding a set percentage of the files each node desires. This simulates the fact that some nodes joining a peer-to-peer network may already have a directory of many files.

The run length control allows the user to set the count of periods to be simulated. The user can set the count to allow both the model's behavior to stabilize and the run time to be reasonable.

A logging control allows the user to observe the behavior of the simulation as it unfolds. The simulator code has many console print statements, some selectively commented out, that report each significant simulator event to the console device.

The next set of controls allows the user to inject a single malware-infected file, of set virulence, to be inserted into the directory of a randomly chosen node. The user can then observe the spread of infection, in terms of the percentage of nodes whose directories contain at least one infected file.

The buttons to load and save parameters allow the user to save the values of all settable parameters for later re-use. This has proved valuable as we sought a configuration that reliably mimicked the known behavior of peer-to-peer network evolution as observed in the real world.

The bottom line of the controls tab contains a status line and a button that quits the program entirely.

Results Tab

Fig. 3 shows the display of the simulation results from the configuration of parameters

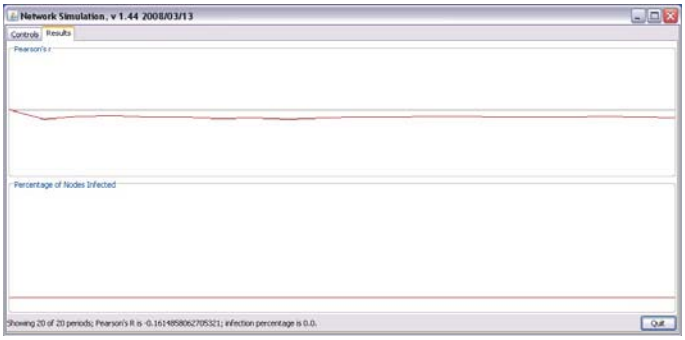


Fig. 3. Display of Network Simulation Results shown in Fig. 1.

Pearson's r

The top segment of the results tab shows how network interconnectedness, in terms of Pearson's r, has evolved during the simulation run. The x-axis varies to accommodate the count of simulation periods set by the user. The y-axis ranges from -1 to +1.

Percentage of Nodes Infected

The bottom segment of the results tab shows how infection evolves within the network. One way for malware

infection to enter the network is via files initialized with infection at the outset of the simulation. The other way is for a single infected file injected at random during the course of the run. The y-axis is the same as that of the Pearson's r display. The y-axis ranges from 0% to 100%, where the value is the percentage of nodes that have at least one infected file it their directory.

We do not explicitly explore the propagation of infection in this paper.

Status Line

The bottom line of the results tab is a status line and the usual quit button. The status line displays the current period's actual values for Pearson's r and infection percentage.

C. Simulation

When the user clicks the "Run Simulation" button on the controls tab of the simulator GUI, all the controls are disabled and the simulation begins. The following is sequence of events in the simulation.

Files are created. A master directory of all possible files, held by the simulator itself, is created and filled with files configured according to user-set parameters.

Nodes are created. A collection of nodes is created and filled with nodes configured according to user-set parameters.

A simulator instance, a Java threaded object, is created.

A network instance, another Java threaded object, is created, furnished with the collection of node instances, and installed as a member of the simulator object.

The simulator object is started. It immediately starts the network and waits for the network to complete its start-up.

The network, on start-up, starts each one of its nodes. The nodes are, at this point, not connected to each other at all yet.

After the network has started, control returns to the simulator object, which then simulates each period, up to the count of periods requested by the user. The activity that takes place in each simulated period is described below.

After all periods have been simulated, the final results (elapsed time of the run, final Pearson's r, final infection percentage, and matrix of node interconnectivity) are displayed. The model is set up for another run and the user controls are enabled.

The foregoing described the flow of control at the highest level of the simulation. We now must describe what takes place as the simulator thread, the master controller, simulates each period in turn. This is the sequence of events in each simulated period.

If the user has specified that an infected file be injected in this period, this specification is carried out. A node is selected at random. One of that node's files is selected at random. The selected file is infected as specified by the user.

The network then notifies each node to carry out its periodic behavior. The activity that each node carries out is described below.

When the network is notified that all nodes have completed their periodic behavior, the results display is updated for that period.

The real action of the simulator takes place as each node carries out its periodic behavior. Each node is a separately executing thread. When notified by the network that a new period has begun, each node's thread awakens and begins its periodic behavior, which consists of the following.

If the node's propensity to download is greater than a randomly generated number, it begins its downloading for the period.

If the node wants to download at all, it first checks to see how many desired files remain to be downloaded. If it has still has files to download, it determines the percentage of the remaining files it wants this period (based on its propensity to download).

For each file it wants to download, the node chooses a file at random from the master directory of all available files.

If the node already has the desired file, it counts that file against its count of files to seek for the period and looks for the next file. If the node doesn't have the desired file, it asks the network for a list of all possible sources of the file. The network returns a list of all sources, sorted either by experience or by price, as specified by the user.

If there is a source of the file, i.e., if the list returned by the network has at least one entry, the node downloads the file from the top entry on the list, updating its directory, its connectedness, and its experience accordingly. Otherwise the node simply gets the desired file from the network's master list of available files, simulating the process of obtaining a file in some other way than from a P2P network.

When the node has downloaded all the files it's looking for this period (or if its propensity to download was less than or equal to the randomly generated number in step 1 above), the node decrements the count of still-active nodes. If this particular node is the last active node this period, the network is notified that all nodes have finished their behavior for the current period.

This completes the description of the simulator's behavior. At this point, the simulator GUI's control tab looks as it does in Fig. 1 and its results tab looks as it does in Fig. 2. Of course, the results will vary from run to run, because each run is stochastic. During each simulation run, each node is a separate thread of execution, pursuing its behavior according to the parameters set by the user. We explore the random nature of the runs in the next section.

VII. RESULTS

Validation of the simulation model requires that sufficient evidence that the behavior of simulation model depicts real P2P network behavior. In this section, we describe model validation and then discuss the comparison of P2P networks generated under the two schemes: experience and pricing.

D. Power Law Degree Validation

Fig. 4 below shows the results of 187 simulation runs of 1000 node networks for 40 periods. These numerical results from our network simulation show that our simulated networks exhibit scale-free characteristics and a power-law degree distribution. Specifically, the node degree distribution exhibits a power law (the R^2 coefficient is 0.6317).

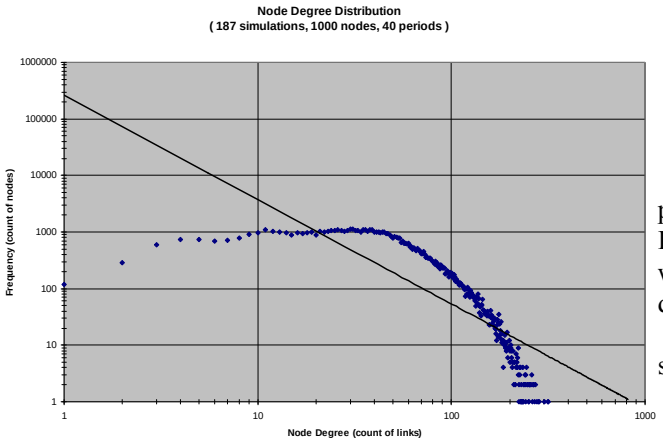


Fig. 4. Log-Log Plot Showing Power Law Degree Distribution

Simulation results show that the P2P simulation model degree distribution is approximately power-law, moving towards the validation of the model.

E. Assortativeness Validation

For validation purposes, we make the assumption that the Pearson of the P2P simulation model should approximate that of the Internet, where r = -0.189. [4] Numerous simulation runs have been used to determine that Pearson is highly sensitive to the configuration of user-set parameters in the simulation. The initial node seeding, the number of files that each node has at the beginning of the simulation, has the greatest impact Pearson's r, and experimentation showed that the hardest part of choosing an initial configuration was in setting node seeding. We would expect that some nodes entering a P2P network would already have a complement of desired files. Others would have few or none. Still others would have almost all the files one could expect them to want. But what combination of seeding (percentage of nodes seeded and percent of desired file complement seeded) would yield a realistic Pearson's r?

We used Monte Carlo simulation to answer this question. We using the general parameter configuration of Fig. 2, we simulated the model ten times each for various combinations of percent nodes seeded and percent files seeded. The percent of nodes seeded varied from 10% to 100% in increments of 10%; the percent of files seeded varied from 0% to 100% in increments of 10%. The results appear in Table 3.

TABLE 3
AVERAGE PEARSON'S R OVER 10 SIMULATIONS W/ VARIED SEEDING

		Percent Files Seeded										
		0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Percent Nodes Seeded	10%	-0.141	-0.144	-0.179	-0.228	-0.265	-0.306	-0.312	-0.387	-0.391	-0.451	-0.477
	20%	-0.142	-0.123	-0.131	-0.164	-0.211	-0.247	-0.272	-0.301	-0.349	-0.365	-0.406
	30%	-0.125	-0.117	-0.115	-0.127	-0.134	-0.182	**-0.186**	-0.217	-0.236	-0.246	-0.274
	40%	-0.135	-0.128	-0.108	-0.117	-0.127	-0.126	-0.143	-0.163	-0.179	-0.194	-0.206
	50%	-0.139	-0.145	-0.119	-0.117	-0.110	-0.122	-0.143	-0.151	-0.148	-0.198	-0.209
	60%	-0.139	-0.164	-0.141	-0.107	-0.114	-0.118	-0.144	-0.163	-0.166	-0.180	-0.183
	70%	-0.139	-0.165	-0.135	-0.133	-0.133	-0.124	-0.139	-0.160	-0.206	-0.202	-0.201
	80%	-0.148	-0.173	-0.150	-0.123	-0.126	-0.115	-0.154	-0.166	-0.163	-0.188	-0.217
	90%	-0.146	-0.188	-0.167	-0.132	-0.128	-0.126	-0.154	-0.149	-0.197	-0.171	-0.206
	100%	-0.134	-0.194	-0.173	-0.132	-0.120	-0.129	-0.163	-0.167	-0.182	-0.226	-0.253

Our objective is to find a seeding configuration that produces valid results, to wit, a Pearson's r of about -0.189. It appears from the table that this is about the average result we get with 30% of the nodes seeded with 60% of their desired file complement.

Another way of looking at the data in Table 3 is the surface shown in Fig. 5.

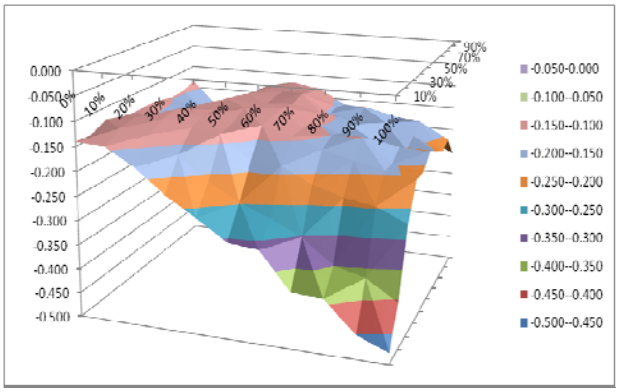


Fig. 5. Average Pearson's r Surface over Ten Simulations with Varied Seeding

Fig. 5 shows the contour of valid initial seeding configurations as light blue (-0.200–-0.150).

F. Application

The configuration shown in Fig. 2 is used with the seeding at the parameter level shown in Table 3 (shown highlighted in yellow, bolded and underlined in Table 3, where 30% of the nodes seeded with 60% of their desired files) to compare the evolution of a P2P network under

experience-based selection and price-based selection. We simulated the configuration 30 times, resulting in a mean Pearson's r of -0.190, standard deviation 0.0326.

The hypothesis that the mean Pearson's r is the nominal -0.189, cannot be rejected. However, is not alarming because exactly matching the Pearson is not absolutely necessary, the goal is to get close which is why that configuration was chosen. The z-value is z = -0.175.

However, more importantly are the results when the configuration as in the paragraph above is used in the pricing case, when the "Pricing" radio button in the "Nodes" section of Fig. 2 checked. That is, the nodes are now choosing to download files from peer nodes based not on experience but on choosing a source node with the lowest value of the price function. As before, we simulated the configuration 30 times, resulting in a mean Pearson's r of 0.107, standard deviation 0.0265.

The hypothesis that the mean Pearson's r under experience-based choice, -0.190, is the same as the mean Pearson's r under price-based choice, 0.107 is rejected (z = -38.737). We conclude that a pricing function that charges more for download choices that decrease the Pearson's r of a P2P network influences node behavior as desired.

G. Continuing Research

The model and set of initial configuration parameters have shown under simulation to faithfully reproduce the behavior of P2P networks on the Internet, when node choice of download source is made on the basis of prior experience. Furthermore, there is confidence exhibited in the model that a pricing function devised to increase the Pearson's r of the evolving network will shape the network as expected. Prior research predicts that that propagation of malware will be less on a price-based network than on an experience-based network. This is what we intend to investigate next.

REFERENCES

[1] R. Anderson, R. May, and B. Anderson, Infectious *Diseases of Humans: Dynamics and Control*, Oxford University Press, Oxford, UK, 1992.

[2] N. Bailey, *The Mathematical Theory of Infectious Diseases*, Hafner Press/ MacMillian Pub. Co.; 2nd edition, London, 1975.

[3] D. Chang, and C. Young, *Infection Dynamics on the Internet*, Computers and Security, Vol. 24, p. 280-286, 2005.

[4] Q. Chen, H. Chang, R. Govidan, S, Jamin, S. Shenker, and W. Willinger, *The origin of power laws in Internet topologies revisited*, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, New York, 2002.

[5] O. Diekmann and J. Heesterbeek, Mathematical Epidemiology of Infectious Diseases: Model Building, Analysis and Interpretation, John Wiley & Sons, New York, 2000.

[6] J. Doyle, D. Alderson, L. Li, S. Low, M. Roughan,, S. Shalunov, R. Tanaka, and W. Willinger, *The "Robust Yet Fragile" Nature of the Internet*, California Institute of Technology Working Paper, obtained March 2005.

[7] F-Secure Virus Descriptions : P2P-Worm, F-Secure Computer Virus Information Pages, http://www.f-secure.com/v-descs/p2pworm.shtml accessed on November 17th, 2005.

[8] M. Newman, Assortative Mixing in Networks, Physical Review Letters, Vol. 89, No. 20, November 2002.

[9] R. Pastor-Satorras, and A. Vespignani, *Epidemics and Immunization in Scale-Free Networks*, published in the Handbook of Graphs and Networks: from the Genome to the Internet, eds. Bornholdt, S., and Schuster, H., Wiley-VCH, Darmstadt, Germany, 2003.

[10] D. Schoder, K. Fischbach, and C. Schmitt, *Core Concepts in Peer-to-Peer (P2P) Networking*, in R. Subramanian and B.Goodman (eds.): *P2P Computing: The Evolution of a Disruptive Technology*, Idea Group Inc, Hershey, 2005.

[11] Symantec Internet Security Report: Trends for January 06 -- June 06, Volume X, Published September 2006, editors D. Turner and S. Entwisle, available at www.symantec.com.

[12] N. Ting and R. Deters, 3LS – *A Peer-to-Peer Network Simulator*, Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P'03)

Keynote (D-2): Securing the Internet Infrastructure: Issues and Problems

John Crain, Chief Technical Officer
ICANN (Internet Corporation for Assigned Names and Numbers)

The Internet has become a part of our everyday lives. It is only a half a century old, but it is difficult to imagine life without it. It has become a world in itself. We have postal addresses on the Internet just like in real life. What will happen when we run out of addresses? Who owns the Internet? How secure is it? How likely would it be for terrorists to disable it and how do we go about preventing this? We need to seriously consider these questions rather than waiting for them to become a larger problem. This talk busts a lot of myths about the Internet and provides a balanced perspective on its reliability and security around the world.

Roundtable Discussion: Challenges of Computer and Digital Forensics Training and Education

Fabio R. Auffant II
Technical Lieutenant, Computer Crime Unit, NYS Police

Cristian Balan
Program Director, Computer and Digital Forensics Program, Champlain College

Sean Smith
Technical Resource Attorney NY Prosecutors Training Institute

The field of computer and digital forensics is changing rapidly and our dependence on computers and network is increasing. Techniques from computer and digital forensics are being used not only for investigating crime, but also for auditing systems as well as for recovery of lost data. Computer and digital forensics involves data not only from computers, but also from servers, networks, and mobile devices. The needs of the public sector workforce are growing as the demand for such expertise increases within existing IT departments and new forensics divisions are created in agencies. However, they are competing with the private sector, which often lure prospective employees with better salaries. Knowledge of computer and digital forensics has become a necessary component of any IT specialist, but due to the changing environment, it is also important to adapt by continuing to learn new tools and techniques.

In this roundtable, the panel will discuss the challenges faced by educators/trainers, law enforcement, and prosecution in terms of training, retraining, and retaining a computer and digital forensics capable workforce. It will also cover novel ways to ensure continuous training to security and forensics professionals. The three panelists at the roundtable bring different perspectives to the discussion. Fabio R. Auffant II is a forensics expert in the Computer Crime Unit of the NYS Police, Sean Smith is a technical resource attorney from the NY Prosecutors Training Institute, and Christian Balan is the director of the Computer and Digital Forensics Program at Champlain College.

On Information Assurance in Nanoscale Networks

Stephen F. Bush, *Senior Member, IEEE*
GE Global Research Center, Niskayuna, NY, 12309, USA
bushsf@research.ge.com

***Abstract*–An intersection of two worlds, emerging nano-technologies and network/communication theory, is poised to change the nature of information assurance. New communication paradigms will be derived from the transition from micro- to nano-scale devices. The related degrees of freedom and constraints associated with these new technologies will change our notions about efficient networks, system design and the nature of information assurance. Work is ongoing on a multi-disciplinary front towards new techniques in modeling, design, simulation, and fabrication of network and communication systems at the nano-scale. This paper reviews the state of the art and considers the challenges and implications for information assurance.**

***Index Terms*–nanotechnology communication networks, and information assurance.**

I. INTRODUCTION

Networks communicating nformation exist on a nanoscale [1]. Interconnected carbon nanotubes, micrometers in length and nanometers in diameter, convey signals across areas of tens of square micrometers [25]. The Nano-Net conference [35] focuses on aspects familiar to those researching today's macro-scale communication systems such as efficient coding, routing, quality of service, but within nanoscale networks. Wireless transmission and reception among components on a single chip have been designed in [34] and patented in [33]. Nanoscale wireless security issues will need to be addressed at some point. Thus, while it is still early in the development of nanoscale networks, it may be worth considering that information warfare and network security may have to be considered at the nanoscale, just as they are on the macro scale.

Information security is typically comprised of *confidentiality*: assurance that information is shared only among authorized persons or organizations, *integrity*: assurance that the information is authentic and complete, and *availability*: assurance that the systems responsible for delivering, storing and processing information are accessible when needed, by those who need them. How are these impacted when entire, or at least significant portions, of communication networks are reduced to the nano scale?

First, consider the impact of the extreme difference in scale between today's networks and nanoscale networks. In Fig. 1 the size of a wireless mote sensor is to a nanotube as the length of a large bridge (or approximately an Ethernet segment) is to a finger on the human hand. Thus, it is clearly much easier to manipulate and replace components in today's Internet.

In terms of nanoscale sensor networks, the network components are on the same scale as the individual molecules of the sensed elements. This close relationship in scale between sensed targets and the communication network has significant implications for information security. Management of the complexity of such systems becomes significantly more difficult. The ability to detect and mitigate malicious behavior is thus more difficult. The problems are twofold: (1) the significant increase in the complexity of nano-scale systems due to their larger number of components within a compact space (2) the mismatch in the size of the networking components, making them individually more difficult to detect and handle.

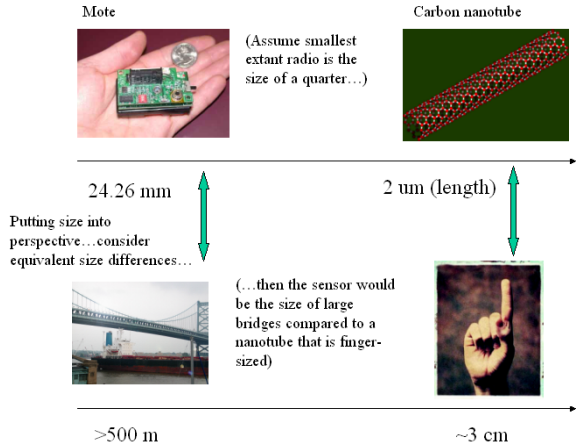


Fig. 1. Comparison of macro and nanoscale networking. The size of a wireless mote sensor is to a nanotube as the length of a large bridge (or an Ethernet segment) is to a finger on the human hand.

As specific examples, consider several manifestations of nanoscale networks: (1) biological networking (2) nanotube interconnections (3) quantum communication. The nature of attacks at the nanoscale will utilize the nature of both the small scale and the strong relationship to fundamental physical objects: real viruses in biological systems may compromise a molecular communication system, eavesdropping may occur by tapping into nanoscale networks with the attacker's network sensors at the same scale, denial of service may be accomplished by flooding nanoscale networks with small physical matter, careful and controlled induced faults in the physical nature of the

nanoscale network in order to discreetly corrupt the integrity of the information.

II. NANOROBOTICS AND SECURITY

There are two major research thrust areas [31]. The first area deals with design, simulation, control, and coordination of robots with nanoscale dimensions. The second research area focuses on overall miniaturization of mobile robots down to μm overall sizes. Nanorobots, nanomachines, and other nanosystems are objects with overall dimensions at or below the micrometer range and are made of assemblies of nanoscale components with individual dimensions ranging approximately between 1 to 100 nm. In these mobile robotic systems, overall system size is very limited, which induces severe constraints in actuators, sensors, and motion mechanisms; power sources, computing power, and wireless communication capability. When scaling down, the surface-to-volume ratio increases and surface forces dominate volume-based forces. At nm scales, inter-atomic forces or surface chemistry plays a significant role in robot mechanics. Thus, inertial forces and weight are almost negligible and micro/nanoscale surface inter-atomic forces, fluid dynamics, heat transfer, surface chemistry, and adhesion-based contact mechanics and friction dominate robot mechanics. These micro/nanoscale forces have many different characteristics compared to macroscale ones [30]. Our focus is on the information transmission among such nanomachines [32] and whether the nanoscale forces have an impact upon the fundamentals of communication and its corresponding security. Research into nanorobotics is well underway [29] and one can easily imagine such robots programmed to carry out the mission of discreetly compromising a nanoscale network. Defense against nanorobots is likely to lead towards a new integration of information and physical security.

III. NANOSCALE NETWORKS

Source and channel coding as well as cryptography require computational overhead which (1) grow very rapidly with the large scale of nano networks and (2) network processing power is reduced at the nanoscale because there is limited processing that can be packed into a ever smaller volumes. Given this limitation, more of the computation will have to be done by non-traditional means, perhaps by utilizing network topology as part of the computation.

Nanoscale networking has been driven by several factors, a significant one being the fact that industry is reaching limits regarding the speed of processors that can be placed onto an integrated circuit chip with acceptable properties of power consumption, current leakage, and heat dissipation. This is leading to new multi-core architectures, where a multi-core processor is an integrated circuit (IC) to which many, sometimes in the hundreds, of processors have been attached for enhanced performance, reduced power consumption, and more efficient simultaneous processing of multiple tasks. A multiple core set-up is somewhat comparable to having multiple, separate processors in the same chip. Multi-core processing is a growing industry trend as single core processors rapidly reach the physical limits of possible complexity and speed.

The current means of connecting elements on a chip will prove insufficient as chips advance to include many independent processing elements (PEs). This motivates research into various forms of networks on chip (NoC) to connect the PEs. Another term often used is system on chip (SoC), which refers to the integration of an entire macroscopic device, such as general-purpose computer, on a single chip. In the short term, current lithography-based approaches will continue to evolve to fabricate chips and only the architecture of the chips will change. However, longer-term at the scale of 22 nanometers and less, current techniques simply cannot be used to produce large-scale integrated circuits. Here, the Nano-Net conference has provided a venue for novel ideas for fabricating computing devices, such as combining self-assembled DNA structures with processing and communication elements based on carbon nanotubes (CNTs).

A. Carbon Nanotube Networks

Current computer chips are fabricated with lithographic techniques operating at 65 nm with predictions for 45-nm scale chips in 2008 [1]. The industry roadmap predicts that in 2018 feature size will reach 16 nm; however, no currently known process can reliably produce this scale of interconnects in mass quantity [1]. Researchers are now looking towards carbon nanotubes to achieve this objective. Currently, the resulting population of carbon nanotubes (CNTs) is highly variable. This is a basis for considering long-term approaches based on self-assembly of DNA and integration with CNTs.

A carbon nanotube (CNT) is a sequence of carbon atoms (C60), which are arranged into a long thin cylinder with a diameter of approximately one nanometer [2]. The atomic structure of CNTs makes them mechanically strong and the atomic properties lead them to be conductors of electric current. Researchers have used CNTs to construct various electronic components, such as resistors, capacitors, inductors, diodes and transistors [2]. CNTs, which appear as rolled tubes of graphite (graphene) with walls constructed from hexagonal carbon rings, can be formed into large bundles (much as typical electronic wires can be bundled) [4]. CNTs come in two general forms: single-walled (SWNTs) and multi-walled (MWNTs). SWNTs have only a single tube, while MWNTs consist of concentric tubes [4].

B. Molecular Communication

Molecular communication aims to allow nanoscale machines to communicate using molecules as a carrier to

convey information. [6] "Molecular communication is inspired by the observation that in biological systems, communication is typically done through molecules. For instance, biological systems perform intra-cellular communication through vesicle transport, inter-cellular communication through neurotransmitters, and inter-organ communication through hormones. Current nano and biotechnology focus on observation and understanding of existing biological systems such as how communication is done within a cell or between cells. Molecular communication would work toward the actual design and control of a nano-scale communication system." [6] The fundamental research issues include: (a) controlling propagation of carrier molecules, (b) encoding and decoding information onto molecules and (c) achieving transmission and reception of carrier and information molecules.

The aim is to achieve communication over 10's of micrometers using carrier molecules, such as molecular motors, hormones or neurotransmitters. Information is encoded as proteins, ions or DNA molecules. The environment is taken to be the aqueous solution found within and between typical cells. [6] One can imagine a variety of new information assurance issues for this type of media.

C. *Solid-State Quantum Devices*

Another approach to nanoscale electronics is to exploit devices based on quantum effects. These include tunneling diodes, single-electron transistors and quantum dots [3]. It is well known that quantum devices are sensitive to noise and, if one assumes lithographic techniques for interconnection, would be highly sensitive to lithographic accuracy since quantum devices operate on the scale of one or a few electrons.

With regard to information assurance, quantum devices may enable the use of quantum cryptographic techniques to improve information assurance at the nanoscale level. The BB84 [39] quantum key distribution scheme developed by Charles Bennett and Gilles Brassard in 1984 is a well-known example. The protocol is provably secure, relying on the quantum property that information gain is only possible at the expense of disturbing the signal if the two states we are trying to distinguish are not orthogonal. Quantum approaches to information assurance are growing rapidly in both macroscale and nanoscale networks.

IV. NETWORKING ON A CHIP

Solutions from macroscale wide-area networking are being proposed for use in on-chip networks. The implementations for the routers vary widely using techniques of packet or circuit switching, dynamic or static scheduling, wormhole or virtual-cut through routing. The majority of the current router implementations for network-on-chip are based on packet-switched, synchronous networks." [7]

Some research has proposed an NoC topology and architecture that injects data into the network using four sub-NICs (Network Interface Controllers), rather than one NIC, per node. This scheme achieves significant improvements in nano-network latency and energy consumption with only negligible area overhead and complexity over existing architectures. In fact, in the case of MESH network topologies, the proposed scheme provides substantial savings in area as well, because it requires fewer routers.

Another theme that drives research in on-chip networks is the likelihood that production of chips with massive numbers of processing elements and interconnections will increase uncertainty with respect to on-chip properties. Researchers following this theme begin to address issues that will also be of concern in the long-term for self-assembled systems. For example, some links might be so long that communications between PEs cannot occur in a single clock cycle [13]. In other cases, chip properties might lead to transient, intermittent or permanent communication errors [14]. Other research considers how to operate a chip when dimensions are so small as to preclude distribution of a reliable clock [15]. Such uncertainty leads researchers to propose various schemes for robust on-chip communications [16-18].

V. ACTIVE NETWORKING AT THE NANOSCALE

Active networks [37] [38] at the macroscale is a network paradigm in which intermediate network nodes—for example, switches, routers, hubs, bridges, gateways etc.—perform customized computation on packets flowing through them. The network is called "active" because new computations are injected into nodes dynamically, altering the behavior of the network. Packets in an active network can carry program code in addition to data. Customized computation is embedded within the packet's code, which is executed on network nodes. By making network node computation application-specific, applications using the network can customize network behavior to suit their requirements.

