INFORMATION AGENTS: AN ACTIVE SECURITY POLICY TO LIMIT IDENTITY THEFT AND ENSURE PRIVACY

by

Muhammad Saaligh Gabier

BCS(Honours), University of Windsor, 2001
BA, University of Windsor, 2001

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN THE SCHOOL
OF
COMPUTING SCIENCE

SIMON FRASER UNIVERSITY
Summer 2006

APPROVAL

Name:	**Muhammad Saaligh Gabier**
Degree:	**Master of Science**
Title of Project:	**Information Agents: An Active Security Policy to Limit Identity Theft and Ensure Privacy**

Examining Committee:

Chair: Dr. Kay Wiese
Chair

____

Dr. Robert Cameron, Senior Supervisor

____

Dr. Arthur Kirkpatrick, Supervisor

____

Dr. Richard Smith
External Examiner
Associate Professor of Communications
Simon Fraser University

Date Defended/Approved: Apr. 13/06

ABSTRACT

Identity theft is the serious problem in North America. Criminals assume another person's identity in order to fraudulently perform actions or crimes in another person's name. These actions range from opening chequing accounts to renting apartments. Identity thieves can even perform serious crimes and impersonate the victim when arrested. This project proposes an *active information policy* in which all organizations must actively inform individuals when that particular individual's personal information is created, edited, transferred, or received. To cope with the influx of information, each individual will have an Information Agent whose responsibility is to protect the privacy of said individual. An Information Agent will be a professional who provides information services that may be impossible for the general public to perform. Information Agents shall be responsible for maintaining and cataloguing information as it is received, ensuring that the information is accurate, and vigilantly detecting anomalies which would indicate criminal activity.

To Mom and Dad

“Society has to be reconstructed in order for it to reap the benefits of technology.”
-Emmanuel Mesthene

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my beautiful wife Noreen for her love, friendship, and support.

I would like to thank Dr Rob Cameron for his guidance and patience. I would not have been able to complete this project without his support.

I would like to thank Dr Ted Kirkpatrick for being my teacher, employer, supervisor, and friend.

I would like to thank my friends and colleagues in the department, the MSA, and Hamilton Hall. Thank you for reminding me to enjoy life outside of the lab.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EQUATIONS

1 INTRODUCTION

1.1 Overview

Identity theft is a serious problem in North America. Criminals assume another persons identity in order to fraudulently perform actions or crimes in another person's name. These actions range from opening checking accounts to renting apartments. Identity thieves can even perform serious crimes and impersonate the victim when arrested. Although there have been a number of advancements in network and database security, identity theft often occurs through very unsophisticated means, such as stealing one's mail. The problem is hard to combat because the public is unaware of who has their personal information.

This project proposes an *active information policy* in which all organizations must actively inform individuals when that particular individual's personal information is created, edited, transferred, or received. To cope with the influx of information, each individual will have an Information Agent whose responsibility is to protect the privacy of said individual. An Information Agent will have to be more than just a layer of software, but a professional who provides information services that may be impossible for the general public to perform. Information Agents shall be responsible for maintaining and cataloguing information as it is received, ensuring that the information is accurate, and vigilantly detecting anomalies which would indicate criminal activity. The technical viability of such a system is investigated in this project. This is achieved by implementing a tool which acts as an intermediary between external organizations sending information updates and individuals who receive them.

1.2 The Vault Analogy

Many researchers are trying to prevent thieves from stealing information, but there is not nearly the same effort to deal with the problem after one's identity has already been stolen. For instance, departments across the world teach and research encryption techniques, but no matter how powerful our encryption techniques, identity theft will still occur.

It is useful to look at a bank's vault as an analogy. We can imagine that there are three basic ways a bank can protect its money:

1. construct a powerful vault with reinforced steel and a nearly impregnable lock,
2. rig the money so that it is valuable when it is in the vault or spent legally, but loses value if it is stolen (this can be achieved by marking the bills or recording the serial numbers),
3. eliminate the need and desire to steal the money

It can be argued that our personal information is as important as our wealth because it is our identity that is the means by which we gain wealth. Put another way, all of our accomplishments, such as academic training and work experience would not benefit us if it was not associated with us through our identity. We go to great lengths to create a figurative vault for our information. We use encryption when we send personal information over the Internet, we lock the entrances to our homes, etc. This is the first level of protection.

This project will attempt to deal with the second level of protection, that is taking away the value of stealing another person's identity. For instance, if a thief tries to open a credit card account in the victim's name, with an active information policy, the victim will immediately be notified. This makes the ill-gotten credit card less valuable because there is a smaller window of opportunity to fraudulently make purchases. Identity thieves rely on the fact that it can take months, even years for the authorities to catch up with their crimes. In this way, the information is “marked”.

