REPORT DOCUMENTATION PAGE

AFRL-SR-AR-TR-04-

0137

The public reporting burden for this collection of information is estimated to average 1 hour per response, including gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments of information, including suggestions for reducing the burden, to Department of Defense, Washington Headqu (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be awa subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
	Final	1 Dec 99 - 31 Aug 03

4. TITLE AND SUBTITLE
Code Hiding Techniques for Mobile Applications

5a. CONTRACT NUMBER

5b. GRANT NUMBER
F49620-00-1-0063

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Rida A. Bazzi
K. Selcuk Candan

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Arizona State University
P. O. Box 873503
Tempe, aZ 85287-3503

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
4015 Wilson Blvd., Room 713
Arlington, VA 22203-1954

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

20040305 024

14. ABSTRACT
The main goal of this effort was to enable lightweight clients(mobile devices) to outsource computations without divulging any information about the data and or code of the application.This problem is known to be very difficult. In order to make progress, we had to attempt to solve the problem under some restrictions on the types of programs. We considered programs that have loops and programs that do not have loops. For both types of programs we restricted our study on programs that have linear assignments and conditionals. We also considered the problem of hiding the access patterns to an encrypted database. Summary or Results:Provable data hiding and tamper resistance for a simple loop program: we developed techniques that enable provable data hiding for a simple loop program. Data and Code Hiding for non-iterative piece-wise linear programs: we developed techniques that enable data and code hiding for non-iterative piece-wise linear programs, but we were unable to fully quantify the achieved degree of hiding. Data hiding for piece-wise linear programs with linear conditionals: we studied Conditions under which hiding of traversal of tree-structured data.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Rida A. Bazzi
					19b. TELEPHONE NUMBER *(Include area code)*

Final Technical Report
Code Hiding Techniques for Mobile Applications
Award Number F49620-00-1-0063

Senior Personnel

- Rida A. Bazzi (Principal Investigator)
- K. Selcuk Candan (co-PI)

Students

- Ping Lin: doctoral student
- Zhichao Liu: doctoral student
- Feras Karablieh: doctoral student
- Aziz Fajri: visiting student (Summer 2000)
- Raphael Badin: visiting student (Summer 2000)

1. Detailed Description of Work

1.1. Introduction

The main goal of this effort was to enable lightweight clients (mobile devices) to outsource computations without divulging any information about the data and or code of the application. This problem is known to be very difficult. In order to make progress, we had to attempt to solve the problem under some restrictions on the types of programs. We considered programs that have loops and programs that do not have loops. For both types of programs we restricted our study on programs that have linear assignments and conditionals. We also considered the problem of hiding the access patterns to an encrypted database.

1.2. Summary of Results

The following are the main results that we have achieved.

- Provable data hiding and tamper resistance for a simple loop program: we developed techniques that enable provable data hiding for a simple loop program.
- Data and Code Hiding for non-iterative piece-wise linear programs: we developed techniques that enable data and code hiding for non-iterative piece-wise linear programs, but we were unable to fully quantify the achieved degree of hiding.
- Data hiding for piece-wise linear programs with linear conditionals: we studied conditions under which
- Hiding of Traversal of tree-structured data.

In the following sections, we describe these results in details.

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

1.3. Code Hiding for a Simple Loop Program

We proposed a non-interactive scheme to provide data hiding for a program of the form

for l **do** $X = MX$,

where X is a complex input vector over C_n, l is an variable integer input, and M is a n x n complex matrix (publication [4]). The scheme allows a client to encrypt X without leaking any information about it to the server. Unlike other proposed schemes, our scheme allows a client to determine efficiently and with high probability whether the results returned by the server are correct or not. The scheme is non-interactive; the client sends only one message to the server and the server sends only one message to the client. This is the first hiding scheme that we are aware of that apply to real and complex inputs. Other hiding schemes typically assume that the inputs are elements of Z/mZ , (integers modulo m, where m is a prime number) which reduces their applicability. While our results apply to a simple class of loop programs, the techniques we develop are interesting on their own. The encryption cost of the scheme is of the order n^2 and is independent of l. The execution time of the program is ln^2 or $T(n)\lg l$, where $T(n)$ is the execution time of an algorithm to square the matrix M. The time ln^2 is obtained by applying a straightforward execution of l iteration each requiring n_2 operations. The time $T(n)\lg l$ is obtained by calculating M^l and then multiplying the result with X. If l is large, in both cases there are large saving in computation for the client. It should be clear that a simple way to hide the input for our program is to have the server calculate M_l and send the result to the client. This has two drawbacks. First, the communication complexity is n_2 to send a matrix instead of n to send a vector. Second, it is not clear in that case how the checking for correctness can be achieved.