A similar concept is seen in [36] where an active network architecture at the nanoscale is used to solve the problem of limited node size, which prevents the design of a single node that can perform all operations. Instead, DNA self-assembly designs different node types (e.g., add, memory, shift) based on node size constraints. A configuration phase at system startup determines defective nodes and links, organizes a memory system, and sets up routing in the network. When executed, an instruction searches for a node with the appropriate functionality (e.g., add), performs its operation, and passes its result to the next dependent instruction. In this active network execution model, the accumulator and all operands are stored within a packet, a

hallmark of macroscale active networks, rather than at specific nodes, thus reducing per-node resource demands. This enables the encoding of a series of dependent instructions within a single packet. Thus, the security techniques used to assure information in macroscale active networks might be called upon to help solve nanoscale active networks.

VI. SELF-ASSEMBLY AND INFORMATION ASSURANCE

Tags are used in DNA self-assembly to stimulate the construction of structures with specific properties. Once a DNA structure exists, other organic components can be attached to the structure and the attached components can be interconnected with communication links, perhaps composed of CNTs, to construct the functional equivalent of a computer chip, including large numbers of processing elements. For the short-term, DNA-based self-assembly is likely to be restricted to two layers. [5]

Alignment of Carbon nanotubes has been the topic of vigorous research. Cost and separation of impurities, namely metallic tubes, is still an unsolved problem. In the approach proposed by GE, lower-cost, randomly oriented tubes are directly utilized as a communication media. [23] Information flow through a CNT network may be controlled in spite of the random nature of tube alignment. The same technique used for sensing in CNT networks, namely, change in resistance of semiconducting material, may be used to effectively route information. The traditional networking protocol stack is inverted in this approach because, rather than the network layer being logically positioned above the physical and link layers, the CNT network and routing of information is an integral part of the physical layer. The potential benefits of better utilizing individual nanotubes within random carbon nanotube networks (CNT) to carry information is distinct from traditional, potentially less efficient and wasteful, approaches of using CNT networks to construct transistors which are then used to implement communication networks. [24]

Self-assembly is currently limited to producing small sized DNA lattices thus limiting circuit size. However, the parallel nature of self-assembly enables the construction of a large number (~109-1012) of nodes that may be linked together by self-assembled conducting nanowires." [26] This implies that control over the production process (for node placement, node orientation, and inter-node link creation) would be quite imprecise. Resulting devices produced by the same process could differ distinctly. Systems created using such techniques would need to discover the placement, orientation and connection among nodes and organize their run-time processes to take maximum advantage of the characteristics of the system. Different systems, created with the same processes, could yield devices with varying capabilities. The same technique that would enable systems to determine and utilize a "nano-network" might also be used to attack such a network.

Alternatively, self-assembled systems might be considered as stochastic systems whose performance envelops can be described only probabilistically. Ultimately, self-assembly at the nanoscale seems destined to create systems with intrinsic defects. Two types of defects have been noted: functional and positional. A functional defect corresponds to a component that does not perform its specified function and a positional defect corresponds to a potentially functionally correct component that is placed incorrectly. This implies that nanoscale systems must be designed with fault tolerance as a fundamental property.

Nanotechnology provides smaller, faster, and lower energy devices, which allow more powerful and compact circuitry; however, these benefits come with a cost—the nanoscale devices may be less reliable. Thermal- and shot-noise estimations alone suggest that the transient fault rate of an individual nanoscale device may be orders of magnitude higher than today's devices. As a result, one can expect combinational logic to be susceptible to transient faults, not just the storage and communication systems. Therefore, to build fault-tolerant nanoscale systems, one must protect both combinational logic and memory against transient faults. Based on these assumptions, researchers are investigating error-correcting codes that can work effectively under the higher error rates expected from nanoscale memories. [19]

VII. CONCLUSIONS

Nanoscale networking is still in its infancy, however, it may not be too early to begin outlining the potential information assurance challenges that such networks will have. This paper attempts to layout the current nanonetworking approaches and identify aspects of their security.

REFERENCES

[1] The Third International Conference on Nano-Networks (Nano-Nets 2008), Sept 15-17, Boston, MA, http://nanonets.org/cfp.shtml.

[2] Adamson, B., Bormann, C., Handley, M., Macker, J., Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol, IETF RFC 3940, November 2004.

[3] http://en.wikipedia.org/wiki/Semiconductor_fabrication

[4] http://www.webopedia.com/TERM/C/Carbon_Nanotube_Technology.html

[5] T. Raja, V. D. Agrawal, M. Bushnell, A Tutorial on Emerging Nanotechnology Devices, 17th International Conference on VLSI Design, Jan. 7, 2004.

[6] http://www.sigmaaldrich.com/Area_of_Interest/Chemistry/Materials_Science/Nanomaterials/Tutorial.html

[7] J. Patwardhan, ARCHITECTURES FOR NANOSCALE DEVICES, PhD Thesis, Department of Computer Science, Duke University, 2006.

[8] S. Hiyama, Y. Moritani, T. Suda, R. Egashira, A. Enomoto, M. Moore and T. Nakano, “Molecular Communication”, Proceedings of Nanotechnology 2005.
[9] P. Wolkotte, G. Smit, G. Rauwerda, L. Smit, “An Energy-Efficient Reconfigurable Circuit-Switched Network-on- Chip”, in Proceedings of the 19th IEEE Parallel and Distributed Processing Symposium, April 2005.
[10] P. Meloni, S. Murali, S. Carta, M. Camplani, L. Raffo, G. De Micheli, “ Routing Aware Switch Hardware Customization for Networks on Chips”, Proceedings of the 1st International Conference on Nano-Networks and Workshops, September 2006.
[11] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, C. Da, “A Distributed Multi-Point Network Interface for Low- Latency, Deadlock-Free On-Chip Interconnects”, Proceedings of the 1st International Conference on Nano-Networks and Workshops, September 2006.
[12] I. O'Connor and F. Gaffiot, “ADVANCED RESEARCH IN ON-CHIP OPTICAL INTERCONNECTS”, report from a research conducted under an EU project: Photonic Interconnect Layer on CMOS by waferscale integration, circa 2005.
[13] A. Bartzas, N. Skalis, K. Siozios, D. Soudris, “Exploration of Alternative Topologies for Application-Specific 3D Networks-on-Chip”, Proceedings of the Workshop on Application -Specific Processors, 2007.
[14] K. Nomura, K. Abe, S. Fujita, A. Detion, “Novel Design of Three-Dimensional Crossbar for Future Network on Chip based on Post-Silicon Devices”, Proceedings of the 1st International Conference on Nano-Networks and Workshops, September 2006.
[15] M. Ghoneima, Y. Ismail, M. Khellah, V. De, “Variation-Tolerant and Low-Power Source-Synchronous Multicycle On-Chip Interconnection Scheme”, in Networks-on-Chip, special issue of VLSI Design, Hindawi Publishing Corporation, 2007.
[16] T. Lehtonen, P. Liljeberg, J. Plosila, “Online Reconfigurable Self-Time Links for Fault Tolerant NoC”, in Networks-on-Chip, special issue of VLSI Design, Hindawi Publishing Corporation, 2007.
[17] T. Bjerregaard, The MANGO Clockless Network-on-Chip: Concepts and Implementation, PhD Thesis, Technical University of Denmark, 2006.
[18] A. Hansson, K. Goossens, A. R adulescu, “Avoiding Message-Dependent Deadlock in Network-Based Systems on Chip”, in Networks-on-Chip, special issue of VLSI Design, Hindawi Publishing Corporation, 2007.
[19] S. Murali, D. Atienza, L. Benini, G. De Micheli, “A Method for Routing Packets Across Multiple Paths in NoCs with In-Order Delivery and Fault-Tolerance Guarantees”, in Networks-on-Chip, special issue of VLSI Design, Hindawi Publishing Corporation, 2007.
[20] P. Bogdan, T. Dumitras, R. Marculescu, “Stochastic Communication: A New Paradigm for Fault-Tolerance Networks-on-Chip”, in Networks-on-Chip, special issue of VLSI Design, Hindawi Publishing Corporation, 2007.
[21] H. Naeimi and A. DeHon, “Fault Tolerant Nano-Memory with Fault Secure Encoder and Decoder” ”, Proceedings of the 2nd International Conference on Nano-Networks and Workshops, September 2007.
[22] T. Mangir, “Integrity and Integration Issues for Nano -Tube Based Interconnect Systems”, Proceedings of the 2006 International Conference on Data Mining, June 2006.
[23] N. Srivastava and K. Banerjee, “Performance Analysis of Carbon Nanotube Interconnects for VLSI Applications”, Proceeings of the 2005 IEEE/ACM International Conference on Computer-aided Design, 2005.
[24] H. Colfen and S. Mann, “Higher-Order Organization by Mesoscale Self-Assembly and Transformation of Hybrid Nanostructures”, Angew. Chem. Int. Ed. 2003, 42, 2350 – 236.
[25] S. Bush and S. Goel, “Graph Spectra of Carbon Nanotube Networks”, Proceedings of the 1st International Conference on Nano-Networks and Workshops, September 2006.
[26] S. Bush and Y. Li, “Nano-Communications: A New Field? An Exploration into a Carbon Nanotube Communication Network, GE Technical Report 2006GRC066, February 2006.
[27] B. Agrawal, N. Srivastava, F. Chong, K. Banerjee, T. Sherwood, “Nano-enhanced Architectures: Using Carbon Nanotube Interconnects in Cache Design”, Proceedings of the 4th workshop on Non-Silicon Computing (NSC-4) held in conjunction with the 2007 International Symposium on Computer Architecture (ISCA'07 workshop), San Diego, California, June 2007.
[28] J. Patwardhan, C. Dwyer and A. Lebeck, “Self-Assembled Networks: Control vs. Complexity”, Proceedings of the 1st International Conference on Nano-Networks and Workshops, September 2006.
[29] Nathan A. Weir, Dannelle P. Sierra, and James F. Jones, “A Review of Research in the Field of Nanorobotics”, System Technologies, SAND2005-6808, October, 2005, Intelligent Systems and Robotics Center, Sandia National Laboratories.
[30] M. Sitti, “Micro- and nano-scale robotics,” in *Proc. American Control Conf.*, Boston, USA, June 2004, pp. 1–8.
[31] M. Sitti, "Microscale and nanoscale robotics systems [Grand Challenges of Robotics]," *Robotics & Automation Magazine, IEEE* , vol.14, no.1, pp.53-60, March 2007.
[32] G. Alfano; D. Miorandi, "On Information Transmission Among Nanomachines," *Nano-Networks and Workshops, 2006. NanoNet '06. 1st International Conference on*, vol., no., pp.1-5, Sept. 2006.
[33] “Electromagnetically coupled interconnect system,” United States Patent 6882239, 2005.
[34] M. Chang, V. Roychowdhury, L. Zhang, H. Shin, and Y Qian, "RF/Wireless interconnect for inter-and intra-chip communications," Proc. of the IEEE, vol. 89, no. 4, pp. 456-466, Apr. 2001.
[35] Nano-Net, Third International Conference on Nano-Networks, Boston, MA, Sept 15-17, 2008, http://nanonets.org/.
[36] J. P. Patwardhan, C. L. Dwyer, A. R. Lebeck, D. J. Sorin. “NANA: A Nanoscale Active Network Architecture”, ACM Journal on Emerging Technologies in Computing Systems) Vol. 2, No. 1, Pages 1-30, January 2006.
[37] S. F. Bush and A. Kulkarni, Active Networks and Active Network Management: A Proactive Management Framework, Kluwer Academic/Plenum Publishers, New York, Boston, Dordrecht, London, Moscow, 2001, 196 pp. Hardbound, ISBN 0-306-46560-4.
[38] S. F. Bush (2007), The Handbook of Computer Networks, John Wiley & Sons, chapter Active Networking, pp. 3008.
[39] C. H. Bennett & G. Brassard, (1984). Quantum cryptography: Public key distribution and coin tossing. In Proceedings of International Conference on Computers, Systems and Signal Processing, New York.

Manuscript received February 6, 2008 (date on which paper was submitted for review). Corresponding author: S. F. Bush (e-mail: bushsf@research.ge.com; phone: 518-387-6827; fax: 518-387-4042).

An Analysis of Information Security Governance Structures: the Case of Société Générale Bank

Ifeoma Udeh and Gurpreet Dhillon
School of Business,
Virginia Commonwealth University, Richmond, VA

***Abstract*–Organizations constantly experience lapses in internal organizational controls thereby affecting the information security of the enterprise. While the problem has been widely acknowledged and sufficient advances made in addressing the issues, yet incidents of gross neglect and failure of information security governance continue to increase. In this paper we analyze the latest casualty of failed security governance, the case of the French Société Générale Bank. Our analysis suggests that a skewed technical orientation in instituting controls was to blame. We propose that a more balanced approach to designing controls needs to be adopted. Such an approach would consider a range of issues at the informal, formal and technical levels.**

I. INTRODUCTION

Information system security is an ongoing concern to businesses, regulators and users alike. The concern about information system security heightens with the advance in technology, and this is so not only because of increased reliance of individuals and businesses on information and communication technologies, but also because the attempts to manage information security have been skewed towards implementing increasingly complex technological controls [3]. Information systems security involves a formal, informal and technical dimensions [2], and a system is vulnerable to attack to the extent that either one of these dimensions is porous. Prior literatures have emphasized the importance of viewing information systems security from a socio-technical perspective. This stance does not underestimate the importance or role of technology, but in addition, it acknowledges the effect of the human factor. According to Dhillon [3], the violation of safeguards by trusted personnel of an organization is emerging as a primary reason for information security concerns.

Organizations of all kinds, government parastatals, public and private companies, and even nonprofit organizations, have each experienced some form of information security system breach. With the capitalistic nature of the environment in which these organizations operate, most of the information systems security breaches that make the popular media are those that have huge financial implications, in other words, those that are closely related to a fraud scheme. Such information system security breaches resulting from violation of safeguards is defined as a deliberate misappropriation by which individuals intend to gain dishonest advantages through the use of computer systems [4]. Not to discredit the goodness in people in general, but as Brewer [1] states fraud often simply starts as good people coming together to create business solutions to satisfy some demand set by external forces. However, as the external (and maybe internal) pressure increases, so too does the 'engineering' of the solutions, which if not checked against some established standards, may escalate out of proportion.

Information system security breaches in general and fraud-related breaches in particular, occur as a result of three main factors: opportunity, rationalization/attitude and pressure. These factors need not all be present for a fraud-related breach to occur, as the presence of either rationalization/attitude or pressure and opportunity is sufficient to motivate and affect a fraud scheme. Opportunity may be manifested in diverse ways, and the most common is the lack of effective system controls. In addition to technical aspects such as passwords and keys, system controls involve promoting the values that a business feels are positive, and monitoring employee behaviors [3]. Rationalization or attitude focuses on the normative beliefs and the personal factors of individuals. Some people may justify an inappropriate behavior on many grounds including, that everyone is doing, or that it is the only way to survive, or that it is their right considering the compensation they receive for their hard work. The third factor, pressure, relates to expectations from both internal (e.g. management) and external (e.g. shareholders, financial analysts, market etc.) forces that demand positive and maybe extraordinary outcomes such as continuous rise the stock price.

This paper is an analysis of the recent Société Générale Bank fraud. This paper analyzes the fraud case from a control perspective, and argues that employees are motivated to indulge in fraudulent practices when adequate controls are not in place or when the established controls are not being strictly implemented. The paper is organized as follows. Section II provides some background information about the Société Générale Bank fraud. Section III presents the control structures. Section IV presents the discussion and evaluation of controls, and Section V presents the conclusion.

II. THE SOCIÉTÉ GÉNÉRALE BANK CASE

In January 2008, Société Générale Bank (hereafter SGB) disclosed that it had suffered losses, equal to more than 7 billion U.S. dollar (Euro 4.9 billion), when it unwound 50 billion Euros worth of unauthorized bets that Jerome Kerviel, an employee of the bank, hid through a series of fictitious transactions. The $7 billion fraud is the biggest in the history of France. Jerome Kerviel, a 31 years old trader at SGB, is accused of breach of trust, fabricating documents and illegally accessing computers. SBG claim that Kerviel took unauthorized positions totaling Euro 50 billion on three European futures markets, which led to more than $ 7 billion (Euros 4.9bn) loss. As a result of the fraud that was detected at SGB, the shares of bank were suspended from trading after falling 4.1 percent to $115 (about 79.08 Euros). Though no evidence has been found to confirm that Kerviel had benefited personally from the fraudulent activities, he may have done so by artificially boosting his paper transactions and thus, the bonus he could claim for 2007. Kerviel was paid Euros 60,000 bonus in 2006, and in 2007, he demanded Euros 600,000, but was only paid Euros 300,000.

III. THE OPPORTUNITY

Since the late 1800s, SGB has grown to become one of the three leading French banks, and their core businesses include Corporate & Investment Banking, Retail Banking, Specialized Financial Services, Asset Management, Private Banking, Securities Services, payment Services and Suppliers. Being established and listed in a country with a communitarian view of corporate governance structures, SGB has an Audit Committee, a Compensation Committee and a Nomination Committee, and claims to implement the recommendations given in the Association Française des Entreprises Privées (Association of French Private-Sector Companies) and Mouvement des Entreprises de France (French Business Confederation) - AFEP-MEDEF, report of September 2002 on the corporate governance of listed companies. SGB has a culture that motivates their employees to achieve their goals; and the bank claims that their success over the years has been a factor of team spirit, professionalism and innovation.

With respect to trading transactions at SGB, there are controls relating to trading limit authorizations, however, Jerome Kerviel was able to circumvent these controls and post fictitious transactions totaling more than $7 billion. Jerome Kerviel was the sole architect of an elaborate fraud involving scores of fake transactions. Jerome's exposure to the banking industry and the manner in which trades are conducted made him aware of the internal controls and the lapses that existed in the internal controls at SGB. More so, Jerome has a brother Olivier Kerviel, who until recently was a former trader at BNP Paribas (the leading French Bank), but Olivier left BNP Paribas after acknowledging to an unrelated trade that caused losses to a client. As such, Jerome Kerviel had knowledge about the shortcomings that may exist in internal controls, and manipulated the system.

The management was passive and negligent about the details of the trades that were conducted by Kerviel that they later blamed for losses of more than $7 billion (4.9 billion euros). Management failed to follow up and investigate further unacceptable trade practices that they witnessed. Kerviel made speculative bets on the order of 500 million ($770 million) to 600 million euros ($925 million), at the workstation of his direct superior, Eric Cordelle, and in his presence, when the maximum authorized ceiling for the desk was Euro 125million. Kerviel unwound the fake transaction positions that he posted from his superior's computer within a day or transferred them to his own computer, from where he either erased them immediately or kept them. When he maintained the fake positions for longer than a day, it was without Cordelle's knowledge. However, though Eric Cordelle initially denied knowledge of the fake transactions, and stated that the necessary software to post transactions was not even installed on his computer, he subsequently admitted that he had witnessed Kerviel taking intraday positions and making unauthorized trades on the computer of a junior trader Bu-Ly Wu, for a few hours as part of his training and confronted him.

Further, the management of SGB ignored the several warnings in 2006 that should have led them to investigate Kerviel's activities more closely. The findings released by a panel of independent SGB board members indicate the bank supervisors and the bank's compliance officers had failed to follow up on at least 75 internal alerts raised by Kerviel's activities. In particular, the findings indicate that his immediate superior was made aware on several occasions of the irregular trades but failed to take action. These findings suggest that Kerviel's actions were tolerated while he was on a winning streak. Similarly, the 27-page interim report commissioned by the three-man crisis committee, paints a picture of lax supervision in which controls appear to have been exercised in a box-ticking manner rather than through persistent inquiry. In particular, one reason the alleged fraud was not discovered in spite of concerns raised by Eurex, Europe's leading derivatives exchange, was that Kerviel's supervisor "was satisfied by the trader's explanation without verifying it". Kerviel's supervisor accepted his explanation in spite of the fact that the explanation contradicted Eurex's concerns.

On the one hand, the bank appears not to have a hotline or whistle-blower program in place, that will enable subordinates and employees in general report illicit activities that they are aware of. Kerviel claims that his assistant Thomas Mougard was aware of the fake transactions, since he had asked him to input fictitious transactions; and that Mougard carried out his instructions knowing that they concerned hiding open positions and earnings. Mougard admitted to inputting fictitious orders on

behalf of Kerviel, but stated that Kerviel had informed him that the trades were meant to cover out-of hours transactions, and that Kerviel never mentioned the multi-billion euro stakes on the stock markets. Mougard conceded that he could have entered, some trades using Kerviel's computer, and that there was the possibility that on some occasions he may have forgotten to log off when he was using Kerviel's workstation to access certain timely information. However, Mougard denied ever knowingly conducting fictitious transactions.

More so, the committee's report stated that there were also no controls on cancelled or modified trades. These were techniques allegedly employed by Kerviel in effecting and unwinding the fake trades. The report also stated that the people charged with oversight over the controls did not routinely inform superiors about anomalies even those involving significant sums of money. Though, no evidence of collusion or of self-enrichment on Kerviel's part was found, the 400% increase in his bonus was a significant change that could have been traced to the sharp change in his trading record, and that could arguably have raised suspicions. Further, although no evidence of accomplices was found, the possibility of the existence is not ruled out, especially in view of the extended internal network of personal relationships that Kerviel had, in particular with the support and control teams.

IV. PREVALENT NORMS

Like many other merchant banks, SGB rewarded their employees with bonuses, which are based on the level of profitability contributed by the employee. The bonus offered were substantial considering that SGB is one of the three leading banks in France. Hence, employees are motivated to be innovative and increase their contribution towards the company profits. More so, SGB over the years has proven to be performance-oriented, such that even in its early days, during the dark years, 1871 to 1893, when France went through a period of economic gloom marked by the failure of several banking establishments, SGB continued to grow at a moderate but steady pace. This demonstrates the bank's capacity to withstand unfavorable economic conditions. Such a drive to succeed at the corporate level has filtered down to the employees and has become a culture in the organization.

Nonetheless, there seems to be an underlying culture to shirk responsibilities even when comparable authorization exists. This evasion of responsibilities results in a lack of accountability. With the fraud at SGB and the reported lack of control, many criticisms have been made against the system and those charged with governance. The French President, Nicolas Sarkozy stated that he was baffled by bank's board loyalty to the CEO. After the discovery of the fake transactions, in late January 2007, Daniel Bouton offered his resignation but the bank's board rejected the offers partly out of fear that a change at the helm then might derail a 5.5 billion euro emergency share sale aimed at shoring up reserves that had been depleted as a result of the scandal. The French President stated that it is not normal for a president of a company to experience a disaster of this magnitude and not face any consequences. Neither the CEO nor any member of management accepted responsibility whether directly or indirectly for the fraud. The CEO Daniel Bouton implicitly assumes responsibility for the success of the bank by demanding to be paid seven million euros a year, however, the CEO failed to assume a similar responsibility when there was a problem.

Further, as commented by Walt Lukken, the top U.S. futures markets regulator, industry participants can become complacent about their own practices, even when they may be following all of their regulator's rules. SGB though it had implemented the corporate governance requirements, was unworried about the daily practices of the employees. Such practices as using others computers to post trades and authorizing trades that were above a trading desk's limit are just a few of the practices that should have signaled to management that controls needed to be evaluated. Also, the fact that such practices were what resulted in the $7 billion fraud, according to Lukken, merely reinforces the basic view that smart business practice extends beyond "checklist regulation. Further, such practices, and the fact that they are allowed to go unchecked in the bank raises concerns about the tone of ethics prevalent in the bank.

V. EXTERNAL VARIABLES

In France, corporations are seen as social organizations having a legal status accorded by the Society, and are expected to act in the best interest of the society, meeting social responsibilities, and not just the demands of the shareholders. France has detailed regulations specifying the social responsibilities of firms towards the communities and most large publicly traded firms are state-controlled. Though SGB is a publicly traded firm that responds to the demands of the investors, being a major organization in France, the government takes interest in the bank, and is concerned about its success. With the discovery of the over $7 billion fraud and the subsequent decline in the shares of SGB, the bank may have seemed a good "prey" for a merger, but the French President Sarkozy is determined to keep SGB as a French institution for both economic and political reasons, in view of the speculations about potential mergers with other non-French institutions, such as Unicredit and Intesa Sanpaolo, of Italy, and Banco Santander, of Spain; and the fact that a merger between a French bank such as BNP Paribas and SGB, each with a sizable investment banking businesses, would probably result in significant layoffs.

To remain viable and meet the demands of the shareholders, in view of the fraud discovery, SGB had a 5.5 billion euro share sale. This right issue was aimed at helping SGB recover from approximately Euros 4.9 billion

of losses due to the unauthorized trading activities of Jerome Kerviel, and a further Euros 2.6 billion write-down on sub-prime exposure in the American mortgage securities market. SGB initiated and completed the rights issue of 5.5 billion euros to bolster its capital. The share sale was a success, attracting
demand from investors for nearly twice the number of shares offered. Though the rights issue was heavily oversubscribed, much of the success is due to the sub-prime crisis, which has forced potential predators to withdraw their advances. Banks such as BNP Paribas pulled out of a merger deal with SGB because of the financial turbulence, unpredictable share movements and the fear of hidden losses that are high possibilities with respect to SGB current situation. Also, Credit Agricole, the French bank that had officially signaled interest in SGB did not pursue further the merger.

Pressures from across the globe continue to worsen the situation at SGB. The sub-prime mortgage market crisis is ongoing and recently, Cohen Milstein Hausfeld & Toll, a New York law firm that specializes in class action cases, filed a lawsuit against SGB in a federal court in New York, alleging that the bank misled American investors by failing to inform them about its sub-prime mortgage market exposure, Kerviel's unauthorized dealings, and for failing to act on information provided to them about Kerviel's trades.

VI. CONTROL STRUCTURES

Quite a lot has been documented in prior literature about internal control especially since the recent accounting scandals involving Enron, WorldCom, and Tyco, and the enactment of Sarbanes-Oxley Act of 2002. The latter resulted in landmark changes in the field of accounting in general and in public accounting, in particular. Internal control, in the accounting community, is typically defined rather narrowly as a tool to promote reliable financial reporting, and in view of that public companies that are subject to the U.S. Sarbanes Oxley Act of 2002 are encouraged to adopt the Committee of Sponsoring Organizations of the Treadway Commission (COSO) "Internal Control - Integrated Framework" or the Control Objectives for Information and related Technology (COBIT).

COSO Internal Control - Integrated Framework states that internal control is a process, established by an entity's board of directors, management, and other personnel, designed to provide reasonable assurance regarding the achievement of stated objectives. COSO control objectives focuses on effectiveness, efficiency of operations, reliable financial reporting, and compliance with laws and regulations. The Control Objectives for Information and related Technology (COBIT) is a set of best practices (framework) for IT management created in 1992. COBIT approaches IT control by focusing on information (not just financial information) that is needed to support business requirements and the associated IT resources and processes. As such, COBIT is useful for IT management, users, and auditors.

TABLE 1
COMPARISON OF COBIT (PLAN & ORGANIZE DIMENSION), COSO, AND IT DIMENSIONS

COBIT	COSO	IT Dimensions
Define a Strategic IT Plan and direction	Control environment	Formal
Define the Information Architecture	Control environment	Technical
Determine Technological Direction	Risk Assessment	Technical
Define the IT Processes, Organization and Relationships	Control Activities	Formal
Manage the IT Investment	Control Activities	Formal
Communicate Management Aims and Direction	Information and Communication	Formal and Informal
Manage IT Human Resources	Control Activities	Formal and Informal
Manage Quality	Control Activities	Formal and Informal
Assess and Manage IT Risks	Risk Assessment and Monitoring	Formal, Informal, and Technical
Manage Projects	Control Activities and Monitoring	Formal, Informal, and Technical

Though a one-to-one mapping of the five COSO control components and the four COBIT objective domains may not be possible as each framework is targeted at a different audience, the intent of this paper is to evaluate the similarities between the COSO framework and the first of the four domains of the COBIT framework, with respect to the dimensions of Information Technology (Formal, Informal and Technical), and on that basis, discuss the internal and/or external control issues at SGB.

The COBIT framework consists of four domains: plan and organize, acquire and implement, deliver and support, monitor and evaluate. However, as mentioned above, the focus of this paper is on the plan and organize dimension. The Planning and Organization domain involves the use of information & technology, and the organizational and infrastructural form IT should take in order to achieve the optimal results. It includes ten high level control objectives (Table 1). The COSO Internal Control - Integrated Framework consists of five interrelated components: (1) control environment, (2) risk assessment, (3) control activities, (4) information and communication, and (5) monitoring.

The control environment sets the tone of an organization, influencing the control consciousness of its people. It is the foundation for all other components of internal control, providing discipline and structure. Risk assessment is an entity's identification and analysis of relevant risks to the achievement of its objectives. It forms a basis for determining how the risks should be managed. The

control activities are the policies and procedures that help ensure that management directives are carried out. The activities help ensure that necessary actions are taken to address risks to achievement of the entity's objectives. Information entails the identification, capturing, distribution and use at all levels of the entity, relevant, reliable and timely information in a form that supports the achievement of the objective. Communication involves providing an understanding of individual roles and responsibilities pertaining to internal control. Monitoring is a process that assesses the quality of internal control performance over time. It involves assessing the design and operation of controls on a timely basis and taking necessary corrective actions.