Obviously, the ideal solution to identity theft (or any type of theft) would be to remove the desire and need to steal, but that is beyond the scope of this project.

1.3 Project Outline

This chapter gives an overview of the Information Agents proposal and the motivation of this project. Chapter two discusses the problem of identity theft. Chapter three describes related work including encryption; Internet Security; and Information Flow Control, a novel approach to controlling the propagation of information. Chapter four reviews technologies necessary for the development of Information Agent tools. Chapter five describes the role of Information Agents and the tools necessary to perform their tasks. The project concludes with a summary and some future directions for this work. The appendix describes a prototype of the Information Agent Tool.

2 PRIVACY

2.1 Introduction

Identity theft is the fastest growing crime in North America. Thousands of companies have enormous databases containing our personal information. Our personal information is being sold to other companies and is used to advertise to us. Our information has become quite a valuable commodity. There are thousands of companies that are dedicated to obtaining, compiling, and selling our information for millions of dollars [19]. Some companies even advertise their ability to obtain illegal information, such as our medical records or cell phone bills [19, 35]. Because all of this information is decentralized, it is the responsibility of the organization to keep the information secure and accurate. As you can imagine, when thousands of companies have records containing millions of pieces of information, the quality of the information is suspect. In fact, criminals take advantage of the fact that these companies can't possibly adequately verify the correctness of said information. Criminals can impersonate individuals because they are aware that there is no verification process that will expose them as impostors. Criminals can open chequing accounts, credit card accounts, or even commit crimes in someone else's name.

The problem of identity theft is exacerbated by the following two facts:

- Thousands of companies have large databases that contain our personal information, but there is not a uniform verification process to ensure that the information is correct,

- There is no adequate verification process to ensure that a customer is who he/she claims to be, and in North America, information can be obtained and sold with very little regulation.

There have been important strides towards treating privacy as a basic human right. The European Union Privacy Directive is a set of laws that require member states of the EU to respect the privacy of its customers by only using the obtained information for the purposes stated. Companies can also only do business with other companies that adhere to these privacy standards. This means that when a person submits information to an organization, the organization must make the customer aware of what the information will be used for. Customers also have the right to request all of their personal information.

2.2 Identity Theft

> The more we rely on automated systems to provide essential services in our daily lives, the more we must recognize the vulnerability of these systems and the importance of taking appropriate security measures [39].

Identity theft is the act of using someone's personal information, such as their social security number or credit card number, to fraudulently perform actions or crimes in said person's name. Criminals can register accounts to credit cards, phone lines, cable lines, etc. Because of these actions/crimes, a person who had their identity stolen can be refused loans for things such as higher education, homes, or cars. They may also be arrested for crimes they did not commit and have to prove that they did not commit these crimes which can take a number of years.

2.2.1 Methods of Identity Theft

When it comes to getting our personal information, thieves can be extremely resourceful. We often envision computer "hackers" stealing information from secure computer databases or tapping a signal as information is transmitted, but the truth is, thieves can steal our personal information through ingenious non-technical means. Thieves can impersonate landlords, government officials, or legitimate business men. They may steal our mail (bills, cheques, credit information), apply for pre-approved credit cards, or fill out the change of address form attached with every bill to divert our mail. Thieves may break into your homes and steal passports, letters, anything with personal information. The waiter or waitress that accepts our credit card can easily copy our credit card number. People often throw all bills and tax information into the garbage without considering how easy it is for thieves to search through ones trash. Wallets and purses are often stolen not only for the cash inside, but the credit cards, drivers license, social security card inside.

2.2.2 Dangers of Identity Theft

Once a thief has an individual's credit card number, he/she can fraudulently make purchases over the Internet or over the phone. Recently, companies have been requiring individuals to enter the three-digit security code found on the back of credit in order to combat fraud. A security code number is a three digit number found on the back of a credit card just above the signature. If the thief has an individual's physical card, they can shop in person because many merchants still do not ask for identification and only glance at the signatures. If a thief applied for a credit card, and had it sent to his/her residence, the card could be signed with the thief's signature. The bills will not go to the victim's residence and the thief will know exactly how much credit is available. It could take months, even years for the victim to discover he/she is a victim. Often this discovery occurs when a victim is denied credit because of unpaid bills. If a thief has an individual's social security number and name, the thief can open phone lines, cable TV accounts, and high speed internet accounts, and even fraudulent chequing accounts. They can rent an apartment in an individual's name and even declare bankruptcy to avoid eviction. If a thief commits a crime and is arrested, they can give the individual's name and social security instead of their own. If they don't show up in court (and they obviously won't), an arrest warrant will be issued in the victim's name.