Another interesting problem that can be solved with our scheme is that of joint computation between the client and server. In that problem, there is a function f with two inputs that the client and server want to calculate. One input is provided by the client and one input is provided by the server. The calculation should be done is such a way that the client and server learn nothing about the other's input except what they can deduce from the computed value. In our setting, if l is provided by the server, then the client and server to compute $f(X, l) = M^l$ without leaking any information the client or server other than what they can deduce from the value of $M^l X$.

1.4. Hiding Queries

With the increasing use of web services, many new challenges concerning data security become critical in various application domains. Especially in mobile computing, where clients are generally thin and have limited computation power or memory, a server (or an oracle) is needed to do the computation or store information for clients. Generally, this server or server farm has very strong computation capability and large memory space.

Unfortunately, such an oracle may not always be trustworthy and clients with sensitive data may want to be protected from malicious oracles. In our work (publication[3]), we developed protocols where data of clients are stored in a server's database in a way that not only the data, but also properties of the data, and the data structures are hidden from the server, while clients can still efficiently retrieve data based on available data structures. We show that this task is achievable with a limited communication overhead, which is essential in low bandwidth (such as wireless) distributed systems. This protocol makes possible clients to outsource their sensitive data on servers without any prior trust. In this protocol, tree nodes are encrypted before being outsourced; hence, their contents (if the nodes are XML elements, also the element types) are hidden from the untrusted data store. Each time a user wants to retrieve a node, it asks for a set of nodes called the "redundancy set" including the target node and additional random nodes. The redundancy set hides the requested node from the server. To prevent the server to infer the locations of the nodes through repeated trials, after each node is retrieved, the node is swapped with a randomly chosen empty node. Thus, the node moves after each retrieval, making any correct guessing of its location temporary and of no permanent use. The size of the redundant set need not be large to hide long paths. Unlike the information theoretic private information retrieval schemes, the technique requires no replication of the database and the communication cost is adaptable and generally much less than the size of the database. Compared with general computationally private information retrieval schemes, our technique is much simpler and does not rely on any cryptographic assumptions except for the ones on which the underlying encryption schemes are built. The protocol requires legal clients to have access to keys and be able to perform encryption and decryption operations. Where encryption and decryption constitute heavy computation costs for clients with very limited computation power and memory, we suggest the use of assistant hardware, such as smartcards, to reduce the encryption/decryption execution costs as well as to securely disseminate keys. To enable multiple users to query a tree simultaneously, which is mandatory for an open and public data store, we also devised deadlock free concurrency control mechanisms. Further study of the security guarantees provided by this protocol requires modeling of user queries and interaction between the client and the server. At the high-level, customer retrieval of a redundancy set constitutes a call. Each query then can be represented as an ordered set of calls. Supposing that there is a transport layer security mechanism that hides the identity of each retrieval request, we can model database/application hosting server's view of data accesses as a stream of calls from unknown origins. The server might still be able infer the tree structure by observing the call stream, guessing which calls in the stream belong to a single query, guessing which queries are the same or similar, and then looking at the intersection of the redundant sets of the corresponding calls in each query. In order to measure the amount of security provided by the system, thus, it is necessary to understand the probability with which the server can use these intersections to break the security:

- **Step 1**: Given two windows of calls observed by the server, each with size w, what is the probability of finding two queries from that are the same.

- **Step 2**: Given two windows of calls observed by the server, each with size w, that are known to contain two identical queries, what is the probability for the server to identify the individual calls of these two queries.
- **Step 3**: Given the calls of two identical queries, what is the probability for the server to discover the query path.

Attacks by the server would rely on the intersection property described above. Intuitively, such attacks can be prevented by ensuring that intersections do not reveal much information. This is achieved by modifying the protocols such that the redundant sets intersect at multiple nodes and by inserting appropriate dummy/interfering calls which break the relationship between queries and intersecting calls.