VII. DISCUSSION AND EVALUATION OF CONTROLS

The discussion on the SGB's alleged fraud case and the actions of Kerviel indicate an intentional breach of the information security system via the violation of safeguards. Dhillon [3] defined information system security breaches resulting from violation of safeguards as a deliberate misappropriation by which individuals intend to gain dishonest advantages through the use of the computer systems. Though from case it appears that Kerviel did not benefit directly from his fraudulent activities, but his subsequent demand for an enormous bonus may have been part of the whole scheme intended to benefit him in a manner that seemed legal.

Though it appears that SGB has designed and implemented an internal control system, which to a great extent addresses the technical dimensions, the flaw relates to the operating effectiveness of the established system. The formal and informal dimensions, but primarily the formal dimension was extensively porous.

A. Formal Dimensions

The need for organizations to design, implement, and monitor the operating effectiveness of its policies as it relates to information systems security can not be overemphasized. Formal security policies and procedures will facilitate communication and minimize ambiguity and misunderstandings with an organization. Management is responsible for the process of establishing these formal policies. It is in their responsibility to influence employees towards the good, and provide an environment for them to achieve organizational objectives. By setting the tone for the organization, management establishes a basic pattern of shared assumptions, values, and beliefs considered to be the correct way of thinking about and acting on problems and opportunities facing the organization [5].

It is as important to enforce these formal policies as it is to establish them. Pressures from within and/or from outside the Organization, including but not limited to those relating to profitability, may make employees overlook established policies, or make management relax the enforcement of the policies. The SGB case suggests that for reasons relating to the improved financial performance resulting from the fictitious trades, management failed to adhere to the multiple "red flags" that may have led to an earlier discovery of the alleged fraudulent practices of the trader Kerviel. Also, as the responsibility structure is an integrated part of the formal dimension, it becomes relevant to question if the multiple sources of the "red flags" could have pursued further their suspicion, after all they were unaware if the management level to which they reported to, was a part of the whole scheme. Further with respect to the authorized trading limits and using others computers, Kerviel's superior could have done much more than comment. At the least, he could have been suspended for some days which would have disrupted his fraudulent trading.

Similarly, the SGB case suggests that overall, the top-down and down-up communications at the bank is at best, skeletal. Information is not communicated timely and appropriately to the targeted recipients. And even when communicated, it appears the recipients do not act on the information received accordingly. From the case, the perceived culture, which seems to be the dominant culture, is one of lackadaisical attitude towards information systems security risk. The underlying reason for such an attitude may stem from an optimistic bias and illusion of control as it relates to information systems security [6].

More so, Kerviel's fraudulent activities spanned for about a year, and within this year internal control procedures were evaluated by two global auditing firms – Ernst & Young and Deloitte & Associates, and both in their combined report had no matters to report. This has implications for the quality of work performed by auditors both internal and external auditors. However, the discussion relating to the effectiveness of auditors is beyond the scope of this paper.

B. Technical Dimensions

SGB appears to have an established policy about authorized trading limits, however, these policies failed to be implemented. From a technical perspective, one would think that each trading desk would not be allowed by the system to post transactions above the set limit. This may be possible via matching the access code/password to the limit authorized and then granting or denying passage. However, since Kerviel was able to post fake transactions above the authorized limit, it indicates that there is a lack of agreement between the formal and the technical dimensions. Needless to mention, but all the three dimensions of information security systems ought to function in unison. More so, the fact that Kerviel was able to use others computers for trading purposes, indicate that either this risk has not been assessed by the organization, or

it has been assessed, but that management did not consider it important enough as to address it.

C. Informal Dimensions

From the SGB case, not much was observed with respect the informal dimension. However, it is important to point out that the employees functioning in any capacity within an organization should not allow their personal relations with their colleagues to cloud their judgment, impair their objectiveness or professional skepticism. Whether or not it was the alleged culprit's intent from the onset, his relationship with those in the support and control team may have affected their decisions even when they observed the abnormalities with respect to his activities.

VIII. CONCLUSION

This paper has presented an analysis of the breach of information systems security through a discussion of the recent fraud perpetrated by a security trader at the Société Générale Bank. The paper suggests that the implementation and operating effectiveness of controls particularly the formal controls is as important as the design and implementation of such controls, and that it is the responsibility of management to enforce both. Thus it is important that within an organization that management establishes and implements effective policies and procedures relating to internal controls and that these are communicated to employees via formal, informal and technical approaches. More so, monitoring should be a core part of the information security system. This suggests that abnormal activities should be fully investigated and not just waved by, and culprits should be appropriately reprimanded. Thus, to prevent adverse occurrences, management ought to set the tone from the top and maintain the tone set.

REFERENCES

[1] L. Brewer,, "Is there a little bit of Enron in all of us?," *The Journal for Quality and Participation*, vol. 30(1), pp.26-28, 2007.

[2] G. Dhillon, *Principles of Information Systems Security.* New Jersey: John Wiley & Sons, Inc., 2006.

[3] G. Dhillon, Gurpreet. "Violation of safeguards by trusted personnel and understanding related information security concerns," *Computers & Security*, vol. 20(2), pp. 165-172, 2001.

[4] G. Dhillon, and S. Moores, "Computer crimes: theorizing about the enemy within," *Computers & Security*, vol. 20(8), pp. 715-23, 2001.

[5] S.L. McShane, and M.A. Von Glinow, eds. *Organizational Behavior*, 2nd ed., New York, NY: McGraw Hills, 2003.

[6] H.S. Rhee, Y.U. Ryu, and C.T Kim, "I am Fine but You are Not: Optimistic Bias and Illusion of Control on Information Security," *International Conference on Information Systems*, Las Vegas, NV, 2005.

Content-sensitive, Temporally Adaptive Metadata

Brendan J. Gilbert[1], Raj Sharman[1], Manish Gupta[1], H.R. Rao[1], Shambhu Upadhyaya[1]
Kenneth P. Mortensen, Esq. [2]
[1] *The State University of New York, Buffalo*
{bg1,rsharman,mgupta3,mgmtrao,shambhu}@buffalo.edu
[2] *Privacy Office, U.S. Department of Homeland Security; U.S. Department of Justice*
kenneth.mortensen@dhs.gov

***Abstract*–Role-based access is the most commonly used method for providing access to information systems. Roles are secured through design principles such as least privilege and separation of duties. However, during emergency situations, system availability to first-responders and emergency coordinators through privilege escalation has proved to offer tremendous benefits. While need for privilege escalation had received much attention, little research and focus has been given to area of ensuring security of information after the emergency. Focus of the paper is secure return of access privilege levels to normalcy after the emergency situation and resulting risks. This paper discusses some models for managed privilege escalation, using a deterministic finite state machine as a framework to select sets of context-sensitive and temporally adaptive metadata, with environmental and temporal state transitions. The framework is demonstrated through its application to a historical scenario whose result could have been improved by having such a framework in place. Risk assessment discussions are also provided to ensure that reliable and secure roles are designed (for emergency) and secure transitions occur (during and after emergency).**

I. INTRODUCTION

The Health Insurance Portability and Accountability Act (HIPAA) protects the disclosure of Protected Health Information (PHI) by limiting access. This has resulted in databases containing PHI to enact different levels of user privilege to provide the minimum amount of information necessary for the intended use [1]. During emergency situations, however, PHI's availability to first-responders through privilege escalation saves lives, as evidenced by the disclosure of PHI to locate tuberculosis patients evacuated across the United States in the response to Hurricane Katrina [2]. The importance of the availability of PHI during duress is supplemented by the Department of Health and Human Services' development of an emergency responder electronic health record to assist with assembling medical histories to support emergency relief efforts and develop an interoperable record of PHI that can be disseminated quickly as needed [3]. A concern of privilege escalation is that once data is made available at a certain privilege, it cannot be made unavailable with assurance.

However, the impact of availability can be assessed and mitigated by creating structured systems with set access privileges. This article addresses some possible models for managed privilege escalation, using a deterministic finite state machine as a framework to select sets of context-sensitive and temporally adaptive metadata, with environmental and temporal state transitions. The framework is demonstrated through its application to a historical scenario whose result could have been improved by having such a framework in place.

II. PREVIOUS WORK

Privacy in healthcare is of a high concern to our society. HIPAA has set expectations and made breaches of this privacy more actionable. Recently this concern has been punctuated by events such as the theft of 40,000 patient records containing the names, phone numbers and social security numbers on April 11th 2008. The scope of the theft at New York-Presbyterian Hospital/Weill Cornell Medical Center in Manhattan was uncovered by a federal investigation and an internal audit, the hospital said [4]. The exposure of this information could cause considerable damage to the patients, whose information has been stolen and can be misused in multiple ways, including perpetrating financial frauds. However, that same identifying data can also be used during emergency situations to save lives. In the wake of Hurricane Katrina's devastation, the U.S. Department of Health and Human Services' Office for Civil Rights issued a memorandum affirming that the Privacy Rule "…allows patient information to be shared to assist in disaster relief efforts, and to assist patients in receiving the care they need." [4] In response to a congressional mandate in the Health Insurance Portability and Accountability Act of 1996 (HIPAA), HHS issued regulations entitled *Standards for Privacy of Individually Identifiable Health Information*. For most covered entities, compliance with these regulations, known as the Privacy Rule, was required as of April 14, 2003 [15]. For covered entities using or disclosing PHI, the Privacy Rule establishes a range of health-information privacy requirements and standards that attempt to balance individual privacy interests with the community need to use such data.

A concern unaddressed by the Privacy Rule's permissions to distribute PHI is the effect of the memory of responders. Once privileged data has been made available to a user, it cannot assuredly be made unavailable. Although access can be rescinded, the ability to recall PHI is not removed with removing access due to memory. From this, it is clear that in order to mitigate the likelihood of PHI being used for unethical purposes, even during emergencies the best practice of least-privilege access should be adhered to. The principle of least privilege has been described as important for meeting integrity objectives [16]. The principle of least privilege requires that a user be given no more privilege than necessary to perform a job. Ensuring least privilege requires identifying what the user's job is, determining the minimum set of privileges required to perform that job, and restricting the user to a domain with those privileges and nothing more. By denying to subjects transactions that are not necessary for the performance of their duties, those denied privileges couldn't be used to circumvent the organizational security policy [6].

Previous work examining context-sensitive access in the scope of security has provided a basis to shift away from the Role Based Access Control (RBAC) model due to the fact that the model does not accommodate flexibility [5]. Roles are strictly defined as well as the access rights provided to them. Users of the system fall into one or more of these roles, and therefore have the associated access rights due to membership. However, this model lacks any awareness or notion of contextual factors as a determinant of access privileges for a given role, nor a sense of a progression through states over time.

A model permitting roles based on context has augmented this, providing a Generalized Role Based Access Control (GRBAC) model. One example of GRBAC is the defining of different user roles based on day of the week. For instance, a payroll administrator in GRBAC system may be allowed to make payroll modifications only on a certain day of the week and resources may be accessible during certain hours of the day [6]. A second example given is to restrict access to services offered by a transportation company to legitimate users of their service—for example, being able to use WLAN services provided to travelers on a railroad carriage. In this case, the environmental role is restricted to users whose location moves at the same speed as a GPS locator on the train, to ensure that they are legitimate customers of the railway [7]. The GRBAC model allows for changes based on environmental context, but does not consider time or the different access privileges of roles based on different contextual triggers.

Applying environment roles that include a shift in the required escalation of privileges due to disaster response, and then de-escalate the privileges incrementally as the situation comes under control or as time passes, is absent from existing work. Particularly with information as sensitive as PHI, the ability to plan for changes to access controls is required to be able to assess the amount of risk associated with privilege escalation, required to engineer a business continuity plan [8, 9].

III. CONTEXTUALITY AND TEMPORALITY

How can least-privilege be maintained during emergency situations? A set of metadata sets, containing the environment roles' access control lists of different parts of the dataset, would yield adaptability to various contexts as well as provide a basis for an incremental, staged return to the least-privilege state over time. This can be viewed as a deterministic finite state machine, with the states representing various sets of roles that provide levels of privilege to responders. Contextual inputs cause transitions from the least-privileged state of metadata to states of progressively higher privilege, depending on the magnitude of the context. In states of escalated privilege, the passage of time or the occurrence of events to rectify the situation that requires higher privilege cause transitions to states of less privilege. This interpretation gives rise to a linear model of the state machine, as shown in Figure 1.

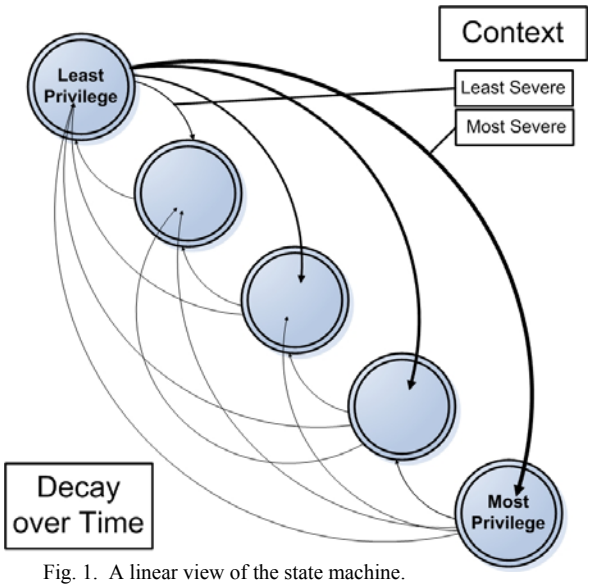


Fig. 1. A linear view of the state machine.

With these models defined, we will examine how having a contingency plan define these states and the privileges associated with each role defined by them improves the integrity of sensitive data, in the event that privilege escalation is required.

The aim of the model's defining levels of privilege is to facilitate the estimation of risk that may arise from disclosure (confidentiality) and un-availability of

information. During an emergency situation necessitating the release of PHI, it is difficult to gauge the scope of the distribution. For example, during the relief efforts for Hurricane Katrina, PHI was released to any organization providing support, such as the American Red Cross. However, since these organizations are not covered entities within the scope of HIPAA, it is likely that the sensitive PHI of some inhabitants of New Orleans are now available to many individuals from organizations that cannot be held liable for the compromised privacy of those whose PHI was distributed. The HIPAA Privacy Rule does not extend to organizations that are provided access to PHI to aid in their contribution to the relief effort [5].

Although access control cannot prevent this situation from happening again, the scope to which the released PHI is made available can be estimated through risk assessment. By planning for the amount of privileges extended to various groups of users with a state machine as shown above, the amount of risk from the scope of availability of sensitive data can be gauged and accounted for. In an effort to maximize the ability to respond to breaches of confidentiality, levels of logging can also be specified to mitigate the risk associated with the increased availability of sensitive data in an escalated privilege state.

IV. A SCENARIO

Treatment for tuberculosis requires at least six months of supervised medication. When Hurricane Katrina struck New Orleans on August 29, 2005, there were 130 residents undergoing this treatment. Ensuring these patients remained on regimen and had an adequate supply of medication took a high priority during the evacuation, to prevent epidemics breaking out over the several states evacuees sought refuge in.

The CDC's controlled efforts of this situation were very successful, getting all 130 patients back into the pharmacological fold by October 13, 2005. PHI was used in order to locate the difficult to find patients, spread over states as distant as Washington and Massachusetts. HIPAA's regulations required the creation of limited arrangements with pharmacies to cross-reference prescriptions dispensed to the tuberculosis patients' information. The CDC admits, "Prearranged agreements of this type, applicable to various health-related emergencies, would have facilitated faster location of patients," as well as "standardize[d] electronic health records" and "HIPAA-compliant platforms for sharing information." [2]

V. PRACTICAL IMPLICATIONS

Risk associated with any state of metadata, governing the roles of users and the privileges granted to each role, is a function of 1) the number of people with access, 2) the amount of access provided to those people, and 3) the amount and sensitivity of PHI that is distributed. This amount of risk is mitigated by the granularity of the result sets: for example, instead of providing a user all of the emergency-contact individuals for a certain missing tuberculosis patient, the user could only see those who were living in New Orleans and therefore might be able to provide input as to their current whereabouts. Additionally, the user would not be able to see the relationship between that contact person and the individual. The restrictions placed on result sets could dynamically change based on contextual triggers modifying the needs of the database's users. The triggers would be initiated during emergencies where adaptive roles would be re-assigned to new (or additional) set of users who would need access to information (per new roles' privileges) to respond to emergency. Possible amounts of risk associated with these are documented in Table 1.

TABLE 1
GRANULARITY OF QUERY STATES, AND ASSOCIATED RISKS

	Access to PHI		Possible Effects
High Risk States	PHI in excess of required amount	Excess PHI, too small result set	Liability for identity theft Loss of public trust/goodwill
	Very little PHI, no range required	More PHI, must define search range	Mitigated chances of identity fraud
Low Risk States	Very little PHI, must define search range		Hacking authorized account
	Denial of any PHI access		Server exploits to modify privileges

Another possible mitigating factor is the role of trust, which would reduce the liability of individual users. Although it is probably safer to assess risk as a matter of worst-case scenarios and thereby discounting the influence of trust, if trust is a significant differentiating factor between users it should also be included. Providing centralized, trusted users a higher level of access than their peers may achieve the same level of operating efficiency while reducing risk based on the number of people with access. For example, providing one "trusted" supervisor, who is a full-time employee of the relief organization with higher access than his or her volunteers might enable the same level of efficiency of locating missing patients due to directive management, but the risk of fraud would be lower, due to fewer individuals having the amount of access that the supervisor does.

In addition, there should exist certain restricted states or duties for which privilege should not exist to users who

are outside the typical assignees of that privilege. These states would have no transitions that lead into them, and may have either no transition to lead out of them or an alternate progression of state transitions than normal users to ensure that they are kept separate. Possible distributions of risk and restricted states are visualized in Figure 2—note that each state in the machine would have an associated risk level (R_N) that may provide an incrementally higher amount of risk than the previous level (the case in the concentric diagram on the right) or may provide similar amounts of risk but not encompass all of the risks of the previous state (the case in the irregular diagram on the left). These amounts of risk are highly variable dependant on the system being evaluated, and are the domain of the system or database architect to consider.

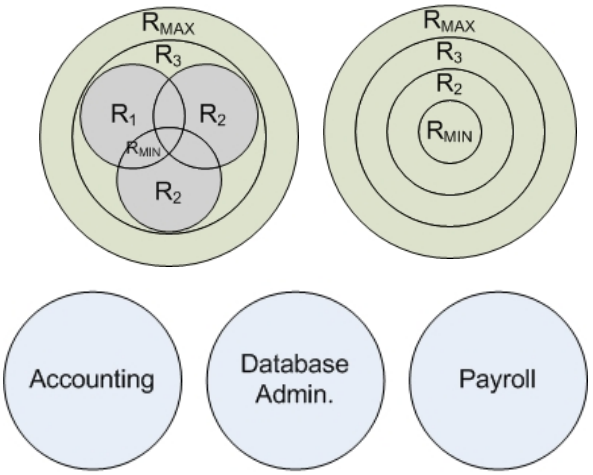


Fig. 2. Possible distributions of risk ($R_{MAGNITUDE}$) and restricted states/duties.

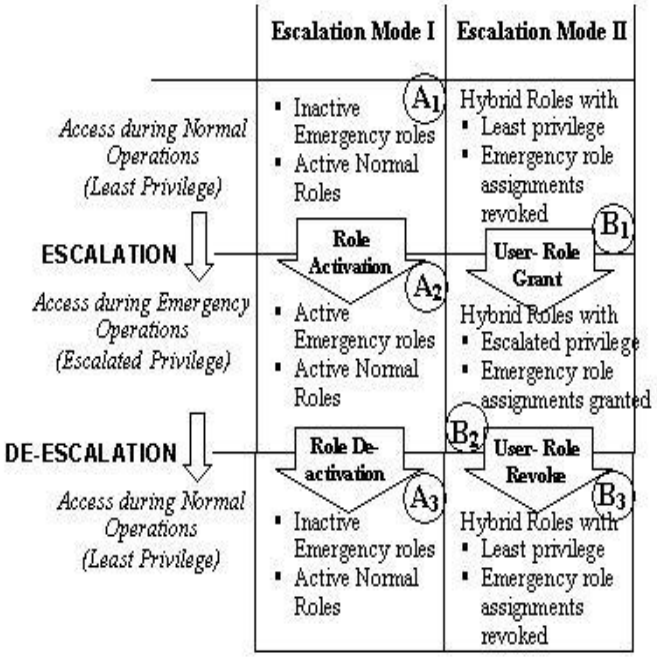
VI. APPLIED EXAMPLES

Applying this model to the after-effects of the September 11th, 2001 attacks, we would see a progression to a very high level of privilege offered to those acting on behalf of national security, with relaxation over the passage of time back to a higher default lowest-privilege level. The steps of returning to the lowest-privilege state across the top of the matrix model are in this case omitted to reflect the higher amounts of access privileges available to all users responsible for the reduced privacy in effect post-attacks. Making such a change would require that new orders be determined for risk progression, based on the new lowest-privilege state. Progressions to higher privilege states could be based on the National Threat Advisory provided by the Department of Homeland Security.

The controlled disclosure of PHI could also be used to ensure that the homes of Hospice patients are given a higher priority for power restoration during a blackout. Following an unexpected blizzard in October of 2006, a million Western New York residents lost power for at least a day, with 350,000 households losing power for the majority of a week. The full death toll for the storm included 13 people. Due to triage the Army Corps of Engineers, as part of the relief effort, supported churches and missions first due to the number of people seeking refuge there. Residents who needed electricity to power home medical equipment "were in dire straits," according to the Corps [10]. If the Corps had access to disclosed PHI, including addresses of local residents requiring power to use home medical devices—for example, a dialysis machine, or a respirator—the residents could have been prioritized in the power restoration process, lowering the number of casualties.

VII. DISCUSSIONS ON RISK MANAGEMENT

While system access structure should be adaptive to accommodate privilege escalation during emergencies, there are several risk factors that should be accounted for while the system roles and escalation workflows are architected.



Typically, there are two common methods of granting escalated system access (See Figure 3): 1) ESCALATION MODE I (ROLE – ACTIVATE/DE-ACTIVATE), which involves creating roles with higher authority, according to requirements to respond to a crisis. In this mode, roles remain dormant until a situation of emergency, and 2) Escalation Mode II (User – Grant / Revoke) where the system already has roles that have sufficient privileges to accommodate efficient response to an emergency. In this case, designated users are assigned to relevant roles to perform duties during an emergency. An automated or

defined process should be in place to initiate the assignment with least latency, while ensuring information assurance.

In either of the above-mentioned escalation modes, there are several threats that can arise:

a. How to engineer roles for system access that can be used during both - normal operations and emergency situations, while ensuring lowest risks and threats to information assurance.
b. How to make sure that escalations happen without compromising confidentiality, integrity and availability of information contained in the system.
c. What is the more suitable design between 2 escalation modes given the system states and ensuring availability of data.
d. How to incorporate accountability and audit-ability of roles usage during escalation and during emergencies.
e. How to create workflows/processes for efficient de-escalation of privileges, in either of modes, without affecting response to crises.
f. General security questions such as:
    i. What's the impact if an attacker, during emergency, can manipulate the escalation process or system itself to read the system data? What happens if access is denied to the system during emergency?

To aid in asking these kinds of pointed questions, we argue for the use of threat categories by adapting and extending the STRIDE threat model. Developed by Microsoft, STRIDE is an acronym derived from the following six threat categories: Spoofing identity (S), Tampering with data (T), Repudiation (R), Information disclosure (I), Denial of service (D) and Elevation of privilege (E) [12]. In fact, the above threat categories may not be mutually exclusive. A threat that is exploited can lead to other threats. Some threat types can interrelate. It's not uncommon for information disclosure threats to lead to spoofing threats if the user's credentials are not secured. And, of course, elevation of privilege threats are, by far, the worst threats—if someone can become an administrator or can get to the root on the target computer, every other threat category becomes a reality [14]. Conversely, spoofing threats might lead to a situation where escalation is no longer needed for an attacker to achieve his goal. For example, using SMTP spoofing, an attacker could send an e-mail purporting to be from the CEO and instructing the workforce to take a day off. To capture the various nuances of this transaction from the threat focus, and to get a better understanding of the components involved, a workflow representation is developed [13].

Similar applications have been suggested in research for developing a framework for the measurement of security levels of any EBPP [14] system to help security personnel to ensure a higher level of understanding of information assurance issues and proactively engage in elevating security measures and fraud protection in their organizations. We studied the 7 steps risk assessment framework [14] and believe that it can be adapted for managing risks for adaptive role systems. Figure 3 shows two escalation modes with system states before, during and after an emergency represented as A_1, A_2, A_3, B_1, B_2 and B_3. There are various threats, that can arise due to design of a role system for emergency roles, which should be evaluated for risk management. Proper analysis of state levels X_i will ensure that role design is secure and efficient transitions of states take place during an emergency and during restoration. The figure shows 3 states for the system with escalation mode I, where specific and exclusive roles are designed for use during emergency. These roles are only activated during emergency and de-activated after. System states, using escalation mode II, rely on assigning additional users to existing system roles.

VIII. FUTURE RESEARCH AND CONCLUSIONS

This work is part of an ongoing larger project to examine how to integrate context-sensitive metadata governing privilege escalation into continuity planning. Until now GRBAC has been viewed as a way of moderating user privileges during normal operations—using it to aid in disaster recovery planning, and to understand the risks during plan implementation is an avenue not yet explored, which should be in greater depth.

Despite the crisis of an emergency, proper emergency management should aim to provide a return to normalcy as soon as possible. The temptation to "open the floodgates" of information by relaxing lowest-privilege access to data is strong, particularly if there exists no framework in place to be able to gauge the effectiveness of progressive amounts of requirement reduction. Application of this framework provides a rational way to provide the amount of access needed to save lives, yet not too much access so as to increase the grief of survivors of an emergency through the possible fraud due to information disclosure can bring.

To efficiently implement a series of metadata that fulfill this concept, it would be best to modularize parts of the metadata. A separate relational database of different metadata states could accomplish this with a minimum of redundancy.

A possible extension of this project, especially given the modularized groups of metadata, is developing a regression analysis to forecast the amount of risk that a given state would produce. A regression analysis equation would support the dynamic creation of new

states based on a certain maximum acceptable threshold for risk, and also certain rules governing how privileges and roles should be combined to make a logical and efficient system. For example, using the regression coefficients provided from the analysis, an algorithm could dynamically determine the most optimal state to provide as much access as possible within a certain allowed maximal amount of risk, and then trigger a change into that optimal state. As the requirements change, the algorithm could again calculate the most efficient state of metadata to use and then transition into the new, dynamic state automatically.

REFERENCES

[1]. "Uses and Disclosure of Protected Health Information: General Rules." Code of Federal Regulations Title 45, Pt. 164.502(b)(1), 2006 ed.

[2]. Center for Disease Control. "Tuberculosis Control Activities After Hurricane Katrina," *Morbidity and Mortality Weekly Report,* vol. 55, pp. 332-335, 2006.

[3]. United States Department of Health and Human Services. "Emergency Responder Electronic Health Record Detailed Use Case.," Loonsk, John W. Washington, DC: U.S. Department of Health and Human Services, 2006.

[4]. United States Department of Health and Human Services, "Hurricane Katrina Bulletin: Disclosing PHI in Emergency Situations," *Department of Health and Human Services,* 2005. http://www.hhs.gov/ocr/hipaa/KATRIN AnHIPAA.pdf, 20 Dec 2006].

[5]. D. Ferraiolo, and R. Kuhn, "Role-Based Access Control." In *Proceedings of the 15th National Computer Security Conference*, October 1992.

[6]. M.J. Covington, W. Long, S. Srinivasan, A.K. Dey, M. Ahamad, and G. Abowd, "Securing Context-Aware Applications Using Environment Roles," I*n Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies,* Chantilly, VA, 2001.

[7]. B. Hulsebosch, A. Salden, and M. Bargh, "Context-Based Service Access for Train Travelers," *In Proceedings of the 2nd European Symposium on Ambient Intelligence (EUSAI),* Markopoulos et al. (Eds.), LNCS 3295, pp. 84-87, Eindhoven, The Netherlands, 2004.

[8].B.J. Dooley, "Preparing a Business Continuity Plan." *Faulkner Information Services,* 2006.

[9]. "Business Continuity Planning Booklet," *Federal Financial Institutions Examination Council,* 2003.

[10].U.S. Army Corps of Engineers, Buffalo District. Home Page. U.S. Army Corps of Engineers., 2006. http://www.lrb.usace.army.mil/

[9] M. Howard, and D. LeBlanc, "The STRIDE Threat Model". *Writing Secure Code*, 2002 ed. Microsoft Press. Chapter 2: Designing Secure Systems, pp. 38 – 60.

[13] Y.I. Song., H.R. Rao., and S. Upadhyaya, S. "Information Assurance Issues of the Workflow management Systems in E-Banking: An investigation on the modal points with high risk," University at Buffalo, Working paper, June 2003.

[14] G. Tanna, M. Gupta, H.R. Rao, and S. Upadhyaya, "Information Assurance metric development framework for electronic bill presentment and payment systems using transaction and workflow analysis," *Decision Support Systems Journal*, vol. 41(1): pp. 242-261, 2004/2005.