2.2.3 How Widespread Is Identity Theft?

The following statistics were taken from US Federal Trade Commission - Identity Theft Survey Report, Sept 2003 [9] which compiled over 4000 completed interviews with a nationally representative sample. Approximately 2.4 percent reported misuse of their existing credit cards or credit card numbers in the last year; 1.5 percent reported that impostors had opened new credit accounts (including credit cards or loans), rented apartments, or obtained Medicare; 0.7 percent reported misuse of existing accounts other than credit accounts, e.g. telephone, savings, and chequing accounts; 6.0 percent had their existing credit accounts were misused; 2.0 percent had their existing non-credit accounts misused; In all, 12.7 percent reported that they have discovered misuse of their information and 4.6 percent said they were victims of identity theft in the last five years.

According to Phonebuster, Canada's fraud call center located in Ontario, over 14599 Canadians have reported being victims of identity theft, losing a combined 21 million dollars in 2003 [30]. Reports of identity theft have decreased in Canada, with reports of 11231 victims losing a combined total of just under nine million dollars in 2005 [32].

Reports of identity theft has decreased slightly in the United States; 10.1 million individuals reported being victims of identity theft in 2003 compared to 9.3 million in 2005 [34].

3 RELATED WORK

3.1 Encryption

3.1.1 Overview

Encryption is the process of disguising messages sent from one party to another. Only the intended party should be able to make sense of the message, thus foiling any third party that intercepts the message. Encryption has evolved over the years, especially in the last century with advent of public keys, discussed later in this chapter. Encryption is the central security issue of any distributed system because it can ensure that messages that are intercepted between two remote parties remains private.

The table below defines some key terms regarding encryption that will be used throughout this section.

Table 3-1 Key Encryption Definitions

Term	**Definition**
Plaintext	The original, unaltered message intended to be sent to the intended receiver.
Encryption	A process of encoding plaintext in order to disguise the message.
Cipher	Algorithm or scheme used to encrypt plaintext.
Ciphertext	Plaintext that has been encrypted.
Decryption	The process of decoding or restoring ciphertext to its original plaintext.

3.1.2 Classical Encryption

Traditionally, the strength of encryption was based on the fact that unintended third parties did not know the algorithm used to encrypt the data. The algorithm below shows that a classical encryption algorithm is a function that accepts the message as its only argument.

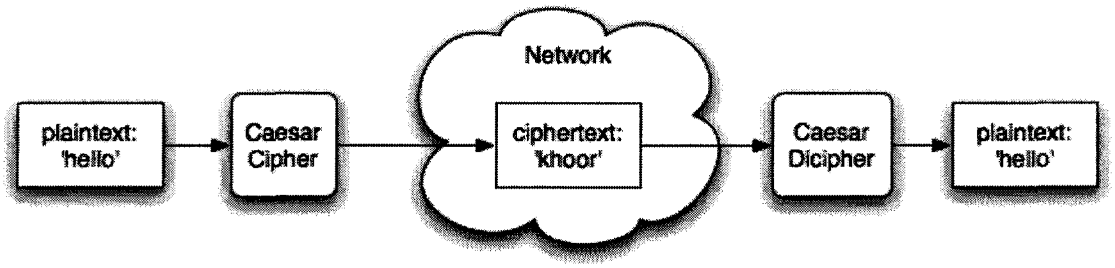
Equation 3-1 Classical encryption function. a) Encrypting plain text. b) Decrypting ciphertext.

a) $Ciphertext = Encryption(Plaintext)$

b) $PlainText = Decryption(CipherText)$

One of the earliest encryption techniques is know as the Caesar Cipher. Here each letter of the alphabet would be replaced with the letter standing three letters after it, assuming the traditional ordering of the alphabet. Hence, 'a' would be replaced with 'd', 'b' would be replaced with 'e', 'c', would be replace with 'f', and 'y' would be replaced with 'b'. Notice that 'y' is replaced with 'b' because when we try to move three spaces ahead and we run out of letters, so we continue with the beginning of the alphabet. In order to decrypt the ciphertext, the intended party replaces each letter with the letter standing three spaces before it. If we reach the end of the alphabet while looking for a letter three spaces before, we continue with the end of the alphabet. For instance, 'a' would be replaced with 'x'. The Caesar cipher is known as a substitution technique because each unit, a letter in this case, is substituted with a different unit. The figure below illustrates the Caesar Cipher.

Figure 3-1 A plaintext message is encrypted using the Caesar Cipher and sent over a network and decrypted by its intended receiver.