1.5. Book Chapters

We wrote two book chapters as part of this effort (publications [1] and [5]). Both chapters survey different aspects of the state of the art in the field. Both chapter provide a good survey of existing techniques for protecting hosts in a distributed setting.

In the chapter by Bazzi and Karablieh, we examined the requirements for developing reliable E-commerce applications in large and open distributed systems. Our emphasis was on mobile agents technology. We identified general requirements and security requirements for developing E-commerce application. We also examined security threats that result from the use of agents. We considered both threats to the agents themselves from hosts on which they execute and threats to the hosts from agents executing on them. Threats to agents fall under the general topic of our grant and most of the results in that area are not developed enough for practical applications. Threats to hosts are a much better understood subject. We classified techniques for dealing with threats to hosts as either compile-time or runtime techniques. The techniques we surveyed include proof carrying codes, sand boxing, execution monitoring, code signing, state appraisal, and path histories. For protecting agents from malicious hosts, we considered clueless agents, partial result encapsulation, computing with encrypted functions, and code obfuscation. Finally, we considered fault-tolerance and service replication as a technique to protect agents against malicious hosts.

In the chapter by Lin and Candan, we provided an overview of typical security challenges faced by distributed application hosting services and surveyed the main security available mechanisms and the open challenges at the different tiers in the information management hierarchy. This chapter also discusses hiding access structure from a server hosting encrypted data. All the topics of the chapter fall under the general topic of our grant. The chapter also discusses access control and a variety of security policies.

1.6. Significance to Field

Our results are the only results that we are aware of that directly apply to imperative programs. Other results in this area apply to Boolean circuits or Boolean predicates. Working with imperative program is necessary if one hopes to have practical applications for the work. Also, our results are the only results that we are aware of that apply to real and complex data. Other results in the field apply to data inputs that are integers modulo some prime number.

1.7. Relevance to AF Mission

Enabling secure computation outsourcing will enable lightweight clients to outsource computations to servers that can be potentially compromised. While our results have limited applicability, they are an important step towards that goal.

1.7. Potential applications to AF and civilian application challenges

The main application is computation outsourcing. Our current results do not have direct practical applications.

1.8. Interactions/Transitions

- Professor Bazzi participated in a NSF CAREER panel and reviewed some security-related proposals (2000).

2. Publications

1. *Ping Lin* and *K. Selcuk Candan*. Data and Application Security for Distributed Application Hosting Services. To appear in *Information Security Policies and Actions in Modern Integrated Systems*, Carlo Bellettini and Maria Grazia Fugini Eds., Idea Group Inc., 2003.
2. *Ping Lin*, *K. Selcuk Candan*, *Rida Bazzi*, and *Zhichao Liu*. Hiding Data and Code Security for Application Hosting Infrastructure. Special poster session at the First NSF/NIJ Symposium on Intelligence and Security Informatics (ISI 2003), Tucson, Arizona, June 2003.
3. Ping Lin and K. Selcuk Candan. Hiding Traversal of Tree-Structured Data from Untrusted Data Stores. Special poster session at the First NSF/NIJ Symposium on Intelligence and Security Informatics (ISI 2003), Tucson, Arizona, June 2003.
4. Rida A. Bazzi, K. Selcuk Candan, Raphael Badin, and Aziz Fajri. Provably Secure Data Hiding and Tamper Resistance for a Simple Loop Program. In Proceedings of SPIE Aerosense Conference, Orlando, Florida, April 2003.
5. *Rida A. Bazzi* and *Feras Karablieh*. Development of reliable commercial applications in large and open distributed systems. In *Handbook of E-business*, Paul B. Lowry, J. Owen Cherrington, and Ronald R. Watson Eds., CRC Press, 2002.

6. Rida A. Bazzi and K. Selcuk Candan. Practical Code Hiding Technical report, Arizona State University, 1999.
7. Ping Lin, K.S.Candan, Rida A. Bazzi, Zhichao Liu. "Data and Code Privacy for Application Hosting Infrastructures. Submitted to Workshop on Issues in the Theory of Security (WITS'04)
8. *Ping Lin* and *K. Selcuk Candan*. Hiding Queries, Data, and Data Distribution from an Oracle. Submitted to The Second International Workshop on Security In Information Systems 2004.