[15] "Clinical Research and the HIPAA Privacy Rule". Retrieved April 19, 2008 from NIH website at http://privacyruleandresearch.nih.gov/clin_research.asp

[16] "Integrity in Automated Information Systems". *National Computer Security, Center*, September 1991.

Recursive Data Mining for Author and Role Identification

Vineet Chaoji, Apirak Hoonlor, Boleslaw Szymanski
Center for Pervasive Computing and Networking, Rensselaer Polytechnic Institute,
110 Eighth Street, Troy, New York 12180
{chaojv, hoonla, szymansk}@cs.rpi.edu

***Abstract* — Like paintings and verbal dialogues, written documents exhibit the author's distinctive style and identification of the author of an anonymous document is an important and challenging task in computer security. Even more challenging is identification of a style of a group of diverse individuals acting in similar circumstances, like authors writing in certain literary period or people writing in a certain social role. The last application is important for analyzing hidden group communicating over the internet in which neither identities nor roles of the members are known. Other applications of the identification of such styles include fraud detection, author attribution and user profiling. The task of finding distinctive features of an artifact has much broader scientific implications that range from art and scriptures to network security.**

In this paper, we focus on capturing patterns in electronic documents. The approach involves discovering patterns at varying degrees of abstraction, in a hierarchical fashion. The discovered patterns capture the stylistic characteristics of either the author, or a group of authors, or even of the specific role that the author plays in relation to others. These patterns are used as features to build efficient classifiers. Due to the nature of the pattern discovery process, we call our approach *Recursive Data Mining*. The patterns discovered allow for certain degree of approximation, which is necessary for capturing non-trivial patterns on realistic datasets. Experiments on the Enron and SEA datasets, the former categorizes members into organizational roles and the latter categorizes a set of computer sessions, are conducted to substantiate our methodology. The results show that a classifier that uses the dominant patterns discovered by Recursive Data Mining performs better than the same classifier without features based on RDM patterns, in role detection and author identification.

***Index Terms*— Data mining, feature extraction, text mining, pattern discovery.**

I. INTRODUCTION

In recent years, the internet has dramatically increased its presence in our day to day activities. From online gaming and shopping to meeting prospective life partners, the online world has captured every aspect of our life. The emergence of weblogs and social networking (online community) sites (e.g., MySpace and Facebook) has enabled people to connect across countries and continents. This has not only reduced the degree of separation between people but also allowed them to collaborate with a larger audience. On the other hand, these online communities provide a safe haven for malicious activities by rogue users. Weblogs too induce online communities by allowing users to share their views and opinions to a large audience base. Consequently, they too have been seen as a propaganda media as well as a breeding place for malicious and covert activities [20]. This massive reach and impact of online media has provided a thriving ground for internet-based terrorism and espionage, where sheer size of the community and its interactions is used to hide the group existence and on-line activities [20]. Identifying such malicious individuals and hidden groups of users is an important task for curtailing cyber-crime and preserving online privacy. Nevertheless, the task is difficult given the size of such social networks and the amount of data involved. Previous efforts [3] have used the structure of the connection network of online communities to identify such groups of users. Few efforts have been directed towards characterizing groups of users based on the content generated within the online communities and weblogs. The availability of large volumes of data - messages between users, blog posts and email communication – provides the necessary impetus for this direction. This text data contains ample clues (referred to as *patterns* or *features* in the rest of the paper) for us to infer the source of the text.

The task of identifying the author of a document is commonly termed as the *author identification problem*. Even more difficult, yet important for hidden group operating over the internet or inside a legal organization is identifying the role or the relation of the sender to the receiver based on their direct emails or messages. This task is called *role identification*. Role identification can be seen as a generalization of the author identification task wherein common characteristics of a group of authors need to be identified. Role identification can help in identifying the relationship between agents within a group, which in turn can help decipher the structure of a group. This can potentially help uncover the roles of different agents in an organized terrorist activity.

In this paper, we present a general pattern extraction method, termed *Recursive Data Mining* (RDM). The basic premise behind our approach is that within any form of

communication, traces of personal style can be identified. Moreover, every person employs multiple styles based on the stature or relationship to the recipient of the communication. For instance, the manner in which a person communicates with his/her superior is quite different from the manner in which one would communicate to his/her friend. Or the manner in which a person writes to his/her parents is different from the way he/she writes to his children. The interesting question is as follows. Are individual style differences between senders in the same role so large that the individual style variations would mask their roles? For example, is a thank you note from a rude person distinguishable from a complain letter of a polite person? Hence, in this paper we want to demonstrate (through our results) that the stylistic characteristics of a certain role overpower the individualistic characteristics of a person.

To address the above mentioned challenges RDM uses an approach that extracts syntactic patterns. These patterns capture the stylistic characteristics, which are in turn used to attribute an authorship or a role to an individual. Within the machine learning community, the term feature extraction is commonly used for techniques that identify relevant features (patterns in our case) for a given application. The term feature can represent keywords for text documents or principle eigenvectors for high dimensional genetic data. Feature extraction is broadly considered to be composed of two sub-tasks – *feature construction* and *feature selection* [6]. Presence of noise in the data hinders the task of identifying meaningful signals, which in turn causes the feature construction stage to return ineffective features. Methods for selecting relevant features fall under the feature selection task. Many key applications in computer and network security have considerably utilized and benefited from feature extraction. Smart feature selection is used for both intrusion detection [2, 10] and compression of data to minimize network traffic [18].

The input data is treated as a sequence of tokens. In any form of sequence-based information, traces of personal styles can be identified by authorship analysis (see [12], [10], and [16]). The RDM framework discovers statistically significant sequence patterns from a stream of data. The approach is independent of any semantic information making it amenable for text documents in different languages, and can be applied to sequence-based data of any nature (time series data, genome data, etc.). The method also provides a certain degree of flexibility by allowing gaps in the patterns. Gaps remove the restriction of exact matches between patterns by acting as a wildcard character (see Section 4), thus enabling approximate matches. RDM for role identification has been initially described in [4] as part of a set of tools for social network analysis. In this paper, we argue that applications in the area of security and forensics have data that can be represented as a sequence of tokens. Furthermore, these applications can be formulated as either the author identification task or the role detection problem in our framework.

Some of the key technical contributions of this work are:

- The patterns formed do not have any length restriction. This allows arbitrary size patterns to be discovered. Most of the other published techniques work on a fixed size window.
- Statistical techniques, such as log likelihood ratio tests, permutation tests and score-based sequence analysis methods are used for identifying significant patterns.
- The method is hierarchical in nature. This enables us to capture patterns at various levels of abstractions. Moreover, the hierarchical nature allows us to remove noisy symbols from the stream as we move from a lower level to a higher level in the hierarchy. This ultimately leads to discovery of *long range patterns* that are separated by long noisy intermediate segments.
- The method is also able to discover approximate (similar) patterns.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 introduces the basic terminology for this work. Section 4 provides a detailed description of our methodology. The experimental results are presented in Section 5.

II. Related Work

Even though the proposed technique is useful for identifying informal groups (internet chat groups, blogger groups, etc.) in web communication, the approach is general enough to be applied to other areas such as masquerade detection [13] and intrusion detection [11]. In the area of computer intrusion, many previous efforts have relied heavily on feature extraction methods from sequential input. Schonlau et al. [15] experimented on a sequence match method on a data set based on UNIX user truncated commands for masquerade detection task. The results from [15] illustrate that a simple sequence match does not perform well. Seo and Cha [16] introduced a combination of SVM and sequence-based user commands profile on Schonlau's dataset. The results show significant improvement over previous work. Hierarchical approach has been applied on the same dataset by Szymanski et al. [19] using RDM algorithm. Instead of using sequential-base features, in [19], RDM was applied to extract statistical information regarding the sequence, such as number of distinct tokens, number of "dominant" patterns, etc. We would like to point out that the work presented in this paper is much enhanced as compared to the initial RDM framework presented in [19]. In [19], statistics over the entire data are used to build a single classifier; in contrast, this work uses patterns as features and it trains an ensemble of classifiers as compared to a single

classifier in [19]. Also, in [19], no gap is allowed in a pattern.

In other areas, previous efforts have explored the benefits of hierarchical approach for analyzing the underlying structure of text documents [14], [17]. In [14] the authors extract a hierarchical nested structure by substituting grammar for repeated occurrences of segments of tokens. Similarly, in [17] the authors present a data independent hierarchical method for inferring significant rules present in natural languages and in gene products. Our efforts differ in that we provide certain flexibility in the patterns found by allowing gaps. This enables us to work with much smaller datasets as compared to [17]. From the experiments, we also show that the hierarchical sequence model can catch certain mannerism in human language and that it is an extension of the bag-of-words approach. The dominant patterns generated might strike a resemblance with frequent closed sequence patterns, but the significance test in RDM is "smarter" than a simple frequency check. Moreover, patterns with gaps are different from both embedded and induced sequence patterns. Identifying patterns using significance tests was used extensively in biological sequence analysis [8, 9]. In recent work [1], the authors modify the frequency-based mining task, to obtain useful patterns which are in turn used for classification.

III. Preliminaries

Consider a set of sequences, denoted as $\mathcal{SEQ}$. Each sequence consists of a series of ***tokens*** from a set $\mathcal{T}$. A sequence $S \in \mathcal{SEQ}$ can be represented as $t_1, t_2, \ldots, t_n$, where $t_i \in \mathcal{T}$ and n is the length of the sequence. Based on the application, a token can represent a different entity. For instance, within text documents, a token can either represent a character or a word and a sequence S would then correspond to the whole document. For stock market data, each token could represent a pair of numeric values (price and volume) and the sequence represents the entire time series of purchases (or sales) of a certain stock. A special token, called the ***gap token***, corresponds to a blank entry and is represented by the symbol $\perp$. The gap token mimics the '.' character in regular expressions - it can be matched with any other token. A ***sequence pattern*** $\mathcal{P}$ is an ordered sequence of tokens from $\mathcal{T} \cup \{\perp\}$. Formally, $\mathcal{P}$ can be denoted as $\{s_i: s_1 \in \mathcal{T}$ AND $s_i \in \mathcal{T} \cup \{\perp\}, i = 2 \ldots p_l\}$, where i is the index of a token in the sequence and p_l is the length of the pattern. Note that the first and last tokens are never the gap token. This condition is useful for combining contiguous patterns. In the rest of the paper, the term pattern would always imply a sequence pattern and terms pattern and feature would be used interchangeably, unless stated otherwise.

Two patterns are said to have an ***exact match*** if they consist of the same sequence of tokens. Given a ***similarity function***, $sim(\mathcal{P}_1,\mathcal{P}_2)$ a similarity score is assigned to each pair of patterns. Exact matching restricts the similarity score to binary values - $sim(\mathcal{P}_1,\mathcal{P}_2) = 1$ if $\mathcal{P}_1 = \mathcal{P}_2$, 0 otherwise. The presence of a gap token in a sequence pattern relaxes the exact match constraint, allowing it to match a wider set of patterns with $sim(\mathcal{P}_1,\mathcal{P}_2) \in [0, 1]$. A match with similarity score greater than $\alpha \in (0, 1)$ is called a ***valid match***. The set $\mathcal{M}_{\mathcal{P}}$ is the ***set of valid matches*** for a pattern $\mathcal{P}$. A pattern $\mathcal{P}$ of length l and g gaps is termed as a ***(l, g)-pattern***. If $\mathcal{P}$ has a match at index i in sequence S, then it belongs to the set of patterns ***S_i(l,g)-patterns***. The set of patterns $S_i(l)$, given by the expression $\bigcup_{g=0}^{max\_gap} S_i(l, g)$ represents all patterns of length l starting at index i in S. *max_gap*, as the name indicates, is the maximum number of gaps allowed in a pattern.

IV. Recursive Data Mining

Recursive Data Mining (RDM) is an approach for discovering features from sequences of tokens. Given a set of sequences as input, the algorithm captures statistically significant patterns from the initial sequences. The patterns obtained are assigned new tokens.

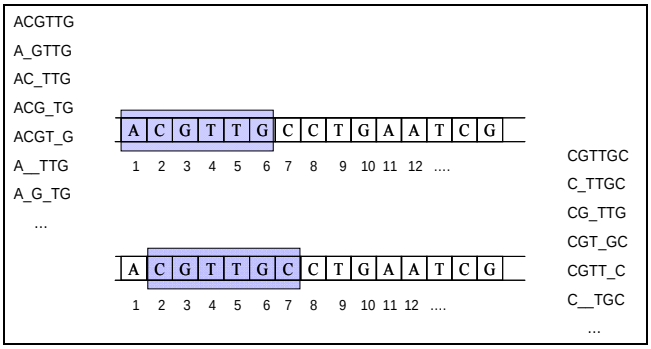


Fig 1. Pattern Generation Step. (l_w = 6,max gap = 2)

The initial sequences are re-written by collapsing each sequence pattern to its newly assigned token, while retaining the rest of the tokens. The algorithm now operates on the re-written sequences and continues to iterate through the pattern generation and sequence re-writing steps until either the sequences cannot be re-written further or a predefined number of iterations is reached. Each generation of sequences in the above process is termed a level, with the initial set of sequences called *level-0 sequences*. The patterns obtained at each level form a set of features. The term "recursive" in the title refers to this iterative step that obtains the next level by operating on the current level. In the RDM process, we claim that the recursive (hierarchical) processing of the data captures distinctive features at varying levels of abstraction. Intuitively, at lower levels the patterns obtained

are more specific; resulting in a smaller set of valid matches ($\mathcal{M}$). At higher levels, the patterns are more general, resulting is a larger $\mathcal{M}$ set. On the other hand, with increasing levels, the number of patterns found decrease monotonically.

In this section we present the details of an RDM based classifier. Like most supervised learning tools, RDM has two stages of processing – training and testing. The *training phase* starts with *pattern generation*, followed by pattern selection through the *pattern significance* step. Out of the significant patterns, the *dominant patterns* form the feature set for a level.

The overall RDM process is outlined in Algorithm 1. The input is a set of sequences $\mathcal{SEQ}_{INIT}$, which also forms the initial set of sequences $\mathcal{SEQ}_0$ in the iterative procedure. The set of sequences for level $(i+1)$ are generated from the sequences in level i and the set of dominant patterns $\mathcal{D}$. $\mathcal{P}_{ALL}$ and $\mathcal{P}_{SIG}$ represent the sets of all and significant patterns respectively. Dominant patterns (denoted by $\mathcal{D}$) for a level are obtained from the *get_domi_patterns* method. The union of dominant patterns at each level is collected in $\mathcal{L}$.

Algorithm 1 Recursive Data Mining

Input: Set of sequences $\mathcal{SEQ}_{INIT}$
Output: Sets of patterns (features) $\mathcal{L}$, one for each level
1: $\mathcal{L} = \{\}$
2: $i = 0$
3: **repeat**
4: **if** $i==0$ **then**
5: $\mathcal{SEQ}_i = \mathcal{SEQ}_{INIT}$ // Level 0
6: **else**
7: $\mathcal{SEQ}_i$ = *make_next_level*($\mathcal{SEQ}_{i-1}$, $\mathcal{D}$) // Level i
8: **end**
9: $\mathcal{P}_{ALL}$ = *pattern_generation*($\mathcal{SEQ}_i$)
10: $\mathcal{P}_{SIG}$ = *sig_patterns*($\mathcal{SEQ}_i$, $\mathcal{P}_{ALL}$)
11: $\mathcal{D}$ = *get_domi_patterns*($\mathcal{SEQ}_i$, $\mathcal{P}_{SIG}$)
12: $\mathcal{L} = \mathcal{L} \cup \mathcal{D}$
13: i++
14: **until** $\mathcal{D} == \emptyset$ **or** $i ==$ *max_level*
15: **return** $\mathcal{L}$

A. Pattern Generation

Each input instance is converted into a sequence of tokens. The initial sequence of tokens, *level-0 sequence*, will be referred to as S_0. A sliding window of length l_w moves over S_v ($v = 0$ initially). At each position p of the window, all possible (l_w,*max_ gap*)–*sequence patterns* are generated. The number of patterns generated equals the number of combinations of tokens covered by the window along with the gap token. Since we do not allow a gap for token s_1 and $s_{max_gap}$of a pattern, the number of patterns generated at each window position is given by $\sum_{g=0}^{\max gap} \binom{l_w - 2}{g}$. So, for a l_w = 5, with max gap = 2, seven patterns are generated. A bounded hash keeps count of the number of occurrences of each pattern at level v, as the sliding window moves over S_v. This forms the first pass over sequence S_v. Figure 1, shows the patterns generated at position 1 and 2 of the sequence.

B. Pattern Significance

The number of (l_w, max_gap)-patterns uncovered in the sequences is generally large. Many of those patterns are either very specific to a certain sequence or insignificant because they contain commonly occurring tokens. In either case, they are ineffective in capturing any stylistic attributes while adding to the computation cost of the algorithm. The "usefulness" of a pattern is computed with a statistical significance test. Patterns that are deemed insignificant are eliminated from further consideration. For a set of sequences $\mathcal{SEQ}$, let the unique tokens in the entire set be denoted by $\mathcal{T}$. The frequency of a token t_i in $\mathcal{SEQ}$ is denoted by f_{t_i}. So the probability of token t_i over $\mathcal{SEQ}$ is $\frac{f_{t_i}}{\sum_{j=1}^{|\mathcal{T}|} f_{t_j}}$. For a pattern $\mathcal{P}$ of length l_w, the probability of tokens in the pattern can be represented as (p_{t1} , p_{t2} ,… , p_{tlw}). Note that gaps are considered as special tokens. The probability of pattern $\mathcal{P}$ is thus given by the expression

$$\mathbf{P}(\mathcal{P}) = \mathbf{P}(RV_1=t_1, RV_2=t_2, \cdots, RV_{lw}=t_{lw}) \quad (1)$$
$$= p(t_1)p(t_2|t_1)\cdots p(t_{lw}|t_1,\cdots t_{lw-1})$$

where RV_i is a random variable for the token t_i. Assuming that the words are independent of each other (this assumption is just for the purpose of measuring pattern significance, because if they are not, we will eventually catch their relationship at the higher level of RDM abstraction), just the marginal probabilities for the words need to be computed resulting in

$$\mathrm{P}(\mathcal{P}) = \prod_{i=1}^{l_w} p_{l_i} \quad (2)$$

The probability of a gap token is ε, which is a user defined parameter (see Section Dominant Patters for details). The probability of occurrence of $\mathcal{P}$ under the random and independent assumption is given by

$$\mathbf{P}_R(\mathcal{P}) = \mathbf{P}(RV_1=t_1, RV_2=t_2, \cdots, RV_{lw}=t_{lw}) \quad (3)$$

Since each token is assumed to be equally likely under the random assumption, the above expression simplifies to $\mathbf{P}_R(\mathcal{P})$

$$= \left(\frac{1}{|\mathcal{T}|}\right)^{l_w}.$$

The ratio $\mathbf{P}_R(\mathcal{P})/\mathbf{P}(\mathcal{P})$ is used to determine significance of the pattern. If the above ratio is smaller than 1, then the pattern is considered to be significant, otherwise it is considered insignificant. The ratio indicates the likelihood of the pattern to occur under the random model as compared to its occurrence under the unknown observed distribution. This is similar in essence to the log-likelihood ratio test, with null hypothesis (H_0), that the observed distribution is similar to the random distribution. The alternate hypothesis H_1 states otherwise. The log-likelihood ratio is given by the expression

$$\mathbf{LRT} = -2\log_e(\mathcal{L}_R(\theta)/\mathcal{L}_O(\theta)) \quad (4)$$

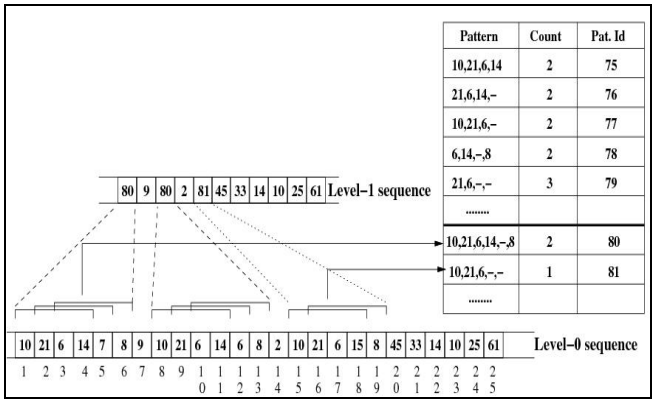


Fig. 2. Sequence Re-writing Step

where $\mathcal{L}_R(\theta)$ is the likelihood function under the random model and $\mathcal{L}_O(\theta)$ is the likelihood for the observed distribution. H_0 is a special case of H_1, since it has fewer parameters (captured by θ) as compared to the more general alternate hypothesis. Applying the significance test to the set of patterns $\mathcal{P}_{ALL}$ gives us a smaller set of significant patterns, $\mathcal{P}_{SIG}$. In practice, computational cost of the pattern generation step can be reduced by checking whether a sequence of tokens in the current window have the ratio of $\mathbf{P}_R(\mathcal{P})/\mathbf{P}(\mathcal{P})$ greater than 1 or not. If not, then we can conclude that no pattern generated from this current window can be significant. Hence, we can reduce the computation cost for generating all possible patterns from this sliding window.

Other significance tests for sequence patterns have been proposed. Permutation tests [5] provide a simple approach for comparing the observed occurrences of a pattern with the number of likely occurrences over a random sequence. The practical application of this method requires generating a large number of random permutations of the input sequence and computing the statistics on the random permutations. If the input sequence is long, this operation can be computationally very expensive. Karlin et al. [8], [9] have proposed many significance tests for identifying relevant regions in protein sequences. Their approach relies of assigning scores to the tokens such that the sum of the expected scores for all the tokens is negative. Such conditions are easier to find for biological sequences as compared to text documents.

C. Dominant Patterns

After the significant patterns at level v are determined, a second pass is made over the sequence of tokens S_v. At each position in the sequence, the tokens in the significant patterns are matched against the tokens in the sequence. The matching score is defined as the conditional probability of a match given two symbols, i.e., if $\mathcal{P}[i]$ and $S_v[j]$ are the same then the conditional probability of a match is 1. On the other hand, if $\mathcal{P}[i] = \perp$ then the conditional probability is ε. The matching score is based on the following rules

$$score(\mathcal{P}[i], S_v[j]) = \begin{cases} 1 & \text{if } \mathcal{P}[i] = S_v[j] \\ \varepsilon & \text{if } \mathcal{P}[i] = \perp, \varepsilon < 1, \\ 0 & \text{if } \mathcal{P}[i] \neq S_v[j] \end{cases}$$

where $\mathcal{P}[i]$ is the i^{th} token of the pattern and j is an index over sequence S. ε is intended to capture the notion that a $\perp$ symbol is not as good as an exact match but much better than a mismatch. The value of ε is a user defined parameter and it is set to be greater than 0.5 in our experiments to favor a match with the gap token. The total score for a pattern, starting at index j in S, is given by $score(\mathcal{P}, S_v[j]) = \sum_{i=1}^{p_l} score(\mathcal{P}[i], S_v[j+i])$. The pattern that has the highest score starting at location j in the input sequence is termed as the ***dominant pattern*** starting at position j. In other words, this is a pattern x defined by the expression $\text{argmax}_{x \in S_v}$ score(x, $S_v[j]$). The term dominant pattern is coined from the fact that this pattern dominates over all other significant patterns for this position in the sequence. Two dominant patterns that are placed in tandem can be merged to form longer dominant patterns. The merging process is continued till no further dominant patterns can be merged. An example of the merging process is shown in Figure 2. A new token is assigned to each dominant pattern. During this second pass over the sequence at level v, the sequence for level v+1 is formed. The sequence corresponding to a dominant pattern is replaced by the new token for this dominant pattern. When a dominant pattern is not found at position j, the original token is copied from sequence S_v to the new sequence S_{v+1}. Figure 2 illustrates this step.

As the RDM algorithm generates subsequent levels, certain tokens get carried over from lower levels without participating in any dominant patterns at higher levels. Such tokens are termed "noisy" for the following reasons. First, they do not contribute to any patterns at these levels. Second, they obstruct the discovery of patterns that are separated by a long sequence of noisy tokens. Patterns separated by noisy tokens are called long range patterns. An example of a long range pattern is show in Figure 3. These long range patterns can be captured only if the noisy tokens lying in between them can be collapsed. As a result, at each level, we collapse contiguous sequence of tokens that have not resulted in new dominant patterns for the last *k* levels, into a special noise token. Figure 3 illustrates the process of collapsing noisy tokens into a single special noise token ***N***. Once the noisy tokens are collapsed, distant tokens may now fall within the same window, leading to more patterns being discovered at higher levels.

The set of dominant patterns $\mathcal{D}_v$ for level *v* form the features for this level. This iterative process of deriving level *v+1* sequence from level *v* sequence is carried on till no further dominant patterns are found or *v+1* has reached a user predefined maximum value. The sets of features extracted are utilized by an ensemble of classifiers.

D. Training Phase

The training phase involves using dominant patterns generated at each level to construct an ensemble of classifiers (C_1, C_2, …, $C_{\max\_level}$), one for each level. The dominant patterns used as features reflect the most relevant patterns, ignoring the highly frequent and infrequent patterns (upper and lower cut–offs). The upper and lower cut–offs are intended to prevent the use of insignificant patterns as features. The classifiers can be created from any machine learning method, such as Naïve Bayes or Support Vector Machine. Given a set of training sequences $\mathcal{SEQ}_{tr}$, along with the labels r_1, r_2, …, r_m of all possible classes, dominant patterns are generated for each sequence starting at level 0 up to level *max_level*. The union of all tokens in $\mathcal{T}$ and key dominant patterns at level *v* across all sequences in $\mathcal{SEQ}_{tr}$ forms the set of feature for classifier C_v. For the ensemble of classifiers, the final posterior class probability is the weighted sum of the class probabilities of individual classifiers. Each classifier is assigned a weight that reflects the confidence of the classifier. In order to determine this confidence value, the set $\mathcal{SEQ}_{tr}$ is further split into a training set $\mathcal{SEQ}_{new}$ and a tuning set. Each classifier in the ensemble trains its model based on $\mathcal{SEQ}_{new}$. The accuracy of the classifier on the tuning set determines the confidence of classifier C_i (Eq. 5, Table 1)

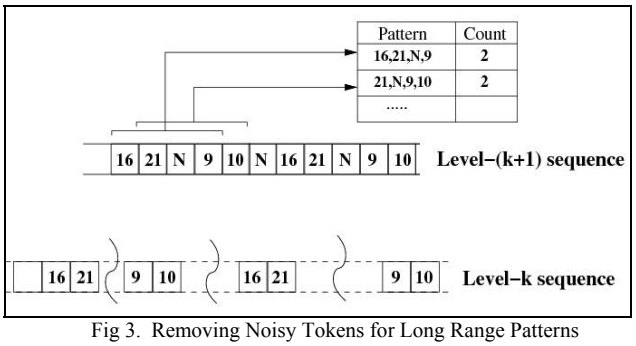


Fig 3. Removing Noisy Tokens for Long Range Patterns

$$conf(C_i) = \frac{accuracy(C_i)}{\sum_{j=1}^{\max\_levels} accuracy(C_j)} \quad (5)$$

TABLE I
COMPARISON OF CLASSIFIERS FOR AUTHOR IDENTIFICATION TASK

Classifiers	Accuracy Rate (%)
Naïve Bayes	91.5
SVM	98.4
RDM [19]	94
RDM with Naïve Bayes	91.5
RDM with SVM	98.5

E. Testing Phase

After the training phase discovers features from the training data, the testing phase finds occurrences of those features in the test data. The testing phase as such follows the training phase in terms of the level by level operating strategy. If a dominant pattern X was discovered at level-Y during the training phase, then it can be only applied to level-Y in the testing phase. Initially, the frequencies of tokens and level-0 dominant patterns are counted over the level-0 test sequence. This vector of frequencies forms the feature vector at level-0. Once the feature vector for level-0 is obtained, the next level sequence is generated. This is achieved by substituting the token of the best matching pattern at every position in the level-0 test sequence. Note that if the best match is less than a user specified threshold then the token at level-0 is carried over to level-1. Now the occurrences of the dominant patterns at level-1 are counted over level-1 test sequence. This process continues till all levels of dominant patterns are exhausted. Each classifier in the ensemble classifies the test data and the final probability is assigned based on the following weighing scheme

$$P(C \mid x) = \sum_{i=1}^{\max\_levels} conf(C_i) \times P_{C_i}(C \mid x) \quad (6)$$

where x is a test sequence and $\mathbf{P}_{Ci}(C \mid x)$ is the posterior probability assigned by classifier C_i.

V. Experiment and Results

We applied RDM to two tasks: (1) author identification, and (2) role identification. For the author identification task, RDM was applied on the SEA dataset [15]. For role identification task, RDM is applied to the Enron dataset. The detailed description of experimental setup and results for each task are presented in Section V-A and V-B respectively.

A. Author Identification Task

We illustrate that RDM using tokens and patterns as features performs as well as Naïve Bayes, Support Vector Machines, and RDM using statistical information as features [19]. We use SEA dataset [15] for this task.