3.1.3 Secret Key Encryption

Secret key encryption, also called symmetric encryption, is similar to classical encryption, except the encryption algorithm takes in a secret key with the plaintext as input and the decryption algorithm takes in a matching key with the ciphertext to restore the plaintext. The functions below show that a secret key encryption algorithm is a function that accepts the message and a key as its arguments.

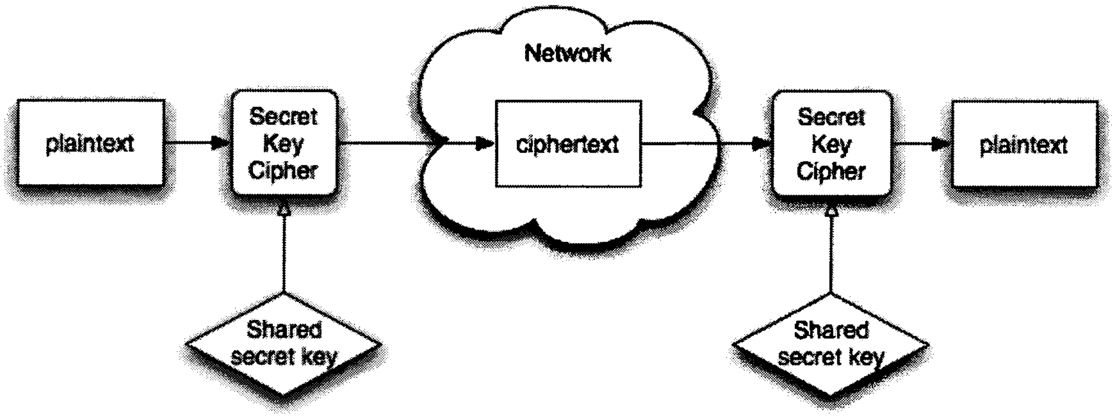
Equation 3-2 Secret Key encryption. a) Encrypting information with a secret key. b) Decrypting ciphertext with a secret key.

a) *Ciphertext = Encryption(Plaintext, secret_key)*

b) *Plaintext = Decryption(Ciphertext, secret_key)*

The figure below illustrates a generalized secret key encryption model.

Figure 3-2 A generalized model of a shared secret key encryption system.



Whereas traditional encryption relied on the ignorance of a would-be eaves dropper's knowledge of a particular encryption algorithm, a symmetric encryption scheme's strength usually lies in the size of its key. Here, an eaves dropper may know the algorithm used but cannot view the plaintext without the actual key.

Most modern encryption schemes are block ciphers which encrypt blocks of data, usually 64 – 128 bits in length (as opposed to stream ciphers which encrypt one bit at a time). Almost all significant block ciphers are based on the Feistel Cipher which proposed a series of substitutions and permutations over a number or rounds. The table below describes the more popular block ciphers available today.

Table 3-2 Description of widely used symmetric block ciphers.

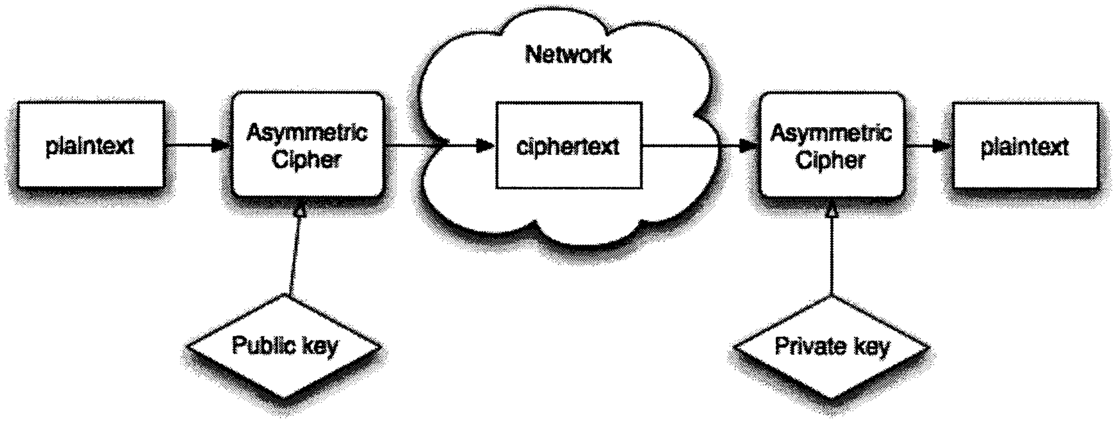
Cipher	Description
DES (Data Encryption Standard)	Adopted in 1977 by the National Bureau of Standards. Data is encrypted in 64 bit blocks with the use of a 56 bit key. There are concerns that a 56 bit key does not offer enough security.
Triple DES	Requires two 56 bit keys and follows an encryption-decryption-encryption scheme using DES. Not susceptible to a brute force attack.
Blowfish	Developed by Bruce Schneier. Keys are variable length and can be as large as 448 bits.
RC5	Developed by Ron Rivest, the "R" of RSA (discussed below).Variable size of keys and blocks (16, 32, or 64 bits).
AES (Advanced Encryption Standard)	Chosen by the National Institute of Standards in 2001. The institute selected an algorithm created by Joan Daemen and Vincent Rijman which allow keys of length 128, 192, and 256 bits. It also allows variable block lengths of 128, 192, and 256 bits.

3.1.4 Public Key Encryption

Public key encryption (also known as asymmetric encryption) is used to solve the problem of requiring a pre-established shared key before secure communication. For instance, if we required a shared key for all credit card transactions online, this would seriously limit where we could shop online. We could only shop at stores that have somehow delivered a shared key to us or vice versa.

Asymmetric encryption uses two keys for secure communication, a private key and a public key. The private key must be well guarded, whereas the public key is to be distributed publicly. Messages encrypted with a public key can only be decrypted by its matching private key. The public key cannot be used to derive the private key, hence, anyone with person A's public key can send encrypted messages to person A without fear of someone eavesdropping. The figure below illustrates public key encryption.

Figure 3-3 Public key encryption. A public key is used to encrypt a message, and the corresponding private key is used to decrypt it.



RSA (named after it's inventors, Ron Rivest, Adi Shamir, and Len Adleman) was created in 1977 and remains the most popular public key encryption algorithm. The public and private key are interchangeable, meaning that the private key can also encrypt data and only the public key can decrypt it. This is useful for digital signatures (see below).

3.2 Internet Security

3.2.1 Message Digests and MACs (Message Authentication Codes)

Message digests are used to ensure that a downloaded message has not been tampered with. Hash functions are used to convert the message into a fixed length bit string. It is highly unlikely for two messages to have the same hash value. The hash function is also a one way function, meaning that it is impossible to derive the original message from its hashed value.

Message digests can be used to verify that a downloaded file, such as a program, has not been altered during the download. It is also widely used by operating systems to verify passwords. When a user creates a password, only the hashed value is stored. When that user enters his/her password to login, it is hashed and the two hash values are compared. This means that it is not necessary to store the actual password of the users on the system.

MD5 and SHA1 (Secure Hash Algorithm) are the most used message digest algorithms. MD5 was written by Ron Rivest. It produces a 128 bit message digest. SHA1 produces a 160 bit digest.

MACs are message digests created with a private key. This means that in order to validate a message, one needs a private key, adding some level of security. The equations below indicate the differences between message digests and MACs.

Equation 3-3 Message digests

$$MessageDigest = MD\_Hash(message)$$

Equation 3-4 MACs (Message Authentication Codes)

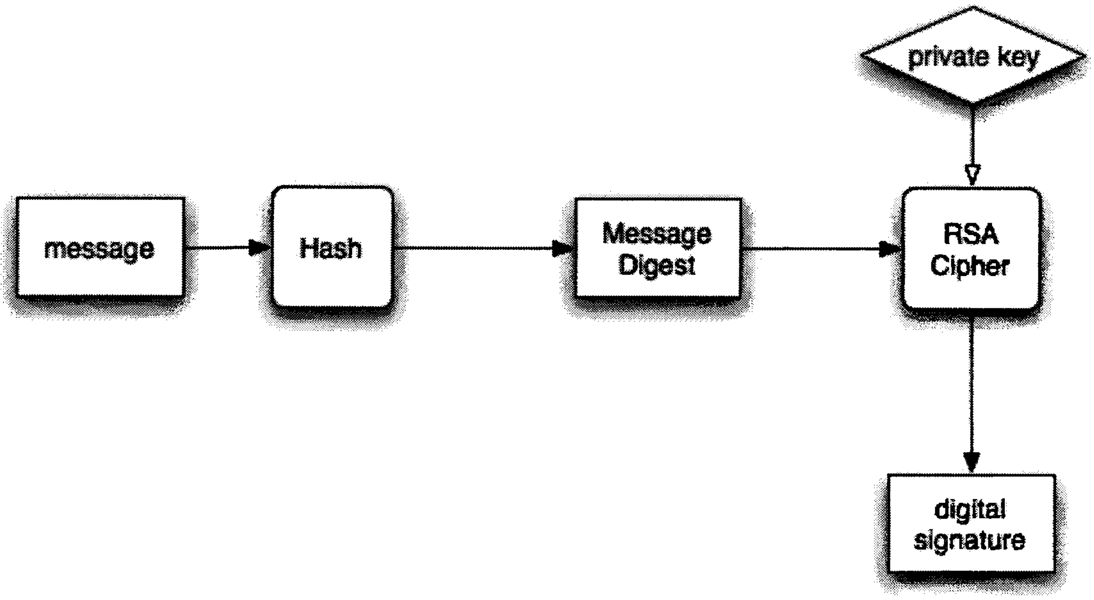
$$MAC = MAC\_Hash(message, privateKey)$$