1) Data Preparation and Experimental Setup

The SEA dataset consists of 50 files, where each file corresponds to one user. Each file contains 15000 commands. The first 5000 commands of each file do not contain any masqueraders, and is considered as training data. The other 10000 commands are seeded with masquerading users. For this experiment, we consider only the training data portion of the file. We split training data of each user into two sets: 2500 commands for training set and other 2500 commands for validation set. The sequences of 2500 commands are partitioned into blocks, where each block contains consecutive 100 commands. For each experiment, the input consists of blocks of 100 commands from each user. To draw analogy with the author identification task, each command corresponds to a token and each block of 100 commands represents a sequence.

TABLE II
Dataset for Role Identification

	Training Set	Testing Set	Total	# Sent folders
CEO	1010	250	1260	3
Manager	1403	349	1752	4
Trader	654	162	816	4
VP	1316	327	1643	4
Total	4383	1088	5471	15

2) Parameter Estimation

The RDM algorithm requires a few parameters to be estimated for the classification model. The parameters to be selected include 1) size of the window, 2) maximum number of gaps allowed in the window, 3) weights assigned to the classifier at each level, 4) upper and lower cut– offs threshold for key dominant patterns. A greedy search over the parameter space is conducted to determine the best set of parameters. In order to access the current parameter values, the training set is further split into two parts. A classifier is trained on the larger part, and tuned on the smaller part (called the tuning set).

3) Results

We compare four classifiers – Naive Bayes, RDM using statistical information as features, SVM, and RDM with dominant patterns as features. For SVM, we ignored the most frequent and the least frequent tokens, and used the rest of tokens as features. For our version of RDM, we applied RDM with both Naïve Bayes and SVM in an ensemble of classifiers. We used *SVMLight* as the SVM implementation. The performance of each classifier is presented in Table I. All classifiers achieve higher than 90% accuracy rate, indicating that this particular classification task is fairly easy. RDM in an ensemble setting does not show any significant improvement over Naïve Bayes and SVM classifiers. On further investigation it was evident that RDM overfits the data, i.e., the training accuracy increases while the testing accuracy decreases. Both SVM and ensemble based RDM with SVM outperform the previous version of RDM. Experiments on the SEA dataset were performed to ensure that RDM is effective for such a setting as compared to the previous RDM version [19]. The superiority of RDM can be seen for more complex tasks, in the following section.

B. Role Identification Task

We show that RDM performs better as compared to Naïve Bayes, Support Vector Machine and Predictive Association Rule based (CPAR [21]) classifiers. CPAR combines the advantages of associative and traditional rule-based classifiers. Support Vector Machines based classifiers have been shown [7] to perform well for text classification tasks. RDM does not use any semantic tools (part-of-speech tagging or synonym groups) in order to extract patterns that later serve as features for the classifiers. As a result, we compare with other techniques that do not utilize domain or semantic knowledge either. A brief introduction to the Enron dataset is provided before the discussion on the experimental setup.

1) Data Preparation and Experimental Setup

Experiments are performed on the March 2, 2004 version of Enron dataset, distributed by William Cohen, http://www.cs.cmu.edu/~enron/. The dataset was cleaned to eliminate attachments, quoted text and tables from the body of the email messages and header fields from the email. No effort was made to correct spelling errors or to expand abbreviations in order to reduce noise in the data. Stemming was performed on the dataset.

For the purpose of identifying roles, employees were

partitioned into groups based on their organizational role in Enron, as suggested in [22]. Only the roles CEO, Manager, Trader and Vice-president were used in our experiments due to a large number of employees designated with these roles. Since we are concerned with identifying roles based on the sent messages we only deal with messages in the Sent folder of each participant. For each of these roles, the emails are divided into two sets as summarized in Table II. Finally, each stemmed word in an email is considered a token, and each email represents one sequence.

2) Results

Binary and multi-class classification: We compare the four classifiers – Naïve Bayes, RDM, SVM and CPAR – under two classification settings. RDM was used both with Naïve Bayes and SVM in an ensemble of classifiers. The previous version of RDM [19] was dropped out of this experiment because the learned model always predicted every instance as the major class. In the binary classification setting, given a test message *m*, the task is to answer "*Is message m sent by a person with role r*"? where $r \in R = \{CEO, Manager, Trader, Vice\text{-}president\}$.

The training set is divided such that all messages belonging to role *r* form the positive class and all messages belonging to R\r[1] form the negative class. The performance for the four classifiers is shown in Figure 4, where the values of (*1 – F-measure*) are presented to highlight the differences in performances. Note that a smaller value of *(1 – F-measure)* indicates a better classifier. In terms of the F-measure, RDM performs better than NB and SVM. RDM outperforms CPAR under all settings.

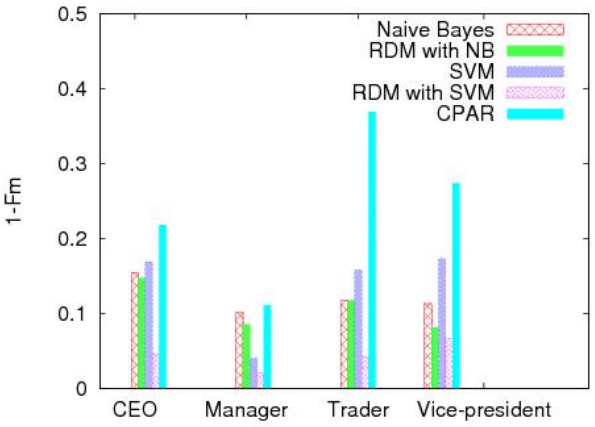


Fig 4. Binary Classification: (1 - F-score)

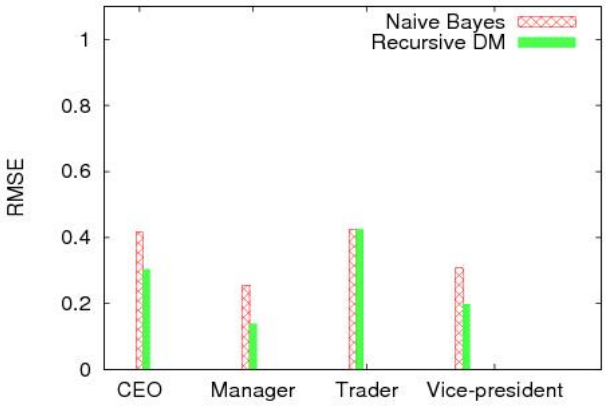


Fig 5. Binary Classification – RMSE Comparison

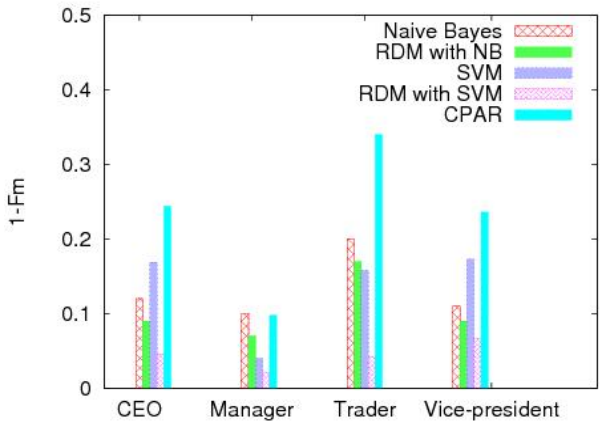


Fig 6. Multi-class Classification: (1 – F-score)

To further analyze the results, we computed the RMSE (Root Mean Square Error) for NB and RDM. The RMSE is computed using the expression

$$RMSE(SEQ_{test}) = \sqrt{\frac{\sum_{i=1}^{|SEQ_{test}|}(1 - P(r \mid SEQ_{test}^{i}))^2}{|SEQ_{test}|}}.$$

SEQ_{test}^{i} is the i^{th} document in the test set and $r = \text{argmax}_c$ **P**(c | SEQ_{test}^{i}). Since the decision function value from *SVMLight* could not be converted to an error term, the plot in Figure 5 does not show comparison with SVM. Similarly, CPAR does not provide any comparable measure. The lower the RMSE value, the more confident the classifier is in its prediction. Figure 5 shows that RDM is more confident in its predictions even when the F-scores for RDM and NB might be very close for a certain role.

[1] A\B implies all elements of set A after removing elements of set B

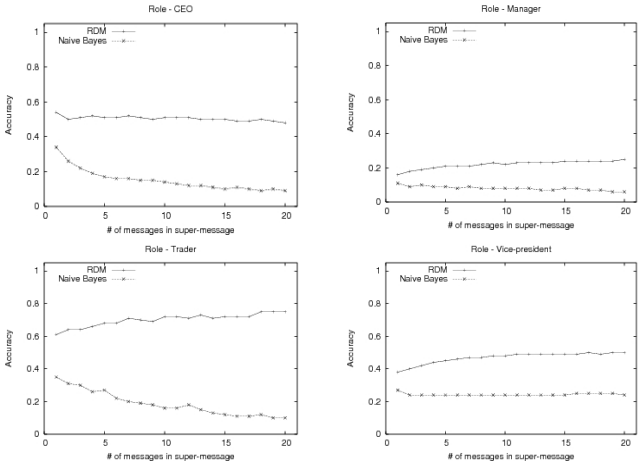


Fig 7. Classification Probability over Unseen Message Folder

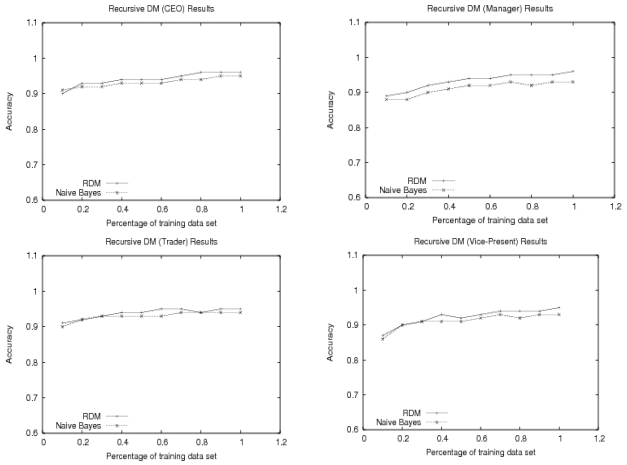


Fig 8. Effect of Changing Training Data Size

The second set of results compares the performance under the multi-class classification setting, wherein the task is to answer "*Which is the most likely role, out of roles R_1, . . . ,R_n, for sender of message m*?" For NB and RDM, the training data is split into 4 groups and probabilities computed for each of the roles. For SVM, four sets of datasets are generated, one each for role (r, $\mathcal{R}\backslash r$) pairs. The comparison for the classifiers is shown in Figure 6. RDM outperforms the other classifiers convincingly.

Paired t-test: To further investigate the results obtained for the multi-class scenario, we performed the paired t-test for statistical significance. The paired t-test provides a hypothesis test of the difference between population means for a pair of random samples whose differences are approximately normally distributed. Note that a pair of samples, each of which may not be from a normal distribution, often yields differences that are normally distributed. The null hypothesis H_0 states that the difference in performance between RDM and the other methods is not significant. In other words, H_0 states that the two methods perform equally well. The alternate hypothesis, states otherwise.

A 20-fold cross validation was performed on the data. The accuracy results obtained therein are used for the t-test, where SVM and CPAR are compared against RDM with SVM, and NB is compared against RDM with NB. The results are shown in Table III. Based on the p-value in Table III we reject the null hypothesis, indicating a definite improvement with RDM. The confidence interval for the mean difference shows that the improvement lies between 1.8% and 3% as compared to NB whereas as compared to SVM (and CPAR) it lies between 8% and 10%.

Accuracy on unseen user: For the final test we divide each role into two parts based on the users. For instance, the folders of *Jeff Skillings*, *David Delainey* and *John Lavorato* form the CEO group[2]. The first part, namely training set, contains messages from *John Lavorato*, *David Delainey* while messages from *Jeff Skillings* form the second part (test set). An RDM based classifier is trained using messages in the first part and tested on messages in the second part. In this experiment we analyze the performance of the classifier for a member whose messages are not in the training set. The results for different roles are shown in Figure 7. The test set size is gradually increased and the accuracy is noted. Notice that for the roles Manager, Trader and Vice-president the accuracy increases with larger number of message. For the CEO role, there is a marginal decline in the accuracy on increasing the number of test messages. On examining the messages for the CEO, we observed that most of the messages were written by a secretary. This explains the poor performance for the CEO role.

C. *Effect of Parameter Changes*

In this section, we take a quick look at the effects of varying certain parameters within RDM using the Enron data set. Figure 8, shows the variation in accuracy with increasing training set size. The training set for each of the roles is increased in steps of 10% of the total training set size. From these results we observe that RDM consistently performs as good as or better than NB. Moreover, it shows that both classifiers are quite robust and attain a fairly high accuracy even for smaller training set sizes.

Figure 9a captures the effect of varying window size on overall accuracy of the multi-class setting. The maximum number of gaps is set to 1. The time taken to build the classifier is shown in Figure 9b. From Figure 9a we see that

TABLE III
RESULTS OF PAIRED T-TEST

Classifier Pair	Mean Difference	Std. Dev. of (đ)	t-statistic (df=19)	p-value	95% confidence interval
NB vs RDM with NB	0.02393	0.002525	9.48	1.23E-08	(0.0186 – 0.0292)
SVM vs RDM with SVM	0.08927	0.00434	20.55	1.94E-14	(0.0818 – 0.0984)
CPAR vs RDM with SVM	0.09329	0.00535	17.45	3.74E-13	(0.0821 – 0.1045)

the accuracy is best for a window size of 3 and reduces as the window size is increased. This result is intuitive as larger significant patterns are captured by merging smaller significant patterns, whereas on the other hand smaller patterns cannot be captured using a large window size. Results for the runtime indicate that increasing the window size increases the runtime due to a larger number of generated candidate patterns.

VI. Conclusion And Future Work

We propose a general framework for feature extraction from a sequence of tokens. The framework is based on the idea of capturing statistically significant sequence patterns at increasing levels of generalization. These patterns act as features for an ensemble of classifiers, one at each level. The proposed method is simple and flexible, such that it can be applied to a range of applications. We applied it for capturing stylistic patterns in the SEA and Enron datasets, which are later used for identifying the authors and the organizational roles of authors respectively.

The method, in its current state, is devoid of any semantic knowledge, which can be easily incorporated to identify semantically related patterns. Techniques such as part of speech tagging and synonym dictionaries can augment our approach. Based on the success of the method on a noisy dataset, the authors believe that the method can perform better on cleaner and structured datasets such as the Reuters dataset or even on a foreign language dataset, such as Russian Blogosphere data. Moreover, the method can be applied on other application areas such as grouping gene products by their families.

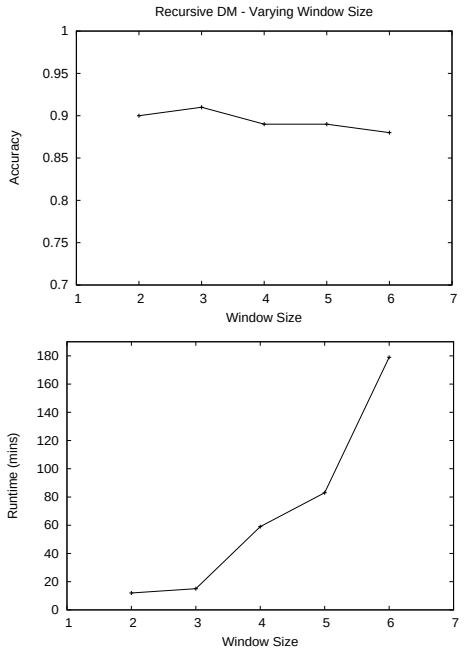


Fig 9. Accuracy (9a) and Runtime (9b) for Varying Window Size

Acknowledgment

We would like to thank the anonymous reviewers for their comments which have helped improve the quality of the paper.

References

[1] H. Cheng, X. Yan, J. Han, and C. Hsu, "Discriminative Frequent Pattern Analysis for Effective Classification," *ICDE*, 2007, pp. 716–725.

[2] S. Coull, J. Branch, B. Szymanski and et al. "Intrusion Detection: A Bioinformatics Approach," *19th Annual Computer Security Applications Conference*, Las Vegas, Nevada, 2003.

[3] N. Du and B. Wu and X. Pei and B. Wang and L. Xu, "Community detection in large-scale social networks", WebKDD/SNA-KDD Workshop, 2007.

[4] M. Goldberg, M. Hayvanovych, A. Hoonlor, S. Kelley, M. Magdon-Ismail, K. Mertsalov, B. Szymanski and W. Wallace. "Discovery, Analysis and Monitoring of Hidden Social Networks and Their Evolution," *IEEE International Conference on Technologies for Homeland Security*, Waltham, MA, (2008).

[5] P. Good, *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses*, Springer, (2000).

[6] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res*., 3 (2003), pp. 1157–1182.

[7] T. Joachims, "Text categorization with support vector machines: learning with many relevant features," *Proceedings of 10th ECML*, (1998).

[8] S. Karlin, and V. Brendel, "Chance and Statistical Significance in Protein and DNA Sequence Analysis," *Science*, 257 (1992), pp. 39–49.

[9] S. Karlin, "Statistical significance of sequence patterns in proteins," *Current Opinion Struct Biol.*, 5:3 (1995), pp.360–371(12).

[10] W. Lee, and S. Stolfo, "Data Mining Approaches for Intrusion Detection," *7th USENIX Security Symposium*, (1998), pp. 79–93.

[11] W. Lee and S. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems", ACM Transactions on Information and System Security, 2000.

[12] T. Li and M. Ogihara, "Music artist style identification by semi-supervised learning from both lyrics and content," *Proceedings of 12th Annual ACM Multimedia*, 2004, pp. 364–367.

[13] R. Maxion, "Masquerade Detection using Enriched Command Lines", International Conference on Dependable Systems and Networks", 2003.

[14] C. G. Nevill-Manning and I. H. Witten, "Identifying hierarchical structure in sequences," *Journal of Artificial Intelligence Research*, 7 (1997), pp. 67–82.

[15] M. Schonlau, W. DuMouchel, and et al. "Computer intrusion: Detecting masqueraders," *Statistical Science*, 16(1), (2001), pp. 58-74.

[16] J. Seo, and S. Cha, "Masquerade Detection based on SVM and Sequencebased User Commands Profile," *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, Singapore, (2007), pp. 398–400.

[17] Z. Solan, D. Horn, E. Ruppin and S. Edelman, "Unsupervised learning of natural languages," *Proc Natl Acad Sci U S A*, 102:33 (2005), pp. 11629– 11634.

[18] K. Sripanidkulchai, B. Maggs, and H. Zhang, Efficient content location using interest-based locality in peer-to-peer systems, In Proc. of IEEE INFOCOM, (2003).

[19] B. Szymanski and Y. Zhang, "Recursive Data Mining for Masquerade Detection and Author Identification," *Proc. 5th IEEE SMC IA Workshop*, 2004, pp. 424–431.

[20] C. Yang and T. Ng, "Terrorism and Crime Related Weblog Social Network: Link, Content Analysis and Information Visualization", IEEE International Conference on Intelligence and Security Informatics, 2007.

[21] X. Yin, and J. Han, "CPAR: Classification based on Predictive Association Rules," *SDM* 2003.

[22] http://isi.edu/~adibi/Enron/Enron Employee Status.xls

Enhancing the Non-Repudiation Properties of EMV Payment Cards

David J. Boyd
Information Security Group, Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, United Kingdom
D.Boyd@rhul.ac.uk

Abstract- Although the vast majority of EMV payments are processed without a hitch, there are occasions when the cardholder later repudiates a payment. The cardholder may be adamant that he or she was not present to authorize the transaction, but the issuing bank considers the transaction data and certificate to evidence otherwise because they show that the chip and PIN mechanism had been used to authenticate the cardholder. A solution is proposed that leaves a cryptographic mark on the card's chip which could help determine whether or not that card was present for the transaction and thereby help adjudication.

I. INTRODUCTION

EMVCo LLC[1] was formed in 1999 by Europay International, MasterCard International and Visa International to manage, maintain and enhance the "EMV Integrated Circuit Card Specifications for Payment Systems" [9, 10, 11, 12], which are primarily concerned with the interaction between the card and terminal. Europay has since been acquired by MasterCard and JCB International has now joined the organization. EMV specification payment cards are mainly deployed in Europe, Asia and South America. Canada started trials in late 2007 but the U.S. has yet to follow.

In the two years straddling the U.K.'s introduction of these EMV smart cards that have an imbedded chip secured by a Personal Identification Number (PIN), otherwise known as chip and PIN cards, face-to-face fraud with retailers fell by 67% and Automatic Teller Machine (ATM) fraud fell by 17% [3]. However there are still occasions when the cardholder is certain that a transaction is fraudulent but the cardholder's bank, the card issuer, thinks the transaction to be authentic because the transaction data and certificate presented by the payment claimant attest that the cardholder presented his or her chip card and entered the correct PIN. The issuer trusts its EMV card environment. The outcome of this discrepancy, where the cardholder is not intimating the card to have been lost or stolen, is often referred to as a "phantom withdrawal" from an ATM or simply a "disputed payment" if goods or services were purchased. Such disputes are difficult to resolve and discussions with people involved with adjudication indicate that when a PIN verified EMV transaction is completed, the card issuer feels confident that the cardholder made the transaction but will look towards the human and environmental factors when deciding who should be liable. The human factors include the length of time the issuer has known the cardholder, the cardholder's reputation with the issuer and how the issuer values the cardholder as a customer. It is in essence a nontechnical solution to a technical problem and the method is not something that is published or openly discussed.

Save for dishonest intent and forgetfulness on the cardholder's part, there are reasons why such a discrepancy could occur making both parties theoretically correct. For example: an acquaintance of the cardholder may know the PIN, "borrows" the card and returns it before its absence is noticed or a trusted person(s) may abuse a position of trust with the issuer to create perfectly formed but fraudulent transactions [1].

The term non-repudiation has differing interpretations in electronic commerce; a cardholder may be liable irrespective of who made the payment. Bohm et al. [4] state that "A technical assessment may prove that it is highly probable that (the payment) was made by its apparent maker. A legal assessment may hold that the apparent maker of (the payment) is bound by it whether the apparent maker made it or not." This paper looks at the technical assessment.

If there were to be a lasting transaction related mark left on an EMV card's chip each time the card processes a payment, then it would increase the confidence of all parties in any technical assessment and give less credence to it being a "payment system fault." It would give objectivity to an otherwise subjective assessment and protect the interests of both the card issuer and cardholder. Presently the only transaction related mark is a 16-bit Application Transaction Counter (ATC) which increments by a bit each time a transaction takes place [10, Annex D3] but it only shows how many transactions have taken place, not the specifics of the transaction, and not all transactions are to certify a payment.

A. Contribution

This paper proposes three methods for leaving a cryptographic mark of previous payments on an EMV card's chip to demonstrate whether or not the card was present for a disputed transaction. If said transaction has a matching on-card mark then it shows, beyond reasonable doubt, that the card was indeed present.

Electronic Receipts The first method provides the easiest to use form of proof, but consumes the most potentially scarce storage space on the chip and could open concerns over

[1] http://www.emvco.com/

privacy. The cardholder can view the receipts and the card issuer can verify their authenticity.

Condensed Electronic Receipts The second method is a subset of the first method and reduces the detail in each receipt making it less meaningful to the cardholder, although still quite usable. On the other hand this method increases the cardholder's privacy, uses less storage space on the chip and reduces payment-time processing. In the relatively unlikely event of the card issuer wishing to verify receipt authenticity, then this method requires more processing.

Electronic Footprint The third method further reduces the stored details and leaves it to the card issuer to demonstrate to an adjudicator, not the cardholder, whether the on-card electronic footprint includes the disputed transaction. The cardholder cannot view any receipts, but conversely the cardholder's privacy is not eroded by the proposal and this method is the most economical with the chip's storage space.

The choice of method is a balance between usability for the cardholder, cardholder privacy, point of processing overhead and storage space on the chip. All three methods could be adapted for other card-based payment systems.

II. Mechanisms For Enhancing non_repudiation properties

The proposed solutions are nothing more than an add-on to EMV and as such could be part of future EMV specifications or be part of an existing EMV compliant application. As evolution, rather than revolution, any or all three variants could be phased in as and when cards are replaced, but only one non-repudiation mechanism per card. It should also be compatible with any future EMV modes of implementation such as on a contact-less card or mobile telephone. In the event of a dispute and where there is more than one payment card associated with an account, it requires the card issuer to identify which card made the payment by means of the optional Primary Account Number (PAN) sequence number, and to pass that information to the cardholder in a digestible form. Without that information all possible cards need to be tested. The transaction data that are retained by the issuer are the likely source for that detail but transaction data content is outside the scope of the EMV specifications.

A. Electronic Receipts

Taking a lead from some digital cash systems, an electronic transaction receipt is retained on the card's chip for the card's most recent card-present transactions [7, 6, 15]. Some of those digital cash providers mention on-card receipts as an aid to resolving disputes but the presence of such information has previously contributed to concerns over terminology [21], particularly when describing the offering as anonymous rather than private – which some digital cash providers still do [14, 20]. Even when read-protected by the cardholder's PIN, which this proposal recommends, on-card transaction information is potentially available for reading by the card issuer when a card is used on-line.

The values suggested by EMV "for example purposes only" for a transaction log are listed in Fig. 1 [11, Table 45] and are proffered with the intent of supporting access to transaction logs by special devices such as personal computers, Personal Digital Assistants (PDA) and ATMs rather than for storage on an EMV card. The specifications don't say how the information is to be used and propagated or how integrity is ensured [13].

20 bytes are specified for the merchant name and location but, unlike the other mentioned data elements, it is of variable length [11, Table 33]. A unique merchant identifier is also available and at a fixed 15 bytes it is also relatively large. Omitting readable merchant information reduces each record from a minimum of 36 bytes to a fixed 16 bytes, which is beneficial in a storage constrained environment and its absence probably does not too adversely affect the usability for most cardholders. Fig. 2 outlines the proposal for an electronic receipt log.

Each log entry is constructed with the same mechanism, one line per payment. Taking the most recent payment j as an example:

X , Y represents a concatenation of X with Y.
L_j is the j^{th} entry in log L and a concatenation of the next 5 values.
D_j is the 3-byte transaction date.
T_j is the 3-byte transaction time.
C_j is the 2-byte transaction currency code.
A_j is the 6-byte amount authorized.
N_j is the 2-byte ATC.
Z_j is an 8-byte Message Authentication Code (MAC) spanning the concatenation of the previous record's MAC, Z_{j-1}, the current line, L_j , and the merchant name and location, V_j. K_S is the key and, as necessary, the message is padded according to international standard ISO/IEC 7816-4 (2005) [18] to the next 8-byte boundary which is equivalent to padding method 2 in international standard ISO/IEC 9797-1 (1999) [16]. Should this payment be disputed then the card issuer can verify the integrity of L_j and its preceding transactions.

Transaction Date	3 bytes
Transaction Time	3 bytes
Transaction Currency Code	2 bytes
Amount Authorized	6 bytes
Merchant Name and Location	20 bytes
Application Transaction Counter	2 bytes

Fig. 1. Example of EMV log format

L_{j-8}	=	D_{j-8} , T_{j-8} , C_{j-8} , A_{j-8}, N_{j-8}	↓	
L_{j-7}	=	D_{j-7} , T_{j-7} , C_{j-7} , A_{j-7}, N_{j-7}	→	Z_{j-7}
L_{j-6}	=	D_{j-6} , T_{j-6} , C_{j-6} , A_{j-6}, N_{j-6}	↓	
L_{j-5}	=	D_{j-5} , T_{j-5} , C_{j-5} , A_{j-5}, N_{j-5}	↓	
L_{j-4}	=	D_{j-4} , T_{j-4} , C_{j-4} , A_{j-4}, N_{j-4}	↓	
L_{j-3}	=	D_{j-3} , T_{j-3} , C_{j-3} , A_{j-3}, N_{j-3}	↓	
L_{j-2}	=	D_{j-2} , T_{j-2} , C_{j-2} , A_{j-2}, N_{j-2}	→	Z_{j-2}
L_{j-1}	=	D_{j-1} , T_{j-1} , C_{j-1} , A_{j-1} , N_{j-1}	↓	
L_j	=	D_j , T_j , C_j , A_j , N_j	→	Z_j

Fig. 2. Proposed electronic receipt log

$$Z_j = MACKS\ (Z_{j-1},\ L_j,\ V_j,\ PAD) \quad (1)$$

PAD = (0x80 , … , 0x00 , … , 0x00)

(i.e. Add a mandatory '*0x80*' and then '*0x00*'s if or as needed).

K_S is the EMV message authentication session key that is already used during the j^{th} transaction. Another symmetric key, the EMV application cryptogram session key, is used to create the transaction certificate.