3.2.2 Digital Signatures

A digital signature, as its name implies, is used to ensure that a message is from its stated source and that it has not been tampered with. If a person signs an envelope over the flap and mails the letter to its intended destination, the person who receives the destination knows that it is from the claimed sender and the contents have not been tampered with if and only if the signature is untouched over the flap. Digital signatures can provide that same service with digital messages.

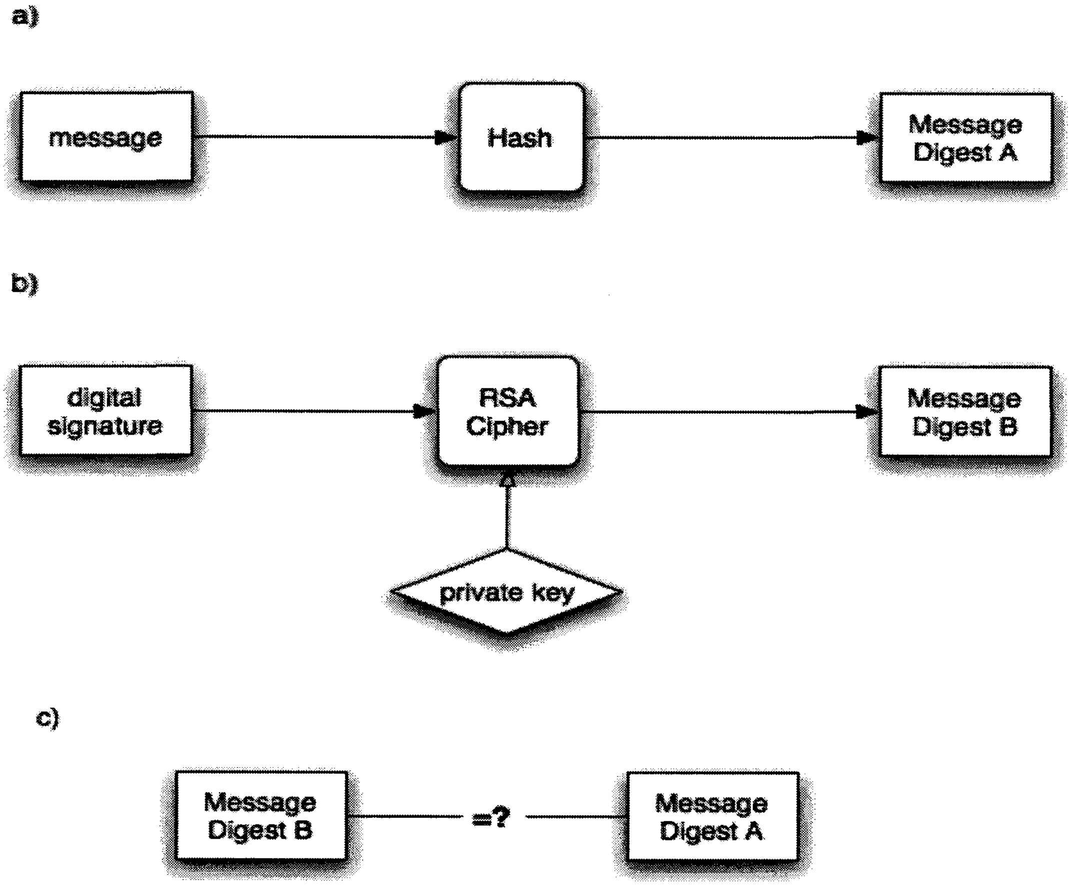
A digital signature for a message is created by creating a message digest for that message. The digest is then encrypted with a private RSA key. Recall that RSA public-private key pairs have the quality of being interchangeable, meaning that someone with a public key can decrypt the signature and verify that it was signed by the sender. The process is illustrated below.

Figure 3-4 Creation of a digital signature



Since both the message and the signature are sent to the destination, the receiver can decrypt the signature with the sender's public key. If they are equal, the message is verified to be from the sender. The figure below illustrates the verification process.

Figure 3-5 Verification of the digital signature. a) Receiver creates message digest A by hashing received message, b) receiver creates message digest B by decrypting the received digital signature with the senders public key, c) compare message digest A and B. If they are equal, the message came from the claimed sender and has not been altered.



3.2.3 Digital Certificates

Although digital signatures allow us to verify that a message comes from its claimed source, we still have to assume that the public key is the actual public key of the sender. In order for us to verify this, we can use a CA (central authority) such as Verisign to act as a trusted third party to ensure that the public key is the actual public key we think it is. This way, we would only need one authentic the public key (that of the CA) and the CA could then certify our other public keys, i.e. give a digital certificate containing a desired public key.

3.2.4 Secure Sockets Layer (SSL)

SSL was designed by Netscape to send and receive encrypted messages over TCP (transport communications protocol). SSL is usually used over HTTP (referred to as HTTPS when SSL is used). SSL is supported by most browsers, including Microsoft and Netscape and has become the de facto standard for secure communication over the Internet. SSL uses RSA's public and private key encryption and uses digital certificates.

3.3 Information Flow Control

3.3.1 Introduction

The distributed nature of computer systems has made issues of security of utmost importance. In the mid to late nineties, there has been an explosion of programs (Java Applets, Javascript) being downloaded onto PCs and running in conjunction with other programs. It is necessary to ensure that information remains private, safe, and available to those who require it. Unfortunately, most of the computer industry's effort has been to assure the safety and privacy of information, however, this has made the propagation of this information over restricted, i.e. it has been very difficult to give access to particular parties and control what these parties do with the information [21]. For instance, the Java security model prevents many useful applications, but still permits Trojan Horse applets to leak private information [21]. The control of the propagation of information will henceforth be referred to as information flow control.

In 1997, Andrew Myers and Barbara Liskov of MIT presented an academic paper introducing the Decentralized Label Model [21]. This model addresses the problem of controlling the propagation of information. Myers wrote a PhD thesis at MIT in 1999 based on this model and won the George M. Sprowls Award for outstanding Ph.D. thesis in the MIT EECS Department. He has since published a handful of academic papers extending the model and introducing JFlow (formerly JIF - Java Information Flow), an extension of Java, which implements the decentralized label model.

3.3.2 Decentralized Label Model

The DLM (Decentralized Label Model) is a finely grained information model. A key feature is that each peace of data is annotated with a label that describes which principals own the piece of data and which principals can read the data.

In this context, a principal is some type of user or group of users. For instance, we can think of anyone with a userid and password as a principal. We can also think of a group, let us say graduate students, as a principal. Principals also have an *act for* principal in which a principal has the authority to read and change the information of the principal it acts for and change who can read the information.

Each piece of data used in a program will have a label associated with it that will describe the policies of the data, i.e. who owns the data and who can read the data. Each value can have multiple policies. A policy includes two parts, the owner and a set of readers. The readers are the principals whom the owner allows to read the datum. For example, let us say that we have a variable total that has the following label:

{Sal: Ted, Rob, James; Jason: William, Ted, Rob}

This label has two policies, namely:

1. Sal: Ted, Rob, James, and
2. Jason: William, Ted, Rob.

In the first policy, the owner is Sal and the Readers are Ted, Rob, and James, while in the second, the owner is Jason, and the Readers are William, Ted, and Rob. Each policy is of the form: owner: reader 1, reader 2, … , reader n.

Because variables may have two or more policies, there is still the question of how to consolidate the readers list. The Decentralized Label Model takes the intersection of the sets of readers and gives read access only to them. This ensures that the only principals who can read the information are agreed upon by *all* the Owners. If a policy only contains an owner and an empty set of readers, then only the owner can read the information. If no policies appear in the label, then anyone can read and propagate the information.

When manipulating labelled data, the program must obey certain rules. Every variable has a label which applies to the value it stores. When the value is read from this variable, it assumes the label of the variable in which it was read. When a value is placed into a variable, it assumes the label of that variable and forgets its old label. To ensure that the program does not leak information, the variable's label in which this new value is placed must be at least as restricted as the label of the value. This type of re-labelling is called restriction.

A label L1 is said to be less restrictive of label L2, if L2 contains all the owners of L1 and the reader set of L2 is a subset of the reader set of L1 [21]. The following list gives some examples of policies which are less and more restrictive:

- {A: B, C, D, E} is less restrictive than {A: B}
- {A: C, Z; B: C} has the same restrictions as {A:C; B:C }
- {A: B, C} is less restrictive than {A: C}
- {} is less restrictive than {A: A}
- {A: B, C} is more restrictive than {A: B, C, D}
- {A: Z; B, C} is more restrictive than {A: B}
- {A: B, C} is more restrictive than {A: A, B, C}

The examples below illustrate how variables can be manipulated. For example, let us assume we have two variables with the following labels:

```
variable1 {Tom: Sal, James, Jack}

variable2 {Tom: Sal, James}
```

If we extract the value from variable1, the value will assume the value label of variable1. If we try to assign this value to variable2, it is considered legal because the label of variable1 is less restrictive than variable2 and the value assumes the label of variable2.