The method for deriving a session key is for the issuer to choose, however EMV's session key derivation annex gives broad guidance on how to derive a 16-byte session key from both the card's respective 16-byte master key, K_M the message authentication master key in this case, and the previous transaction's 2-byte ATC, N_{j-1} [10, Annex A1.3]. f is a diversification function that maps the master key and ATC to a large and uniformly distributed number of 16-byte outputs. Function *PAR* then sets the least significant bit of each byte to odd parity, a Data Encryption Standard (DES) [19] algorithm recommendation for keys.

$$K_S = PAR\ (\ f\ (K_M)\ [N_{j-1}]\ ) \quad (2)$$

This proposal does not require the ATC to be included in the diversification function, only that the uniformly distributed number can be correlated with the ATC.

DES is the only approved symmetric cryptographic algorithm and uses a double-length key [10, Annex B1.1] although, depending on the version of MAC mechanism, may just use the most significant 8 bytes of the key.

$$K_S = K_{SL}\ ,\ K_{SR} \quad (3)$$

K_{SR} is ignored for single DES.

MAC Processing The MAC's input is divided in to 8-byte blocks: X_1 , X_2 , … , X_p. Those 8-byte blocks X_1 , X_2 , … , X_p are enciphered (ENC) with a block cipher in Cipher Block Chaining (CBC) mode.

$$H_i := ENC_{KSL}[\ X_i \oplus H_{i-1}\ ], \quad \text{for } i = 1, 2, \ldots, p \quad (4)$$

H_0 is an initial value of 8-bytes of binary zeros: 0x0000000000000000, which in this implementation is equivalent to using the enciphered previous MAC as the initial value.

If processed according to ISO/IEC 9797-1 algorithm 1, with one 8-byte key, then:

$$Z_j := H_p \quad (5)$$

If processed according to ISO/IEC 9797-1 algorithm 3, with a 2^{nd} 8-byte key K_{SR} , then result H_p is decrypted (DEC) and encrypted with the 2^{nd} and 1^{st} keys respectively:

$$Z_j = ENC_{KSL}\ (DEC_{KSR}\ (H_p)\ ) \quad (6)$$

EMV allows for an *s*-byte MAC, where $4 \le s \le 8$, by striping the least significant bytes [10, Annex A1.2] which mirrors international standard ISO/IEC 9797-1 (1999) [16], that uses a 64-bit block cipher algorithm in CBC mode. This proposal doesn't truncate the MAC, although it could if storage is of the essence.

A MAC does not need to be retained for every transaction, just sufficient to allow for verification. With the most recent electronic receipts on the card's chip in chronological order, at a minimum just the oldest transaction's MAC and latest transaction's MAC are needed. In normal operation those MACs need to be re-calculated just before the oldest transaction is deleted or when a new transaction is added. However any failed verification could point to an inconsistency that has since been age-deleted unless routine MAC verifications are performed to reduce the time window for such a problem. Instead, keeping a set cycle of MACs, every 5^{th} MAC, and the latest is proposed allowing 58 electronic receipts to be stored per kilobyte of Electrically Erasable Programmable Read-Only Memory (EEPROM), but the oldest few entries may not always be included in a MAC.

Each log entry remains on the card's chip until the table wraps to overwrite the oldest entry with a new transaction.

Bolstered by the card's tamper resistant properties and programming language-based technique for security, the presence or absence of an electronic receipt could be the deciding factor in a dispute; but this evidence must be considered to be in an otherwise hostile environment. Logically it is the issuer's trusted environment but physically the cardholder's trusted environment. The cardholder is primarily concerned with disclosure whereas the issuer is more concerned with integrity.

Protecting against information disclosure is by means of PIN protection with viewing or printing through a secure device: say an ATM, dedicated kiosk or hand-held reader which could be a design extension to Visa's reader that is used for dynamic passcode authentication [22][2]. In many disputes the cardholder could check the card without involving the issuer,

[2] MasterCard has its Chip Authentication Program. APACS, the UK trade association for payment providers, generically refers to such offerings as Remote Card Authentication systems.

but only the issuer can verify the MAC to ensure integrity. Both the card and its issuer are able to derive session keys and calculate the associated MACs.

EEPROM is the card's only storage area that can be both updated and preserved between payments when there is no electrical power to the circuitry. The quantity of EEPROM has quite a bearing on the cost of the card giving the issuers the incentive to purchase just enough EEPROM to meet the immediate need with not too much left over. Most Visa approved chip cards [23] contain between 2 kilobytes and 8 kilobytes of EEPROM, although some top of the range cards have as much as 64 kilobytes with even more potentially available if it were to be a dual-chip card. Per card an extra four kilobytes probably costs in the region of one US dollar when bought in bulk, although it is difficult to be precise on cost, but collectively the cost to the issuer is more notable.

Of the three, this mechanism is the most comprehensive for the cardholder but there could be concerns over EEPROM consumption and privacy with respect to the user's spending patterns, although it could replace paper receipts from the merchant and the on-card receipt does not state the payment recipient or what was purchased.

B. Condensed electronic receipts

The Condensed Electronic Receipts proposal cuts away at the demand for EEPROM, but there are both positive and negative consequences for the cardholder. As pieces of information are removed the cardholder's privacy is increased, but each receipt carries less information so placing greater dependence on the issuer for receipt verification and dispute resolution. In the event of a dispute the assurance for the issuer is equal to that for the Electronic Receipts proposal, but the cardholder needs to place greater trust in the issuer and that probably affects his or her level of assurance.

i. The mechanism

When an EMV card accepts a transaction it generates an 8-byte Application Cryptogram (AC), viz. a Transaction Certificate (TC) [11]. This lasting proof of the transaction is simply a MAC of the transaction data using the transaction's application cryptogram symmetric session key. The key is derived with the same mechanism as the message authentication session key, as discussed in Sect. 2.1, only with a different master key. The TC ties together all the data elements listed in the two Credit Risk Management Data Object Lists (CDOL1 and CDOL2) and any additional data objects that are more related to the terminal's business environment and listed in the Transaction Certificate Data Object List (TDOL). Those objects may include the cardholder verification method that shows whether the PIN was entered [11, Annex C3]. When compared to the Electronic Receipts proposal, the TC spans far more information than the electronic receipts MAC but it is not linked to adjacent transactions other than by taking increasing ATC values. On their own neither form of MAC compromises privacy and both are equally economic on storage.

There are gaps in the chain because not all ATCs result in a payment. The ATC is incremented in response to a GENERATE AC command and, for example, the AC may have be an Application Authentication Cryptogram (AAC) where the transaction was declined. It is only successful payment transactions that are of interest.

To overcome that problem, in this proposal sixteen ordered tables of 256 bits each are allocated, initially with all 4096 bits set to zero. When a TC is issued, the lowest 8 bits of the ATC are examined and the matching bit in the lowest bit map table is changed from zero to one. When the ATC is incremented and the lowest bit of the leading byte is altered then the oldest table is set to zeros and moved to become the current table with all other tables moving up one in the hierarchy. This gives an efficient method of mapping which of the previous 4,000 or so ATCs became a TC, all in half a kilobyte. It even allows for situations where the ATC does not start at zero and removes the requirement for any other mapping between TCs and ATCs.

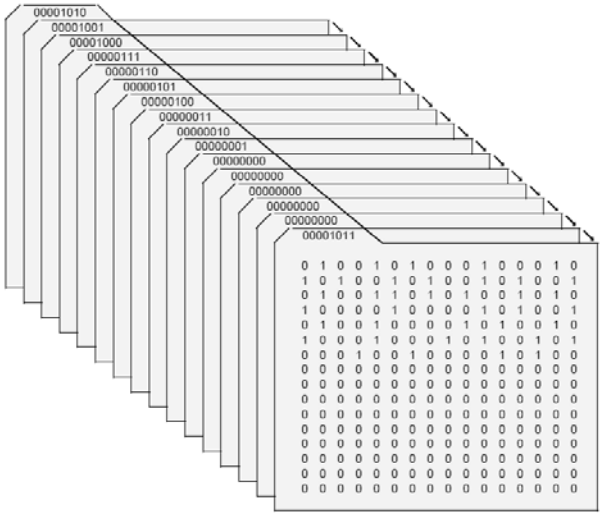


Fig. 3. Bit map of the ATCs

Fig. 3 illustrates the principle by using 16 stacked folders to represent the tables, where the last TC used ATC value 0x0B6D, decimal 2925. The highest bit that is set to one is in table 0x0B, row 0x6, and column 0xD. The last four folders have never been used so are already set to zero throughout. When the present folder becomes full and the leading byte of the ATC increments, the current folder moves to the back and the remaining folders all move forward one place with the front folder's bits set to zero. The folder tabs are not actually present; the tabs are calculated by working back from the current ATC.

$L_{j-8} = C_{j-8}, A_{j-8}$	↓	
$L_{j-7} = C_{j-7}, A_{j-7}$	→	Z_{j-7}
$L_{j-6} = C_{j-6}, A_{j-6}$	↓	
$L_{j-5} = C_{j-5}, A_{j-5}$	↓	
$L_{j-4} = C_{j-4}, A_{j-4}$	↓	
$L_{j-3} = C_{j-3}, A_{j-3}$	↓	
$L_{j-2} = C_{j-2}, A_{j-2}$	→	Z_{j-2}
$L_{j-1} = C_{j-1}, A_{j-1}$	↓	
$L_j = C_j, A_j$ →	Z_j	

Fig. 4. Proposed condensed electronic receipt log

The condensed electronic receipt log follows the same outline as the electronic receipt log but only the amount authorized and transaction currency code are available to the cardholder. Fig. 4 outlines the proposal, again with one line per payment and using similar notation.

Taking the most recent payment *j* as an example:

L_j is the j^{th} entry in log *L* and a concatenation of the next 2 values.
C_j is the 2-byte transaction currency code.
A_j is the 6-byte amount authorized.
Z_j is a similar 8-byte MAC to that outlined in the Electronic Receipts proposal but using the TC, W_j, in place of V_j. Again not all MACs are stored. In difference this variant does not require session key K_S to be correlated with the ATC.

$$Z_j = MAC_{K_S}(Z_{j-1}, L_j, W_j, PAD) \quad (7)$$

With just 8 bytes per payment, plus the MACs and the ATC table, 100 transactions and a table of 512 ATCs could be stored in each kilobyte of EEPROM. This is 72% more than the Electronic Receipts proposal, but at the expense of detail for the cardholder. As a consequence, in the event of a dispute there is a greater reliance on the card issuer and more queries need issuer involvement.

ii. CDA cards

There are three methods for confirming the legitimacy of the data on an EMV card [10, Chapters 5 & 6] and this is the main variable that influences the design, capabilities and operation of an EMV card.

Static Data Authentication (SDA) is the most basic design with the cheapest to produce cards; these cards need only process symmetric cryptography. It appears that EMV cards in the UK are predominantly SDA cards and presumably most other countries are much the same [2, 23].

Dynamic Data Authentication (DDA) is a more complex and expensive design; most notably DDA uses some asymmetric cryptography. However the use of asymmetric cryptography only extends to validating the card and not to the transaction data.

Combined DDA/Application Cryptogram Generation (CDA) is an extension to DDA and the use of asymmetric cryptography can extend to the transaction data.

Both SDA and DDA cards operate as described in this proposed solution, but when a CDA capable card is used in a CDA enabled terminal, the TC is a (longer) digital signature rather than a MAC [10, Chapter 6.6]. The Electronic Receipts proposal can still proceed because all the required information and the capability to generate a MAC are still present. The condensed electronic receipt mechanism can use either form of TC: a MAC or a digital signature.

C. Electronic Footprint

The Electronic Footprint proposal further reduces the information stored on EEPROM, leaving just the MACs and bit map of ATCs. Transaction details are not stored which ensures privacy for the cardholder, but the footprint can only be interpreted by the issuer or a Trusted Third Party (TTP). In the event of a dispute, the assurance for the issuer is equal to that given by the other two proposals but the cardholder needs to place complete trust in the participants in the dispute resolution process. The cardholder will probably have little trust in that process should the resolution not fall in his or her favor, in part because the evidence is intangible to the cardholder. Involving a TTP gives independence to the dispute resolution process.

When a TC is created the bit map of ATCs is updated. In the event of a dispute, the card issuer identifies the ATC value and the bit map can be checked for the presence or absence of a matching entry. Some disputes may be resolved by this simple check alone.

The MAC is created and retained along similar lines to the Condensed Electronic Receipts proposal. Fig. 5 outlines the proposal, illustrated with the 9 most recent payments and using similar notation.

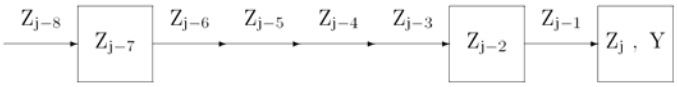


Fig. 5. Proposed electronic footprint

Taking the most recent payment *j* as an example:

Z_j is still an 8-byte MAC using key K_S, but just covering a concatenation of the previous record's MAC, Z_{j-1} and the TC, W_j. All other aspects of the MAC mechanism are as outlined in the Condensed Electronic Receipts proposal.

$Z_j = MAC_{K_S}(Z_{j-1}, W_j, PAD)$

For SDA and DDA cards, an XOR operation could have been used to amalgamate the two 8-byte codes, Z_{j-1} and W_j , but the resulting Z_j would only be the result of two encipherments, rather than three, with potentially known input text.

A hash is not used because EMV only uses a hash for digital signatures and SDA cards are not capable of producing signatures. Some SDA cards may not be able to produce a hash because it is a redundant capability consuming scarce resources.

Y is a 4-bit counter towards the next boundary for retaining a MAC. 4-bits are allocated so that the counter can increment to any value between 1 and 15, with a maximum value of 4 used in this example.

With just one-in-five MACs plus the latest MAC and the ATC table, 400 transaction footprints and a table of approximately 3,000 ATCs could be stored in each kilobyte of EEPROM. This is 590% more than the Electronic Receipts proposal, but void of detail for the cardholder. In the event of a dispute there would be absolute reliance on the TTP who would require the card issuer to detail how the MAC that spans the disputed entry was built, specifically: the transaction certificates, the relevant message authentication and application cryptogram session keys, and the data that made up each transaction certificate. With increasing brevity the burden moves to a small overhead with every transaction with a far greater overhead on each dispute.

D. Privacy and Integrity

Each of the three types of marking is a combination of transaction information with a MAC that binds the current payment with the preceding payment.

Integrity of the transaction marks is ensured with a MAC which is created using a key that can be derived by both the card and the card issuer. Logically the card is the issuer's trusted environment but physically the card is the cardholder's trusted environment. It is theoretically possible for an agent of the issuer with inappropriate access to tamper with the marks during a transaction, but the difficulties of accessing that information and ensuring concurrency make exploitation exceedingly difficult. Should the general integrity of the marks be in question, then in the event of a dispute it is possible to prove the correctness of the other transactions by regenerating the preceding MACs. This proposal's MAC is cryptographically similar to the transaction certificate that ensures the integrity of the transaction data; they both use a similar process but with different master keys.

Privacy for the cardholder could be eroded by the inclusion of information that another party may be able to read. On the EMV card that information is resistant to physical and logical penetration but there are occasions when others may read the data, covertly or overtly. For example, by someone looking over the cardholder's shoulder when viewing the receipts at an ATM or by any person involved with resolving a dispute. The electronic receipts proposal provides the most information to keep the cardholder fairly self-sufficient but it also has the most to disclose. The electronic footprint does not allow the cardholder to make any meaningful assessment of the transaction data, hence the need for a TTP, but there is little additional information to disclose. The condensed electronic receipt proposal seeks the middle ground.

III. RESIDUAL THREAT ANALYSIS

As with any application, there are potential threats to this proposal's smooth operation. These mainly concern the processing, storage and presence of the additional evidence of payment rather than affecting the fabric of the underlying payment process. That said, the vast majority of payments will continue to be processed without a hitch.

A. The Evidence Causing Disputes

Particularly with electronic receipts and to a lesser extent with condensed electronic receipts, the cardholder is able to view the on-card evidence of card-present payments and may be able to see an opportunity to dispute a genuine payment. This is most likely to be the manifestation of another listed residual threat. However, the transaction data and its certificate would still be valid and whenever a cardholder disputes a payment the human factors presently used still have a role to play.

B. Too Much Trust in the System

Where a disputed payment is supported by an on-card receipt or footprint, the case for dismissing the dispute is strengthened. But for example, a new card and PIN could be sent through the postal service and both intercepted, exploited and then forwarded to the intended recipient. Particularly where there is no obvious evidence of attack, the presence of the on-card receipt or footprint could cause a wrong conclusion to be drawn.

C. Race Conditions

A residual threat with all three proposed solutions is that the receipt log and the ATC table are both smaller than the total number of possible entries and when either or both are full they wrap to overwrite the oldest records.

The Condensed Electronic Receipt and Electronic Footprint proposals both use a bit map of ATCs that resulted in a TC. For example, if an attacker could cause the ATC to increase by the number of entries in the ATC table through bogus activity without issuing a transaction certificate, then the bit map would no longer relate to any of the transactions. This could be counteracted by increasing the size of the bit map to 8 kilobytes so as to span all possible ATC values. However exploitation should be evident to the issuer by the sudden increase in ATC values. This attack would not affect the other transaction information, for example any transaction log and the MACs.

With all three proposals only a certain number of transactions are retained on the card's chip which could be a genuine problem for cards in very frequent use. There are 65,535 (2^{16}–1) available ATC values but not all result in a TC although if this were to happen and all transactions were stored, 1.1 megabytes would be needed for the electronic receipts, 614 kilobytes for the condensed electronic receipts and 102 kilobytes for the electronic footprint.

D. Misplaced Trust

Both the card issuer and cardholder trust others. That trust could be misplaced, or appear to be misplaced.

The issuer is responsible for preparing an EMV card and setting the payment application criterion, retaining the capability to derive all keys held on or generated by a card. Insiders may be able to exploit the issuer's trust but the scope for fraud is very much reduced because any exploit has to keep the chip's cryptographic mark in synchronization with the issuer's records. Once the card is in the field it is very difficult to insert an erroneous transaction although the issuer has the theoretical capability. The issuer probably supplies a

handheld reader for viewing receipts, but the cardholder can use independent devices such as an independent ATM or another issuer's portable reader. Many recorded exploits and errors [1] could have been detected through any of the proposed mechanisms. The most effective issuer control against this threat is to segregate the people analyzing disputes from the people with access to other parts of payment card processing; using a TTP for disputes gives much of that segregation.

If used, the TTP resolves disputes and is therefore in a position of trust. It may well be selected and financed by the issuer, making the cardholder perceive it to be an extension of the issuer.

The trust that cardholders need to place in card terminals and the difficulties with detecting rogue terminals is a threat that still requires attention [2, 3.5] [8]. Cited as an example is where a modified terminal with a genuine card communicates with a nearby genuine terminal with a modified card in real time so that the transaction appears as if the genuine card made a payment on the genuine terminal. The false terminal hoodwinks the cardholder into paying for a high value item rather than the intended purchase. However this proposal could help detection because the higher value transaction's receipt would be recorded on the card's chip and the cardholder may notice before receiving his or her statement.

E. Cardholder Practices

None of the proposed solutions address cardholder practices. In particular, if the PIN is not kept secret and another person who knows the PIN makes use of the card and returns it before the card's absence is noticed, then the transaction appears to be genuine with a matching on-card receipt. The cardholder agreement should cover this situation from a legal liability perspective and a technical assessment under this proposal will work as intended to show that the card was present, ideally causing the cardholder to consider a culprit close to home. This scenario could be addressed by extending the electronic receipt proposal: taking and keeping a copy of a biometric sample on the card's chip with the transaction mark. Although storage space is configuration, biometric and algorithm dependent, finger minutiae handled in keeping with international standards use 102 bytes per sample when viewing just one finger on an extraction device that is set to limit data capture to 12 minutiae, the recommended minimum for verification [17, Annex D.1.1]. The number of receipts per kilobyte reduce from 58 to 8½, making biometrics an expensive option and impractical for cards in frequent use. Upgrading a low-specification 2-kilobyte card to a top-of-the-range 64 kilobyte card allows 530 electronic receipts with a biometric to be stored, almost 6 a day for three months which would be reasonable for many users. Boyd proposes a non-repudiation mechanism using biometrics to link Internet payments to the payment maker in an environment where that amount of storage is not a constraint [5].

F. Inconsistencies between Statements

On-card receipts and footprints only include payments processed by the card's chip. The most notable omissions are: Card Not Present (CNP) transactions, any fallback mode payments that use the magnetic stripe and credit card repayments. Conversely, any card-processed payments that have not been presented to the issuer will be missing from any issuer supplied statement. Payments that are authorized before the final transaction amount is known will also not necessarily tally, with the card showing the amount authorized but the statement showing the final transaction amount including tips and other adjustments [12, Sect. 6.5.1]. Neither source can give an up-to-the-minute balance of account.

IV. CONCLUSION

This paper proposes three mechanisms for enhancing the non-repudiation properties of EMV chip and PIN payments by creating a cryptographic mark of the most recent transactions processed by the card. In the event of a disputed payment, the card's chip could be inspected to tell whether the card was present for the transaction and the presence or absence of a cryptographic mark used to help decide the outcome.

The proposal does not affect the underlying payment process but there are occasions when the card does not participate in a transaction or is only aware of the amount authorized and not the final transaction amount. The presence or absence of a receipt could cause disputes or encourage too much trust to be placed in the payment system. Capturing a biometric sample from the cardholder at the time of payment would create a closer linkage between the person and the transaction but is an expensive option for storage.

All three proposed mechanisms consume some computing resource at the time of payment and storage space on the card, but these are presented to strike a balance between payment-time overhead, cost and overhead in the event of a dispute and designed to operate as an optional application extension to the present EMV specifications. As a framework rather than a specific design, the amount of information processed and stored could be further adjusted in keeping with need, cost and risk.

REFERENCES

[1] R. J. Anderson. Why Cryptosystems Fail. *Communications of the ACM*, 37(11):32-40, November 1994.

[2] R. J. Anderson, M. Bond, and S. Murdoch. Chip and Spin. *Computer Security Journal*, 22(2):1-6, 2006.

[3] APACS. Card fraud losses continue to fall. Press rlse., APACS, March 2007. http://www.apacs.org.uk/media_centre/press/07_14_03.html.

[4] N. Bohm, I. Brown, and B. Gladman. Electronic Commerce: Who Carries the Risk of Fraud? *The Journal of Information, Law and Technology*, Issue 3, October 2000.

[5] *J. Boyd. Enhancing the Non-Repudiation Properties of Internet Payments Through a Third Dimension. In K. Adi, M. Debbabi, and L. Logrippo, editors, Proceedings of the 2nd Workshop on Practice and Theory of IT Security (PTITS), pages 33-39, January 2008.*

[6] *Chaum. Achieving Electronic Privacy. Scientific American, pages 96-101, August 1992.*

[7] *K. Clemons, D. C. Croson, and B. W. Weber. Reengineering money: the Mondex stored value card and beyond. In System Sciences, 1996.*

Proceedings of the 29th Annual Hawaii International Conference on System Sciences., volume 4, pages 254-261, January 1996.

[8] S. Drimer and S. Murdoch. Keep your enemies close: Distance bounding against smart card relay attacks. In 16th *USENIX Security Symposium. Security '07*, August 2007.

[9] EMVCo LLC. Integrated Circuit Card, Specifications for Payment Systems. Version 4.1. Book 1 — Application Independent ICC to Terminal Interface Requirements. EMV, May 2004.

[10] EMVCo LLC. Integrated Circuit Card, Specifications for Payment Systems. Version 4.1. Book 2 — Security and Key Management. EMV, May 2004.

[11] EMVCo LLC. Integrated Circuit Card, Specifications for Payment Systems. Version 4.1. Book 3 — Application Specification. EMV, May 2004.

[12] EMVCo LLC. Integrated Circuit Card, Specifications for Payment Systems. Version 4.1. Book 4 — Cardholder, Attendant, and Acquirer Interface Requirements. EMV, May 2004.

[13] EMVCo LLC. Specification Update Bulletin Nº19. EMV, May 2004.

[14] EURO Kartensysteme GmbH. Answers to some frequently asked questions on GeldKarte.

[15] http://www.geldkarte.de/_www/en/pub/geldkarte/service_navigation/faq.php, 2007.

[16] GeldKarte. Home page — http://www.geldkarte.de.

[17] International Organization for Standardization and International Electrotechnical Commission. ISO/IEC 9797-1:1999. Information technology - Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher, 1997.

[18] International Organization for Standardization and International Electrotechnical Commission. *ISO/IEC 19794. Information technology - Biometric data interchange formats - Parts 2: Finger minutiae data*, September 2005.

[19] International Organization for Standardization and International Electrotechnical Commission. ISO/IEC 7816-4:2005. Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange, 2005.

[20] National Institute of Standards and Technology (NIST). Federal Information Processing Standards (FIPS) Publication 46-3 - Reaffirmed - Data Encryption Standard (DES), October 1999.

[21] PayLife Bank GmbH. Quick Wertkarten).

[22] http://www.quick.at/plb/opencms/de/Home/Ueber_Quick/Karten_mit_Quick_Funktion/Quick-Wertkarten/index.html, 2008.

[23] Privacy International. Mondex Decision Letter (from Office of Fair Trading). http://www.privacy.org/pi/activities/mondex/mondex_response.html, June 1996.

[24] Visa Europe. Dynamic Passcode Authentication. Overview Guide. http://www.visaeurope.com/documents/merchant/dynamicpasscodeauthentication.pdf, 2006.

[25] Visa International Service Association. Visa Approved, Visa Smart Debit Credit (VSDC) Chip Cards.

[26] https://partnernetwork.visa.com/vpn/global/retrieve_document.do?documentRetrievalId=69, May 2007.

Formal Methods for Intrusion Detection of Windows NT Attacks

Sahika Genc
Sensor Informatics Technologies Laboratory, Computing and Decision Sciences Laboratory
General Electric Global Research Center
Niskayuna, NY 12309, USA
gencs@ge.com

***Abstract*—This study develops a method to build Finite State Automaton (FSA) that models the dependencies between the operating-system (OS)-level events recorded in the audit logs of a Windows NT machine. The FSA model contains both sequential and branching relations among audit log events that help the system administrator follow the steps of the intruder. Audit-log-FSA (ALFSA) are relatively proportional to the size of the audit log, hence, may have too many states and transitions. The superset-intrusion-signature-FSA (SISFSA) is extracted using formal methods on ALFSA. The SISFSA accepts a super set of the intruder's steps since not all of its actions may result in system failure, yet it provides the system administrator with a smaller set of patterns compared to ALFSA. Once the intrusion signature is finalized, fault diagnosis methods, developed for detection of faults in systems such as Heat Ventilation and Air-Conditioning (HVAC) systems, are used to detect the intrusion under full and partial-observation. Performance results are provided in the computer intrusion data set collected by Information Systems Technology Group at Massachusetts Institute of Technology Lincoln Laboratory under Defense Advanced Research Projects Agency and Air Force Research Laboratory (IST-MIT-LL data set) in 1999 for Windows NT attacks.**

***Index Terms*—Intrusion detection, discrete-event systems, pattern diagnosis.**

I. Introduction

Intrusion detection systems provide the system administrator with various tools, one of which is audit log that helps the administrator analyze the intruder's steps to develop detection and defense mechanisms against future intrusions. There are three types of Windows NT audit logs: 1) System, 2) Security, and 3) Application. Windows NT audit log files keep records of events defined in the audit policy. Each computer system with Windows NT has an audit policy defined in the Start Menu's Programs, Administrative Tools, User Manager, Policies, Audit, and the Audit Policy dialog box. Full-auditing functionality enables auditing of logon/logoff, file and object access, use of user rights, user and group management, security policy changes, restart, shutdown, and system, process tracking events for all user accounts, and base objects such as files and drivers. Information Systems Technology (IST) Group at Massachusetts Institute of Technology (MIT) Lincoln Laboratory under Defense Advanced Research Projects Agency and Air Force Research Laboratory designed a test bed computer network to collect data sets for evaluation and study of intrusion detection systems in 1998 and 1999. In 1999, three Windows NT machines were added to the test bed: victim, inside- and outside- attacker. Twelve Windows NT attacks including denial-of-service attacks, remote-to-local attacks, and user-to-root attacks were developed. A detailed description of the test bed and Windows NT attacks are shown in [1]. The author successfully used Windows NT audit logs of the victim machine to detect ten out of twelve attacks under full-auditing. He also showed that ten attacks leave a trace in the Windows NT Security Log file, but not in System or Application Log files, and two attacks left no trace in any of the NT audit log files, and developed intrusion signatures based on the traces left in the Windows NT Security Log files collected during the 1999 evaluation.

There have been studies in the literature on FSA based methods for intrusion detection. In [2], the authors monitor privileged processes on a UNIX machines such as TELNETD and LOGIND. They assume that the process is a black box and outputs *observable* (system call) that describes the dynamics characteristic of the program. The abnormal behavior is defined as deviations from the normal behavior and can be quantized through various measures. The authors were able to detect several classes of abnormal behavior such as failed intrusion attempts and error conditions in the UNIX program SENDMAIL. In [3], the authors developed methods that overcome the difficulties in [2], namely complexity in building FSA. The weakness of [3] was in tracing program calls for FORK and TRACE. In [4], the authors present an algorithm for automatically constructing a FSA that accepts the normal behavior and rejects all the other sequences. The algorithm minimizes the FSA by substituting macros for frequently occurring sequences and combining multiple sequences. The results of the experiments on several well-known data sets were promising: perfect detection rate, low False Positive Rate (FPR), and improved performance with respect to [2].

Our approach considers OS-level events logged by Windows NT auditing program, and combines the human intuition with machine automation. That is, one can easily go through the event set and built simple relations such as "*Process A creates process B*". The number of audit-log events is finite and there are few significant events used by the system administrator [5]. This is not much different that what a system administrator would do to identify the intruder's step after a system failure

or detection of abnormal behavior or otherwise during routine checks for system consistency. However, the administrator would have to analyze one or more audit-logs from start to end building these relations on the spot. Our approach automates this processes of parsing through thousands of audit-event logs. Building dependency (relational) graphs (e.g., FSA) to identify the sequences of OS-level events that led to an intrusion has been studied in the literature. In [6], authors describe a tool BackTracker that identifies event sequences leading to intrusion detection point. Intrusion detection point refers to the state on the local computer upon detection of a compromise such as modified system file or port-scan. The authors recommend various tools in the literature and market to detect a compromise. BackTracker works backwards from the intrusion detection point and forms chains of events that lead to the compromise. Then, an administrator may analyze these sequences of events to quickly identify vulnerabilities in the system. Our study follows a more comprehensive approach. First, there is no other software that detects a compromise. Second, our approach requires a collection of audit logs to identify the steps of the intruder, i.e., intrusion signature, and then, simple FSA operations are performed on the collection of audit logs to minimize the set of sequences the administrator works on to understand the vulnerabilities. Our study differs from [1] because in that study extraction of intrusion signatures requires extensive knowledge on the structure of the attack. In this study, signature extraction is automated through series of FSA operations on the collection of audit-log files collected during normal operation and intrusion. The audit-logs with intrusion is used to identify similar abnormal behavior that differs from normal behavior.

The paper is organized as follows. In Section II, a summary of terms, notation, and operations in FSA theory is presented. In the following sections the construction of ALFSA and SISFSA are described, respectively. Then, intrusion signature diagnosis is described. Throughout the paper, the techniques and methods are supported via examples built from the audit-logs and attacks in IST-MIT-LL data set. Finally, some concluding remarks are provided.

II. Preliminaries

Let Σ be a finite set of events. A *string* is a finite-length sequence of events in Σ. Given a string s, the length of s (number of events including repetitions) is denoted by $\parallel s \parallel$. The set of all strings formed by events in Σ is denoted by Σ^*. The set Σ^* is also called the Kleene-closure of Σ. Any subset of Σ^* is called a *language* over Σ. The *prefix-closure* of language L is denoted by $\overline{L}$ and defined as $\overline{L} = \{s \in \Sigma^* : \exists t \in L \text{ such that } st \in L\}$. Given a string $s \in L$, L/s is called the *post-language* of L after s and defined as $L/s = \{t \in \Sigma^* : \exists st \in L\}$. L is *live* if every string in L can be extended to another string in L. Let L be a language over $\Sigma = \Sigma_o \dot\cup \Sigma_{uo}$, where Σ_o and Σ_{uo} denote the observable and unobservable event sets, respectively. The projection of strings from L to Σ_o^* is denoted by P. Given a string $s \in L$, $P(s)$ is obtained by removing unobservable events (elements of Σ_{uo}) in s. The inverse projection of a string $s_o \in \Sigma_o^*$ with respect to L is denoted by $P^{-1}(s_o)$ and is equal to the set of strings in L whose projection is equal to s_o.

Given an event $\sigma \in \Sigma$ and a string $s \in \Sigma^*$, we use the set notation $\sigma \in s$ to say that σ appears at least once in s. Given a string of the form $u = stv$ in L where s and t are also strings in L, then s is called a *prefix*, t is called a *substring* of u, and v is called a *suffix* of u. Given a string $s \in L$, a *subsequence* of s is obtained by deleting zero or more events in the string s.

Given a language L and a finite set of bounded strings K over Σ, we define the set $\mathcal{S} \subseteq L$ as $\mathcal{S} = \{s \in L : (\exists u \in K)(u \text{ is a subsequence of } s)\}$ and the set $\mathcal{T} \subseteq L$ as $\mathcal{T} = \{s \in L : (\exists u \in K)(u \text{ is a substring of } s)\}$. We define the set $\Psi_{\mathcal{S}}(K) \subseteq \mathcal{S}$ as $\Psi_{\mathcal{S}}(K) = \{s\sigma \in \mathcal{S} : (\exists u\sigma \in K)(u\sigma \text{ is a subsequence of } s\sigma)\}$ and the set $\Psi_{\mathcal{T}}(K) \subseteq \mathcal{T}$ as $\Psi_{\mathcal{T}}(K) = \{s\sigma \in \mathcal{T} : (\exists u\sigma \in K)(u\sigma \text{ is a substring of } s\sigma)\}$.

A Finite State Automaton (FSA) is a five-tuple

$$G = (Q, \Sigma, \delta, q_0, F) \tag{1}$$

where Q is the set of states, Σ is the finite set of events, $\delta : Q \times \Sigma \rightarrow Q$ is the state transition function, q_0 is the initial state, and $F \subseteq Q$ is the set of marked states. We extend δ from domain $Q \times \Sigma$ to domain $Q \times \Sigma^*$ as follows: $\delta(q, \epsilon) = q$, $\delta(q, s\sigma) = \delta(\delta(q, s), \sigma)$, for $s \in \Sigma^*$ and $\sigma \in \Sigma$. The language *generated* by G is $\mathcal{L}(G) = \{s \in \Sigma^* : \delta(q_0, s) \text{ is defined.}\}$. The language *marked* by G is $\mathcal{L}_m(G) = \{s \in \Sigma^* : \delta(q_0, s) \in F\}$. A set of states $\{q_1, \ldots, q_l\} \subseteq Q$ and a string $\sigma_1 \ldots \sigma_l \in \Sigma^*$ form a *cycle* in G if $q_{i+1} = \delta(q_i, \delta_i)$ for $i = 1, \ldots, l-1$ and $q_1 = \delta(q_l, \sigma_l)$.

Product of two FSA G_1 and G_2 is denoted by $G_1 \times G_2$ and defined as

$$G_1 \times G_2 = ((Q_1 \times Q_2), \Sigma_1 \cap \Sigma_2, \delta, (q_{0,1}, q_{0,2}), (F_1 \times F_2)) \tag{2}$$

where $(Q_1 \times Q_2)$ is the cartesian product of two sets and the state transition function is $\delta((q_1, q_2), \sigma) = \{(\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)), \sigma \in \Sigma_1 \cap \Sigma_2\}$. The language generated by the product of two FSA is $\mathcal{L}(G_1 \times G_2) = \mathcal{L}(G_1) \cap \mathcal{L}(G_2)$ and the language marked is $\mathcal{L}_m(G_1 \times G_2) = \mathcal{L}_m(G_1) \cap \mathcal{L}_m(G_2)$. Product operation synchronizes the event sequences common to both FSA.

Parallel composition of two FSA G_1 and G_2 is denoted by $G_1 || G_2$ and defined as in Equation 2 where the state transition function is

$$\delta((q_1, q_2), \sigma) = \begin{cases} (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)), & \sigma \in \Sigma_1 \cap \Sigma_2 \\ (\delta_1(q_1, \sigma), q_2), & \sigma \in \Sigma_1 \setminus \Sigma_2 \\ (q_1, \delta_2(q_2, \sigma)), & \sigma \in \Sigma_2 \setminus \Sigma_1 \end{cases} \tag{3}$$

The language generated by the parallel composition is $\mathcal{L}(G_1 || G_2) = P_1^{-1}(\mathcal{L}(G_1)) \cup P_2^{-1}(\mathcal{L}(G_2))$ and the language marked is $\mathcal{L}_m(G_1 || G_2) = P_1^{-1}(\mathcal{L}_m(G_1)) \cup P_2^{-1}(\mathcal{L}_m(G_2))$. Parallel composition of the two FSA synchronizes on the common sequences of events while keeping the rest of the sequences from both FSA. Both product and parallel composition operations can be extended to more than two FSA.

TABLE I
WINDOWS NT AUDIT-LOG EVENT IDS IMPLEMENTED IN THIS STUDY AND THEIR DESCRIPTIONS.

Windows NT Audit-log event ID	Description
528	Logon/Logoff: Successful logon
538	Logon/Logoff: User logoff
576	Privilege use: Special privileges assigned to new logon
577	Privilege use: Privileged Service Called
592	Detailed Tracking: Create Process
593	Detailed Tracking: A process has exited
636	Account Management: Security Enabled Local Group Member Added
639	Account Management: Security Enabled Local Group Changed
642	Account Management: User Account Changed
632	Account Management: Security Enabled Global Group Member Added

Complement of FSA G with respect to an event set Σ is another FSA G^{comp} such the language generated by the complement FSA accepts $\Sigma^* \setminus \mathcal{L}(G)$.

III. AUDIT-LOG FINITE STATE AUTOMATA

Windows NT audit-log files record events with a unique event identification number (ID). Each audit-log event is stored with various fields. A complete list of events, their descriptions, and fields can be found in [7]. The list of audit-log events modeled in this study are provided in Table I. We define (natural) relations based on the fields of audit-event. Fields are specific to each audit-log event type. For example, event *592* logs creation of a new process and stores the new process ID and creator's process ID in two separate fields. Thus, a relation based on these fields can be "*Creator process with ID 2154371904 creates a new process with ID 2153437184*". Each relation is mapped to an FSA in consistent with its natural language description. For example, the relation "*Creator process with ID 2154371904 creates a new process with ID 2153437184*" is mapped to FSA with the set of states $Q = \{$ *Creator process ID*, *New process ID* $\}$, event set $\Sigma = \{$ *create* $\}$, and the state transition function *Creator process ID* $=$ *δ(New process ID, create)*. In this study, we focus on the audit-log event *592*. The ALFSA algorithm composes the FSA built for each audit-log event together through their states like blocks quilted together through their edges. Thus, FSA for each audit-log event type or each audit-log event can be built independently and composed into ALFSA as required. In this section, we describe the ALFSA algorithm formally and provide examples from IST-MIT-LL data set.

Windows NT audit log files of the 1999 evaluation are distributed freely on [8]. We convert the audit log files to ASCII format with *Event Log Explorer* [9]. An example log of a *592* event is shown in Table II. The set of relations for this event is $R = \{$*Creator logs into Domain with User Name and Logon ID, Creator runs Creator process ID, Creator process ID creates new process ID with Image File Name* $\}$. The FSA built is shown in Fig. 1 on the left. The initial state is 0 and there is a hierarchical relation between *Domain, User Name,* and *Logon ID*. Now suppose that there is another *592* logged after the *592* event shown in Table II and have the same *Domain, User Name,* and *Logon ID* but the creator and new process IDs are different. When these two events are composed, the resulting FSA is as shown in Fig. 1 on the right. That is, the dashed (green line) states and transitions are added to the FSA of the *592* event shown on the left. That is, the same user runs another process that creates a new process. The FSA for an audit-log event is sequential. The branching processes are formed through the composition of the FSA.

TABLE II
592: CREATE PROCESS AUDIT-LOG EVENT EXTRACTED FROM ONE OF THE AUDIT-LOG FILES IN IST-MIT-LL DATA SET.

Date	3/29/1999
Time	9:43:08 AM
ID	592
Description	Security Detailed Tracking
Host	HUME
Details	A new process has been created:
	New Process ID: 2153437184
	Image File Name: INSTALL.EXE
	Creator Process ID: 2154371904
	User Name: Administrator
	Domain: EYRIE
	Logon ID: (0x0,0x3AFF)

We now define the composition operation. Suppose that there are N *592* events. Let $G_i = (Q_i, \Sigma_i, \delta_i, q_{0,i}, F_i)$ denote the FSA for a single *592* event. ALFSA for *592* events is

$$ALFSA = (Q, \Sigma, \delta, q_0, F) \tag{4}$$

where $Q = \cup_{i=1}^{N} Q_i$, $\Sigma = \cup_{i=1}^{N} \Sigma_i$, $\delta = \cup_{i=1}^{N} \delta_i$, $q_0 = 0$, and $F = \cup_{i=1}^{N} F_i$. That is, ALFSA (except the initial state) is simply the union of all the FSA built for the *592* events. The number of states, events, and transitions of the *592* ALFSA built for the audit-logs in IST-MIT-LL data setis listed in Table III. Audit-logs were collected everyday for five weeks. The first three weeks of the evaluation contain normal data, i.e., no attacks. Attacks were performed during the last two weeks of the evaluation. Two files in the data set did not convert to ASCII files, thus, are excluded. The audit-logs that are not listed in Table III and convert to the ASCII format correctly did not provide additional information and are excluded in our study.

IV. SUPERSET INTRUSION SIGNATURE FINITE STATE AUTOMATA

The construction of SISFSA requires two sets of audit-logs: 1) logs during normal operation and 2) logs collected during intrusion. We assume that the audit-log files are not altered by the intruder. Our goal is to identify the event sequences

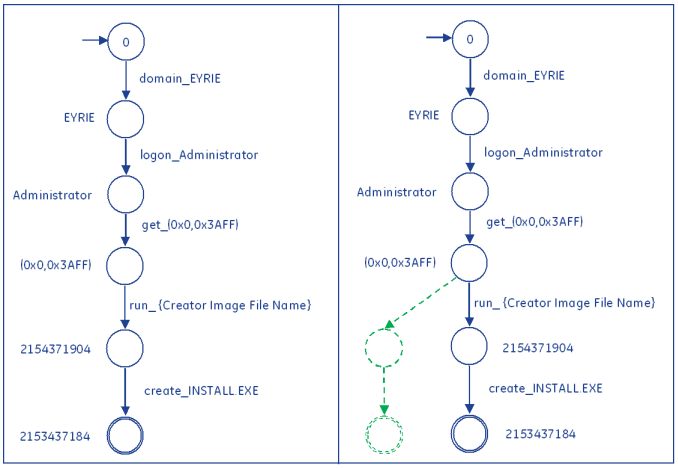


Fig. 1. FSA for *592* and composition of the two FSA for *592* events.

TABLE III
NUMBER OF STATES, EVENTS, AND STATE TRANSITIONS OF *592: Create process-ALFSA* BUILT FOR IST-MIT-LL DATA SET.

Week	Day	Number of states	Number of events	Number of transitions
1	Monday	925	183	1354
1	Wednesday	208	90	302
1	Thursday	178	79	233
2	Thursday	123	77	161
2	Friday	99	69	120
3	Wednesday	52	54	63
3	Friday	74	62	92
4	Monday	123	84	158
4	Tuesday	186	102	248
5	Monday	238	123	325
5	Tuesday	150	87	199
5	Thursday	382	94	437

that appear in the logs collected during the intrusion but are not contained in the logs during normal operation. Thus, we define SISFSA as follows

$$SISFSA = (G_1^A \times \ldots \times G_n^A) \times (G_1^N || \ldots || G_m^N)^{comp} \quad (5)$$

where the superscript A and N stands for ALFSA built from logs collected during intrusion and normal operation, respectively. The product operation synchronizes the sequences common during intrusion. Parallel composition forms a complete picture of the normal behavior by synchronizing on the common sequences and accounting for the individual sequences in each normal ALFSA. The complement operation with respect to the event set of the product of the logs during intrusion builds sequences that are not in the normal behavior. The product of the complement of the normal behavior with the synchronized log collected during intrusion creates the SISFSA.

We now consider two attacks implemented in the 1999 evaluation CrashIIS and Yaga. In [1], the CrashIIS attack is described as a denial-of-service attack against the Windows NT IIS web server. The attacker sends a malformed GET request via telnet to port 80 that crashes the web server, FTP and Gopher daemons. The victim machine was attacked with CrashIIS on Week-4-Tuesday, Week-5-Monday, and Week-5-Tuesday. The normal audit-log files used in the example are listed in Table 3, i.e., the audit-log collected during the first 3 weeks of the evaluation. We automate the calculation of SISFSA via a MATLAB batch file that calls the UMDES Software Library [10] to perform FSA. The SISFSA for CrashIIS in the IST-MIT-LL dataset is shown. When IIS service is turned on, a process called INETINFO.EXE is created and when it crashes DRWTSN32.EXE (debugger for application errors) is started. The CrashIIS SISFSA shown in Figure 3. The CrashIIS SISFSA contains the event sequence INETINFO.EXE followed by DRWTSN32.EXE which indicates that the attacked log files contain the CrashIIS attack.

In [1], the Yaga attack is described as follows: Yaga edits the Registry by replacing lines corresponding to the call for DRWTSN32.EXE when a service crashes. The intruder attacks the victim machine with CrashIIS that causes a service crash and the replaced line in the Registry is executed that gives the intruder administrative privileges in the Domain. The audit-log event *592: Create process* ALFSA identifies the part of the Yaga attack that involves processes. We can get a more complete picture by including the Account Management audit-log events (see Table 1) in ALFSA. The victim machine was attacked with Yaga on Week-4-Monday, Week-5-Thursday. The Yaga SISFSA is shown in Figure 4. When the service crashes (via CrashIIS attack) INETINFO.EXE is run and then the replaced line in the Registry executes NET.EXE to add the intruder to the Domain. In Figure 4, there are two sequences where INETINFO.EXE creates NET.EXE: 1) States 8-12-9 and 2) States 10-19-9. Also, the very same user who probably replaced the DRWTSN.EXE with NET.EXE runs REGEDIT.EXE. The CrashIIS attack used to crash the IIS service is not apparent in this FSA because of a bug in MAILSRV.EXE that was discovered later during the evaluation. The bug in the MAILSRV.EXE results in calling DRWTSN32.EXE. However, it is seen in Figure 4 that the processes (states 8 and 10) that run part of the Yaga attack, i.e., INETINFO.EXE followed by NET.EXE, also runs MAILSRV.EXE followed by DRWTSN.EXE suggesting the link with the service crash.

V. DIAGNOSIS OF INTRUSION SIGNATURE

In the previous section, we described a method to extract a super set of intruder's steps from two sets of audit-logs where one set is collected during normal operation and the other during intrusion. We showed that event sequences in SISFSA might provide enough evidence to identify exactly the intruder's steps. In this section, we present a method for pattern diagnosis that has been successfully used to detect and isolate patterns in partially-observed discrete-event systems (DES) such as HVAC that result in system failure. In HVAC, observable events can be events that are directly recorded by sensors attached to the system. The reason we

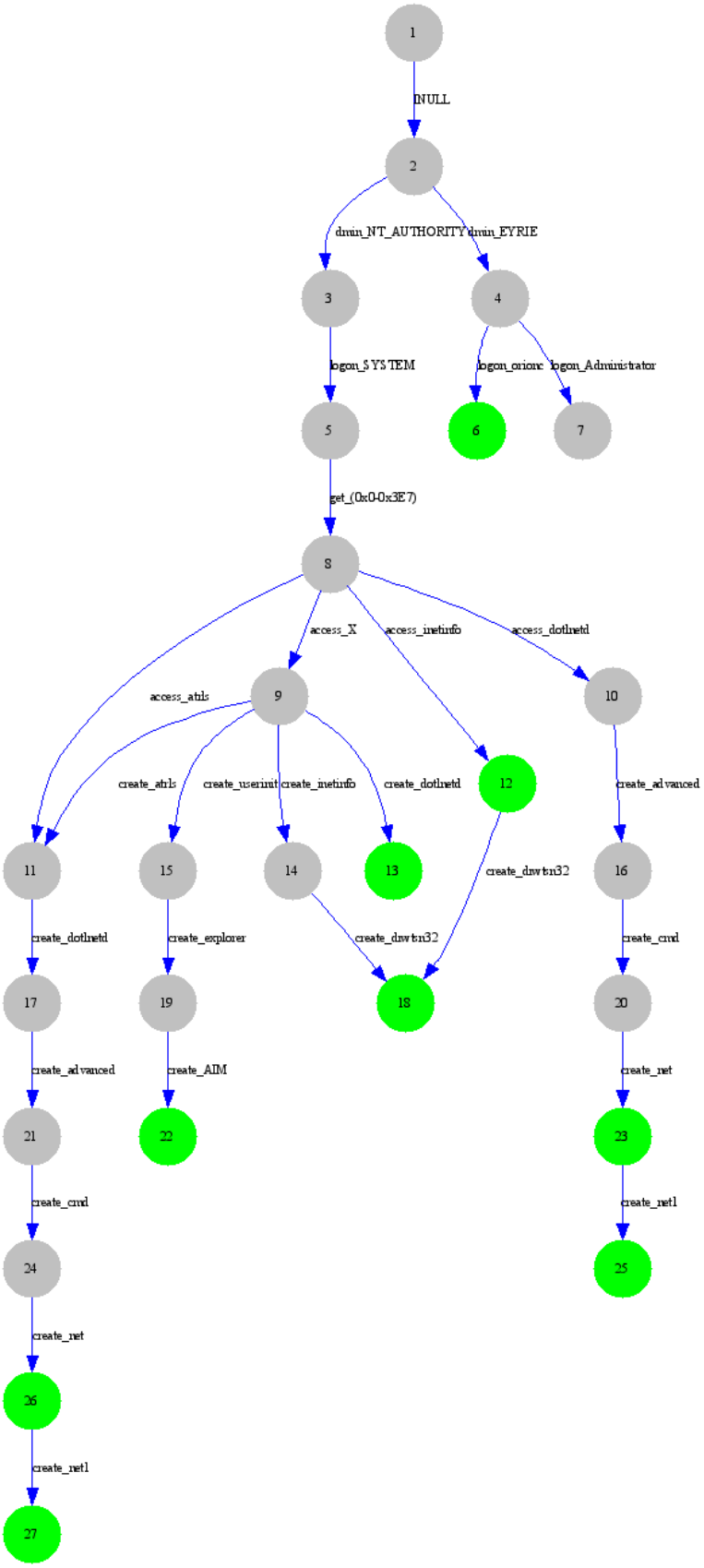


Fig. 2. The ALFSA for the audit-log event 592: Create process built from the audit-event log Week-4-Tuesday.

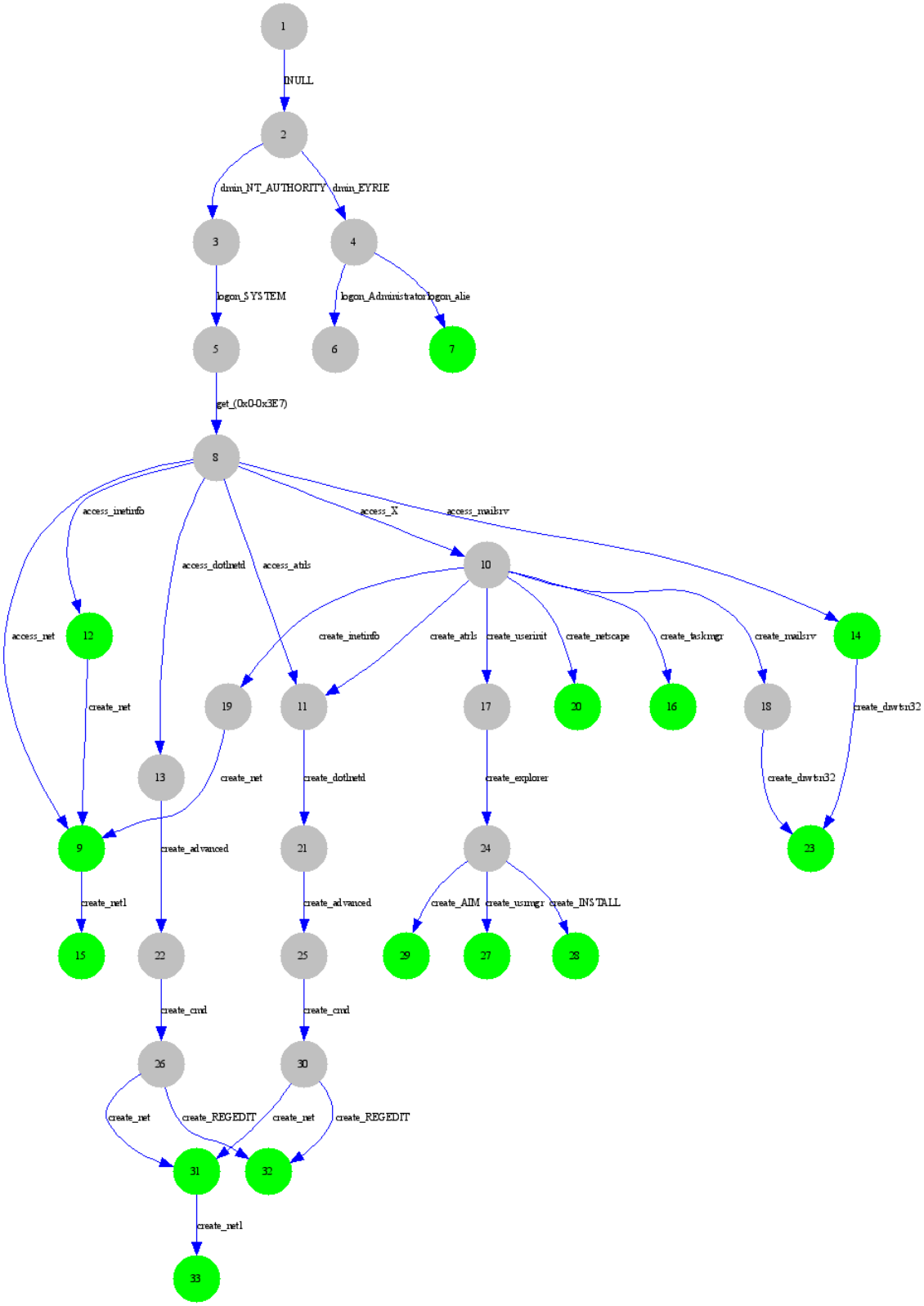


Fig. 3. The ALFSA for the audit-log event 592: Create process built from the audit-event log Week-4-Tuesday.

consider a method for partially-observed systems is as follows: A complete ALFSA built for *all* audit-log events may be very large and the administrator may choose to filter one or more audit-log events to focus his detective work on a selected set of events. Partial-observation methods help us understand the effect of filtering on pattern diagnosis. In this study, observable events are audit-log events that are recorded according to the audit policy or otherwise significant events such as *login/logout* and *create process*.

The problem of fault diagnosis for discrete-event systems has received considerable attention in the last decade and diagnosis methodologies based on the use of discrete-event models have been successfully used in a variety of technological systems ranging from document processing systems to intelligent transportation systems; see [11] and the references therein. The methodology of the Diagnoser Approach introduced in [12] is generalized (from diagnosis of a single event) to event patterns in [13], [14]. The contribution of [13], [14] is two-fold: 1. Off-line verification of the diagnosability property of the system with respect to the pattern, i.e., *if* the system is diagnosable with respect to the pattern. 2. On-line monitoring of the system and diagnosis of the pattern, i.e., *how* to detect the occurrence of the pattern while *partially* observing the behavior of the system.

In [13], two different notions of pattern diagnosability are defined in the context of formal languages: (i) S-type pattern diagnosability and (ii) T-type pattern diagnosability. These two different types stem from different approaches to defining the occurrence of a pattern. In S-type pattern diagnosability, a pattern is detected if all the sequences executed by the system that record the same observed event sequences contain *subsequences* in the pattern. In T-type pattern diagnosability, a pattern is detected if all the sequences executed by the system that record the same observed event sequences contain *substrings* in the pattern. In other words, there could be events interleaved between the events that make up the pattern in the S-type case, but not in the T-type case.

In order to diagnose a T-type (S-type) pattern s in a DES modeled by FSA with event set Σ, we first built FSA $H_{\mathcal{T}}(\Sigma, s)$ such that the projection of the language marked by the T-type (S-type) pattern FSA to the set of observable events contains the intrusion signature as a *substring* (*subsequence*). The necessary and sufficient condition for pattern diagnosability is based on another DES structure called *Observer*. The observer of FSA $G = (Q, \Sigma, \delta, q_0, F)$ is (see, e.g., [15] for further details)

$$\mathbf{Obs}(G) = (X, \Sigma_o, \delta_o, x_o), \tag{6}$$

where $x \in X$ is a set of states in Q,$\Sigma_o \subseteq \Sigma$ is the set of observable events, and x_0 is the initial observer state.

An observer state $x = \{q_1^x, \ldots, q_l^x\}$ is *marking-certain* if $q_i^x \in F$ for $i = 1, \ldots l$, and *marking-uncertain* if there exists $q_i^x \in F$ and $q_j^x \in Q \setminus F$ for some $i, j \in \{1, \ldots, l\}$. Let $\{x_1, \ldots, x_l\}$ and $\sigma_{o,1} \ldots \sigma_{o,l} \in \Sigma_o^*$ form a cycle in $\mathbf{Obs}(G)$. The cycle in $\mathbf{Obs}(G)$ is a *marking-indeterminate cycle* if the following are satisfied

1) x_i is marking-uncertain for $i = 1, \ldots, l$,
2) $\exists q_i^k,\ r_i^l \in x_i$ for all $i = 1, \ldots, m$, $k = 1, \ldots, M$, and $l = 1, \ldots, N$ such that
   a) q_i^k is marked and r_i^l is not marked for all i, k, l,
   b) there are two corresponding cycles in G[1]:

$$\begin{array}{l} q_1^1 \xrightarrow{\sigma_{o,1} t_1^1} q_2^1 \ \ldots \xrightarrow{\sigma_{o,m-1} t_{m-1}^1} q_m^1 \xrightarrow{\sigma_{o,m} t_m^1} \\ q_2^2 \ \ldots \xrightarrow{\sigma_{o,m-1} t_{m-1}^2} q_m^2 \ldots\ q_1^M \xrightarrow{\sigma_{o,1} t_1^M} q_2^M \\ \ldots \xrightarrow{\sigma_{o,m-1} t_{m-1}^M} q_m^M \xrightarrow{\sigma_{o,m} t_m^M} q_1^1 \end{array} \tag{7}$$

and

$$\begin{array}{l} r_1^1 \xrightarrow{\sigma_{o,1} u_1^1} r_2^1 \ \ldots \xrightarrow{\sigma_{o,m-1} u_{m-1}^1} r_m^1 \xrightarrow{\sigma_{o,m} u_m^1} \\ r_2^2 \ \ldots \xrightarrow{\sigma_{o,m-1} u_{m-1}^2} r_m^2 \ \ldots\ r_1^N \xrightarrow{\sigma_{o,1} u_1^N} r_2^N \\ \ldots \xrightarrow{\sigma_{o,m-1} u_{m-1}^N} r_m^N \xrightarrow{\sigma_{o,m} u_m^N} r_1^1 \end{array} \tag{8}$$

where $t_i^k,\ u_i^l \in \Sigma_{uo}^*$ for all i, k, l. $\square$

A union FSA $\mathbf{U}(G_1, G_2)$ of G_1 and G_2 is such that $\mathcal{L}(\mathbf{U}(G_1, G_2)) = \mathcal{L}(G_1) \cup \mathcal{L}(G_2)$ and $s \in \mathcal{L}_m(\mathbf{U}(G_1, G_2))$ if $s \in \mathcal{L}_m(G_1)$ or $s \in \mathcal{L}_m(G_2)$. The extension of the union of two FSA to more than two is a recursive operation.

The necessary and sufficient condition for T-type pattern diagnosability of a regular language with respect to a pattern is as follows:

Theorem 1 (T-type): A prefix-closed, live language $L = \mathcal{L}(G)$ is T-type pattern diagnosable with respect to a pattern K and projection P iff $\mathbf{U}_{s \in K}(G \times H_{\mathcal{T}}(\Sigma, s))$ does not contain any marking-indeterminate cycles.

The condition for S-type is the same except for the construction of the pattern FSA. Pattern diagnosis as defined in this section can be performed in polynomial time with respect to the number of states of ALFSA.

In the following, we present two examples where one attack is a T-type pattern and the other is S-type. The intrusion signature for the CrashIIS attack (studied in the previous section) is a T-type pattern because DRWTSN.EXE is immediately followed by *INETINFO.EXE*. The T-type pattern FSA $H_{\mathcal{T}}$ for CrashIIS is shown in Fig. 4. Suppose that all the audit-log events in the ALFSA are observable, then *CrashIIS* is correctly detected on Week-4-Tuesday, Week-5-Monday, and Week- 5-Tuesday, but not on Week-4-Monday and Week 5-Thursday when launched as part of *Yaga* attack and during the normal operation. If INETINFO.EXE is filtered out and considered as an unobservable event, then *CrashIIS* is wrongly detected on Week-1-Wednesday, Week-2-Thursday, and Week-2-Friday ALFSA that are collected during normal operation. This means that filtering out events in the intrusion signature may result in False Positive (FP) detection of the alarm.

We now consider an S-type intrusion signature. The Casesen attack separated into three separate event sequences and all three need to be diagnosed in the ALFSA. The *Casesen* attack does not require consecutive event occurrences, thus it is an

[1] $q \xrightarrow{s} q'$ denotes $q' = \delta(q, s)$ where q and q' are states and s is a string.

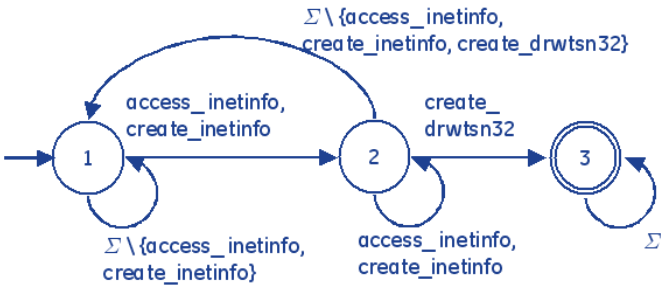


Fig. 4. CrashIIS T-type pattern FSA.

S-Type pattern. The S-type pattern FSA H_S for the Casesen attack is shown in Fig. 5.

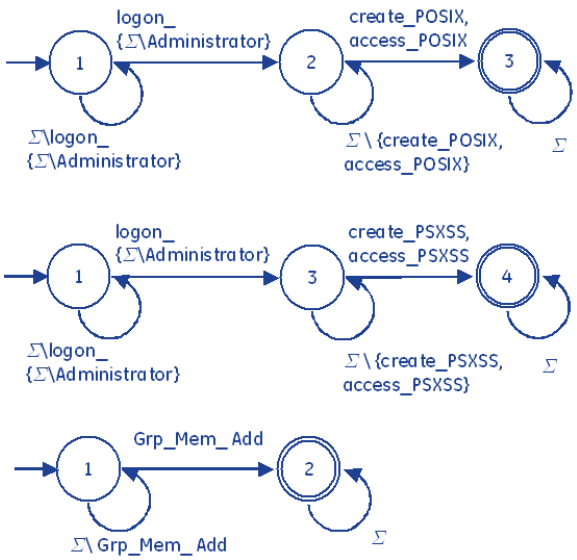


Fig. 5. FSA model of *Casesen* attack.

Suppose that all events are observable, then *Casesen* attack is correctly detected on Week-5-Tuesday and Week-5-Thursday, and there are no FPs for audit logs during normal operation. If all transitions created by *Logon/Logoff* audit-log events are unobservable, then the *Casesen* attack is still detected. There are no indeterminate cycles in Week-5-Tuesday ALFSA, so a sub-attack of *Casesen* is correctly detected. If we consider the very same unobservable events for Week-1-Wednesday, the occurrence of the same sub-attack is ambiguous, i.e., there is an indeterminate cycle. In this case, operator (or administrator) may either consider increasing the set of observable events gradually, act conservatively and declare intrusion, or otherwise declare no attack. The advantage is that if the pattern is diagnosable with a smaller set of observable events, then the operator can declare attack with certainity, and investigate more only if there is an indeterminate cycles. Thus, filtering out audit-log events may result in True Positive (TP) detection of the intrusion, however, may increase the number of FP detections.

VI. Conclusion

In this paper, we described an algorithm to build an FSA from audit-log files that reveals dependency relations between various OS-level events. The algorithm builds ALFSA by stitching smaller FSA built for each audit-log event to each other as it parses through the audit-log. Thus, our approach naturally combines the multiple sequences, unlike [4]. The length of the sequences are limited by the number of dependencies among the processes. This suggests that the length of the sequences are rather short compared to previous methods for construction of FSA.

The SISFSA built for two of the attacks studied in this paper revealed useful information in understanding and tracking the intruder's steps. Finally, we discussed an application of a pattern diagnosis method developed for detecting and isolating patterns that result in system failures in partially-observed FSA. The previous studies have not considered including the audit-logs collected during the same intrusion. The audit-logs collected during the same intrusion from multiple machines can be used in the method. Since the auditing program is universal (with the exception of level of auditing) over the Windows NT machines, there would be few consistency issues.

Partial-observation (and methods that allow the study of partially-observed systems) can be used to reduce the audit-log event set to fewer significant events, after which the detective work is performed on the smaller set of events, at the expense of increasing FPR. A comprehensive study of all attacks in the 1999 evaluation would be beneficial in providing more evidence on the strength of the approach. Also, extending of the study to contemporary attacks and technology would prove very useful in understanding the strengths and limitations of the techniques and methods described here.

References

[1] J. Korba, "Windows nt attacks for the evaluation of intrusion detection systems. Master of Engineering. Massachusetts Institute of Technology, Electrical Engineering and Computer Science," Jun. 2000.

[2] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *Journal of Computer Security*, vol. 6, no. 3, pp. 151–180, 1998. [Online]. Available: citeseer.ist.psu.edu/article/hofmeyr98intrusion.html

[3] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni, "A fast automaton-based method for detecting anomalous program behaviors," 2001, pp. 144–155. [Online]. Available: citeseer.ist.psu.edu/sekar01fast.html

[4] K. Wee and B. Moon, "Automatic generation of finite state automata for detecting intrusions using system call sequences," in *Lecture Notes in Computer Science: Computer Network Security*, vol. 2776/2003. Heidelberg, Germany: Springer Berlin, 2004.

[5] R. F. Smith, "Interpreting the NT security log," *Windows & .NET Magazine*, April 2000.

[6] S. T. King and P. M. Chen, "Backtracking intrusions." *ACM Trans. Comput. Syst.*, vol. 23, no. 1, pp. 51–76, Feb. 2005.

[7] Microsoft. Windows 2000 security event descriptions. [Online]. Available: http://support.microsoft.com/?kbid=299475

[8] IST-MIT-LL. Data sets. [Online]. Available: http://www.ll.mit.edu/IST/ideval/data/data_index.html

[9] FSProLabs. Event log expoler. [Online]. Available: http://www.eventlogxp.com/

[10] UMDES. Umdes software library. [Online]. Available: http://www.eecs.umich.edu/umdes/toolboxes.html

[11] S. Lafortune, D. Teneketzis, M. Sampath, R. Sengupta, and K. Sinnamohideen, "Failure diagnosis of dynamic systems: An approach based on discrete event systems," in *Proc. 2001 American Control Conf.*, Jun. 2001, pp. 2058–2071.

[12] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using discrete event models," *IEEE Trans. Control Systems Technology*, vol. 4, no. 2, pp. 105–124, Mar. 1996.

[13] S. Genc, "On diagnosis and predictability of partially-observed discrete-event systems. Ph.D. Thesis. University of Michigan, Electrical Engineering: Systems," Apr. 2006.

[14] T. Jeron, H. Marchand, S. Pinchinat, and M. Cordier, "Supervision patterns in discrete event systems diagnosis," in *Proceedings of the 8th International Workshop on Discrete-Event Systems*, 2006.

[15] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

AUTHOR BIOGRAPHIES

Fabio R. Auffant II
Technical Lieutenant, New York State Police, Computer Crime Unit
FAuffant@troopers.state.ny.us

Fabio R. Auffant II has been employed by the New York State Police for the past 22 years. He is a Technical Lieutenant in the Computer Crime Unit of the Bureau of Criminal Investigation. T/Lt. Auffant is the Manager of the Computer Forensic Laboratory located in the Forensic Investigation Center in Albany, NY, and Coordinator of the field digital forensics for the Computer Crime Unit. He has received extensive computer forensics training, holds several certifications in Computer and Digital Forensics and has conducted forensic examinations and data analysis in hundreds of investigations. T/Lt. Auffant has testified extensively throughout the State of New York in Grand Jury, Suppression Hearings and criminal trial proceedings. T/Lt. Auffant is a member of several professional organizations such as, the High Technology Crime Investigation Association (HTCIA), the International Association of Computer Investigative Specialists (IACIS), the High Tech Crime Network (HTCN), the Institute of Computer Forensic Professionals (ICFP), and InfraGard Albany Chapter executive committee. He has provided training and lectured extensively to government and law enforcement personnel in the field of Digital and Computer Forensics, as well as academic institutions such as John Jay College Criminal Justice School, University of Albany Business School and NY Prosecutors Training Institute Summer College at Syracuse University. T/Lt. Auffant is an adjunct professor at Columbia Greene Community College in the Computer Security and Forensics degree program.

Cristian Balan
Program Director, Computer and Digital Forensics Program, Champlain College
balan@champlain.edu

Professor Cristian Balan is the Program Director of Computer and Digital Forensics Program at Champlain College in Burlington, VT. He has extensive consulting experience working with the law enforcement community on both the system administration and information security. He is an active member of the Burlington Chapter of Infragard and is the Chief of the Vermont Army National Guard Computer Network Defense Team. CPT Balan is a National Guardsman with 25 years of experience with the last 8 years spent in the Information Assurance field. CPT Balan holds DOD Certification in IA Level III both technical and management. Professor Balan holds the CISSP Certification from the International Information Systems Security Certification Consortium, Inc. [(ISC)²]. Along with teaching Digital Forensics courses, Professor Balan has spent the past year developing a hands-on course to teach information security concepts for junior undergraduate students. The course relies heavily on the installation, configuration and add-ons for the IPCop Open Source product.

David J. Boyd
Information Security Group, Royal Holloway, University of London
D.Boyd@rhul.ac.uk

David Boyd is a research student with the Information Security Group at the Royal Holloway College, University of London, UK. His research interests include the non-repudiation and dispute resolution mechanisms for electronic payment systems that are used by consumers both over the Internet and in public places. Previously, David was an information security advisor and an information systems auditor for a large multi-national oil company and left in 2006 after 32 years service.

Madhusudhanan Chandrasekaran
State University of New York at Buffalo
mc79@cse.buffalo.edu

Madhusudhanan is a fifth year Ph.D. candidate. He is currently doing research in the field of computer security, with particular focus on malware defense, anti-spyware, intrusion forensics, anti-phishing. He received an M.S. in Computer Science and Engineering in 2004. He also served as an ads backend engineering and as a member of their payment fraud team and was working on security issues related to checkout at Google.

Vineet Chaoji
Department of Computer Science, Rensselaer Polytechnic Institute
chaojv@cs.rpi.edu

Vineet is a fourth year Ph.D. student in the Computer Science department of Rensselaer Polytechnic Institute. He also has a M.S. in Computer Science form the Rochester Institute of Technology. Within the last year, he has worked at the Center for Software Excellence in Microsoft Corporation by building a framework for analyzing failure patterns of over 10,000 machines through log analysis and applying itemset and sequence mining techniques to infer probabilistic rules defining failure conditions. He has research interests in pattern mining algorithms, specifically, properties (formal concepts) and applications (bioinformatics, text mining, etc.) of graph and sequence patterns. In the past, he has also worked on text classification, social network analysis, and link prediction problems.

John Crain
ICANN (Internet Corporation for Assigned Names and Numbers)
http://www.icann.org/

John Crain is the Chief Technical Officer of ICANN. ICANN is responsible for coordinating domain names, addresses, and other unique Internet identifiers. John has been contributing his technical expertise and leadership to ICANN for over three years. In this role, he has taken on the task of enhancing and improving the professional technical operations of ICANN and will lead the organization's root management, website development and information services functions. John enjoys an excellent and respected reputation with the global technical community. He conducts much of ICANN's liaison work with the global technical community in order to facilitate and listen to discussion on the many technical-based issues facing ICANN. John, a native of the United Kingdom and fluent in Dutch and English, spent a significant portion of his professional career working in the Netherlands for the RIPE NCC as part of the senior management team and was responsible for infrastructure, software development and operations. Prior to RIPE NCC, John worked as a design engineer in research, design and development of advanced materials.

Gurpreet Dhillon
Virginia Commonwealth University
gdhillon@vcu.edu

Dr. Gurpreet Dhillon is Professor of Information Systems in the School of Business, Virginia Commonwealth University, Richmond, USA. He holds a Ph.D. from the London School of Economics and Political Science, UK. His research interests include management of information security, ethical

and legal implications of information technology. Gurpreet has published over 100 research manuscripts in some of the leading journals in the field. He has also authored six books including Principles of Information Systems Security: text and cases (John Wiley, 2007). He is also the Editor-in-Chief of the Journal of Information System Security.

Sahika Genc
Sensor Informatics Technology Laboratory, General Electric Global Research Center
gencs@ge.com

Sahika is an electrical engineer in the Sensor Informatics Technology Laboratory at the General Electric Global Research Center. Her research interests are in monolithic and modular (distributed) algorithms for fault isolation and detection and alarm management in Discrete-Event Systems (DES), modeling and performance analysis communication networks and emerging behaviors. She received her M.S. and Ph.D. degree in Electrical Engineering: Systems from the University of Michigan, Ann Arbor, Michigan in 2002 and 2006, respectively.

Brendan J. Gilbert
School of Management/Law School, State University of New York at Buffalo
bg1@buffalo.edu

Brendan is currently pursuing a law degree and an MBA from the State University of New York at Buffalo (expected in 2010). He graduated with a B.A. in Computer Science in 2006 and has worked as an Information Risk Management Associate in KPMB within the last year.

Manish Gupta
M&T Bank Corporation, Buffalo, NY
mgupta3@buffalo.edu

Manish is currently an executive in M&T Bank Corporation, Buffalo, NY, USA and also adjunct instructor/professor (Fall 2007) at State University of New York at Buffalo. He received his bachelor's degree in mechanical engineering from Institute of Engineering and Technology, Lucknow, India in 1998 and an MBA in Information Systems from State University of New York, Buffalo in 2003. He is also a Ph.D. candidate at State University of New York, Buffalo. With more than a decade of experience in information systems, policies and technologies, he has published 3 books in the area of information security, ethics and assurance including Managing information Assurance in Financial Services (publisher: Idea Group Inc.). He serves in editorial boards of International Journal of Electronic banking and International Journal of Liability and Scientific Enquiry (IJLSE) and has served in program committees of several international conferences. He holds several professional designations including CISSP, CISA, CISM, ISSPCS and PMP. He has also received advanced certificates in information assurance (SUNY, Buffalo), IT Benchmarking (Stanford University) and cyber law (Asian School of Cyber Law).

Apirak Hoonlor
Department of Computer Science, Rensselaer Polytechnic Institute
hoonla@rpi.edu

Apirak is a graduate student in the Computer Science department in Rensselaer Polytechnic Institute. In the past, his work has been published in Bioinformatics, one of the leading journals in the bioinformatics field.

Daniel R. Kerr
SORCER Laboratory, Texas Tech University
daniel.robert.kerr@gmail.com

Daniel is a graduate computer science student at Texas Tech University expecting to graduate in May 2008. He is currently the Project Manager/Lead Programmer/Designer of the Application for Space Grid Computing Project and also serves as a software intern at Hewlett Packard in Richardson, TX taking part in development of web applications for Hewlett Packard Real Time Management System, support of Experience Center project development and operations, and designing and testing of software solutions.

Kenneth P. Mortensen, Esq.
U.S. Department of Justice
kenneth.mortensen@usdoj.gov

Kenneth P. Mortensen is the Acting Chief Privacy and Civil Liberties Officer for the U.S. Department of Justice. As the former Deputy Chief Privacy Officer of the Department of Homeland Security and a practicing privacy attorney, Mr. Mortensen brings expertise not only in protecting and safeguarding privacy and civil librerties, but also integrating those protections and safeguards into an operational framework for law enforcement and national security. Within the Office of the Deputy Attorney General, he determines appropriate privacy processes collaborating in the development of policy supporting the mission of the Department. As the Acting Chief Privacy and Civil Liberties Officer, he serves as the primary policy advisor to the Attorney General and Deputy Attorney General on privacy and civil liberties matters concerning agency operations. In this role, Mr. Mortensen oversees the implementation privacy and civil liberties policy throuthout the Department, including the constituent bureaus, such as the Federal Bureau of Investigations, the Drug Enforcement Aagency, and the Office of Justice Programs, and agency components, such as the Criminal Law Division, Justice Management Division, and Civil Law Division.

Terri Oda
School of Computer Science, Carleton University
toda@scs.carleton.ca

Terri is a PhD student in the School of Computer Science at Carleton University. Her Master's work (Carleton, 2005) involved using the human immune system as a model for detecting junk email (spam). Her current research interests include computer security, particularly web security, usability, and artificial life.

H. Raghav Rao
School of Management, State University of New York at Buffalo
mgmtrao@buffalo.edu

Dr. Rao has a Ph.D from Purdue University, an M.B.A from Delhi University, and a B.Tech. from the Indian Institute of Technology. His interests are in the areas of management information systems,

decision support systems, and expert systems and information assurance. He has chaired sessions at international conferences and presented numerous papers. He has authored or co-authored more than 100 technical papers, of which more than 60 are published in archival journals. His work has received best paper and best paper runner up awards at AMCIS and ICIS. Dr. Rao has received funding for his research from the National Science Foundation, the Department of Defense and the Canadian Embassy and he has received the University's prestigious Teaching Fellowship. He has also received the Fulbright fellowship in 2004. He is a co-editor of a special issue of The Annals of Operations Research, the Communications of ACM, associate editor of Decision Support Systems, Information Systems Research, and IEEE Transactions in Systems, Man and Cybernetics, and co-Editor- in -Chief of Information Systems Frontiers.

Daniel O. Rice
The Sellinger School of Business and Management, Loyola College in Maryland
drice2@loyola.edu

Daniel is currently an Assistant Professor of Information Systems in Loyola College in Maryland. He received his Ph.D. in Information Systems and a MBA in Finance from the University of Connecticut. He has research interests in the following areas: computer and information security economics, data confidentiality and privacy, network security, e-commerce, electronic markets, and complex/social networks. His current projects include security of peer-to-peer (P2P) networks and the measurement of differential value of privacy.

Billy (BK) Rios
Microsoft Corporation
http://xs-sniper.com/blog/

Billy is currently a Security Engineer for Microsoft, helping secure software used by millions of people across the world. Before his current role as a Security Engineer, Billy was a Senior Security Consultant for VeriSign, where he broke into the information systems of various clients in the Fortune 500 and helped them understand existing and emerging security risks. Before his life as a consultant, Billy helped defend Department of Defense networks as an Information Assurance Analyst. He looked at packets, monitored for suspicious network activity, took apart malicious code, and formally reported network and security incidents of all shapes and sizes. Before attacking and defending networks, he was an active duty Marine Officer (Semper Fi!). Billy spent some time in a hot desert, carried a side arm (sometimes a machine gun), and got real up-close and personal with physical and operational security… Billy has an undergraduate degree in Business (with a formal concentration in Information Systems) from the University of Washington and a M.S. Degree in Information Systems (with Distinction) from Hawaii Pacific University. Billy is currently pursing his MBA at Texas A&M (Commerce).

Vidyaraman Sankaranarayanan
School of Management, State University of New York at Buffalo
vs28@cse.buffalo.edu

Vidyaraman is a doctoral student at the State University of New York at Buffalo and graduated with degrees from the University at Buffalo and the University of Kansas. He has previous publications in the areas of trust modeling, security policies, simulation, document management, and game theory. He also serves as the Workshop Planning & Management Chair at the Second Workshop on Intelligent

Networks: Adaptation, Communication & Reconfiguration (IAMCOM 2008) and is on the program committee at the New Security Paradigms Workshop 2008.

Raj Sharman
School of Management, State University of New York at Buffalo
rsharman@buffalo.edu

Dr. Raj Sharman is an assistant professor in the School of Management at the State University of New York, Buffalo, New York. He received his Bachelors degree in Engineering and Masters Degree in Management from the Indian Institute of Technology, Bombay, India. He also received a Masters in Industrial Engineering, and a Doctoral degree in Computer Science from Louisiana State University. His research interests are in the areas of Information Assurance, Disaster Management, and Internet Technologies. He is a recipient of several grants, both internal and external grants in the area of Information Security. His publications appear in peer reviewed journals and international conferences in both the Information Systems and the Computer Science disciplines. Dr. Sharman serves as an associate editor for the Journal of Information Systems Security.

Sean Smith
Technical Resource Attorney, New York Prosecutors Training Institute
Sean.Smith@nypti.org

Since 1997 Mr. Smith has been an attorney with the New York Prosecutors Training Institute in Albany, New York. In this capacity, Mr. Smith assists prosecutors with issues arising in felony cases, and assists prosecutors across the country by providing them with valuable information on expert witnesses. In addition, as the Technical Resource Attorney Mr. Smith has provided technical trial assistance in numerous high profile cases by both developing and presenting in-court multi-media presentations. Mr. Smith regularly consults with prosecutors from across New York on using technology to help present cases to juries.

Michael Sobolewski
Director, SORCER Laboratory
Computer Science, Texas Tech University
sobol@cs.ttu.edu

Dr. M. Sobolewski is a Professor of Computer Science at Texas Tech University since September 2002. He teaches courses related to distributed computing: network security, advanced network programming, P2P, and mobile computing. He is a director of the SORCER laboratory at Computer Science Department, TTU. The laboratory research is focused on service-oriented computing systems. Before, he worked with General Electric Global Research Center as a Senior Computer Scientist since August 1994. From 1999 he has worked on service-grid computing systems and developed a service-based programming methodology for the FIPER/NIST (Federated Intelligent Product Environment) project. While at GE GRC, he was a FIPER chief architect and lead developer. In the period of 1997-2000 he lead and developed web-based computing framework (GApp/DARPA) and demonstrated 17 successful applications for various GE businesses including a document management system for the family of F110 engines – GE Aircraft Engines, a Web-EMPIS system - GE engineering specification system, an Engineering Calculator - GE Plastics. He led GE's successful CAMnet/DARPA project (1995-1996), developing tools and methodology to deliver manufacturing and engineering services via the World Wide Web. Also, in 1996 he led a successful Lockheed Martin EDN Toolkit project that provides

enablers to built Web-based workbooks and record books. From November 1989 until February 1994 he was invited to work on DICE program at Concurrent Engineering Center (CERC), West Virginia University, where he developed a knowledge-based environment for concurrent engineering (DICEtalk) based on his novel percept knowledge representation scheme, a Motif-based generic application, a GUI client for information sharing system, and a GUI interface for medical informatics system (ARTEMIS).

Anil Somayaji
Director, Carleton Computer Security Lab
Computer Science, Carleton University
soma@ccsl.carleton.ca

Anil is an Assistant Professor in the School of Computer Science at Carleton University. He received a B.S. (1994) degree in mathematics from the Massachusetts Institute of Technology and the Ph.D. (2002) degree in computer science from the University of New Mexico. He has served on the program committees of the USENIX Security Symposium and the New Security Paradigms Workshop, among others. His research interests include computer security, operating systems, complex adaptive systems, and artificial life.

Boleslaw K. Szymanski
Director, Rensselaer Center for Pervasive Computer and Networking
Computer Science, Rensselaer Polytechnic Institute
szymansk@cs.rpi.edu

Boleslaw received his Ph.D. in Computer Science from the National Academy of Sciences and his Masters in Engineering (Electronics) from Warsaw Polytechnic University in Warsaw, Poland. He is currently a Claire and Roland Schmitt Distinguished Professor in Computer Science and the Rensselaer Polytechnic Institute. He is also a founding Director of the Center for Pervasive Computing and Networking. His research interests include wireless and sensor networks, analysis and design of distributed and parallel algorithms, simulation of computers, networks and biological/ecological phenomena. His projects include sensor network protocols and algorithms, data fusion, large-scale parallel and distributed computing, simulation. Dr. Szymanski led the development of COST and DSIM, two innovative simulation systems, SENSE for sensor network simulation, GENESIS for real-time network management and VOGUE for innovative HMM models.

Ifeoma Udeh
Virginia Commonwealth University
udehia@vcu.edu

Ifeoma is currently pursuing her doctoral studies at the School of Business, Virginia Commonwealth University, Richmond, USA. Ifeoma has extensive experience in undertaking internal audits at various Fortune 500 companies. She holds an MBA with concentration in Accounting from Virginia Commonwealth University.

Shambhu Upadhyaya
Director, Center of Excellence in Information Systems Assurance Research and Education
Computer Science and Engineering, State University of New York at Buffalo
shambhu@cse.buffalo.edu

Dr. Shambhu J. Upadhyaya is an Associate Professor of Computer Science and Engineering at the State University of New York at Buffalo where he directs the Center of Excellence in Information Systems Assurance Research and Education (CEISARE), designated by the National Security Agency. Prior to July 1998, he was a faculty member at the Electrical and Computer Engineering department. His research interests are information assurance, computer security, fault diagnosis, fault tolerant computing, and VLSI Testing. He has authored or coauthored more than 150 articles in refereed journals and conferences in these areas. His current projects involve insider threat modeling, intrusion detection, security in wireless networks, and protection against Internet attacks. His research has been supported by the National Science Foundation, Rome Laboratory, the U.S. Air Force Office of Scientific Research, National Security Agency, IBM, and Cisco.

Tony White
School of Computer Science, Carleton University
arpwhite@rogers.com

Tony White is an Associate Professor in the School of Computer Science at Carleton University. He received his M.A. (1981) in theoretical physics from Cambridge University and the Ph.D. (2001) degree in electrical engineering from Carleton University. He has served on several program committee, most notably GECCO. His research interests include self-organizing systems, distributed computing and web-based information systems.

George Wright
The Sellinger School of Business and Management, Loyola College in Maryland
geo@loyola.edu

George is an Associate Professor of Information Systems in Loyola College in Maryland. He has a D.B.A. in Infomration Technology and an MBA in Health Information Systems from George Washington University. He also has a B.S. in physics and mathematics form the U.S. Naval Academy. He has experience in federal government, trade association, and consulting positions. His current research areas include Peer-to-Peer (P2P) network topology and simulation as well as network security.

INDEX OF AUTHORS