If a value is obtained by two or more variables, then this new value must contain the restrictions of all the variables which are used to obtain it, i.e. the label of the new variable must be at least as restrictive as all of the variables that were used to obtain the value of the new variable. To do this, we simply take the union of all the policies. For example, let us say the value1 was obtained using value2, value3, and value4, which have the following labels:

```
value2 {Sam: Jen, Anne, Tammy}

value3 {Jill: Jen, Anne}

value4 {Dave: Jen, Anne, Tom; Jack: Jen, Anne}
```

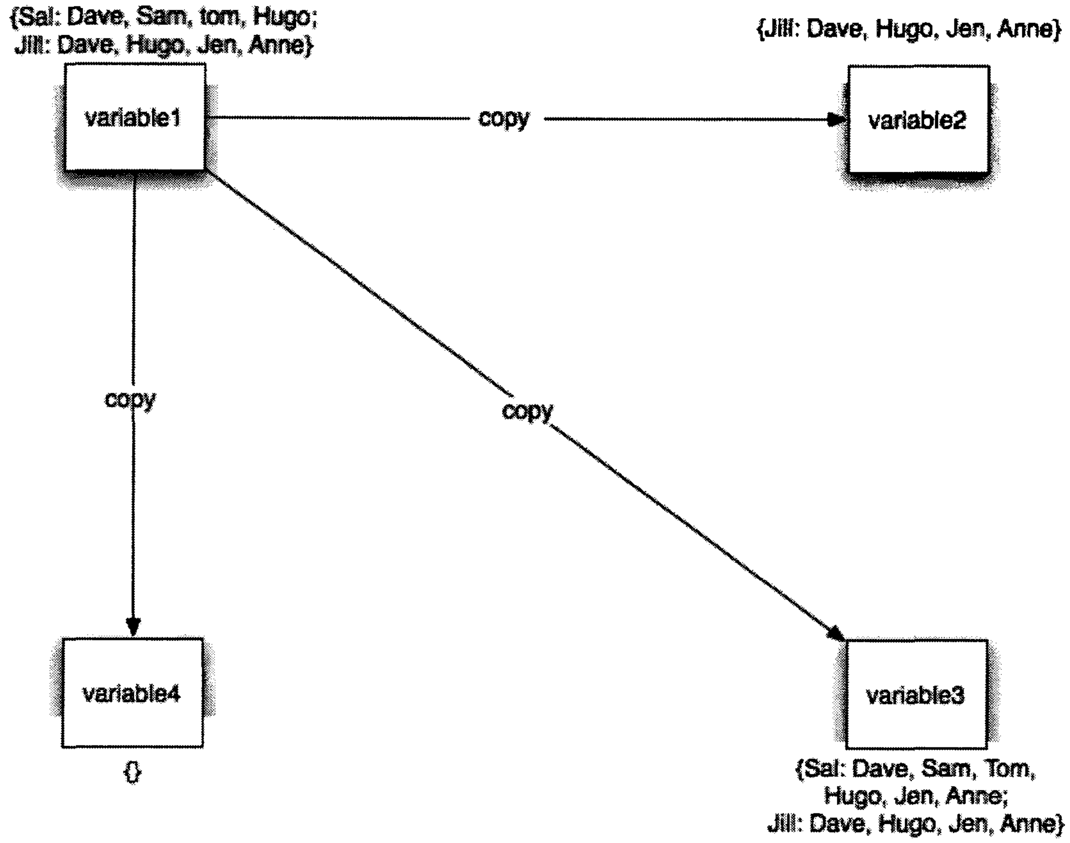
Then the label of value1 would be as follows:

```
value1 {Sam: Jen, Anne, Tammy; Jill: Jen, Anne; Dave: Jen,
Anne, Tom; Jack: Jen, Anne}
```

This protects the privacy of variables even when it is only being used for computation.

The distributed label model also has a method for safe declassification. Each label has information about the owners of information. These owners can relax the policies when appropriate. Any process that occurs in a program is authorized to act for a set of principals, referred to as the authority of the process. This process can declassify, or relax restrictions on data, by creating a copy of the data whose label is less restrictive. This is done by adding readers to the reader set of the principal who has the authority of the process, or removing the policy completely. Declassification is limited because it does not allow declassification of information that is not owned by the principals who have authority for the current process. Declassification slows down the decentralized label model during run-time because declassification depends on the runtime authority of the process, which requires a runtime check of authority. The diagram below illustrates declassification of a variable.

Figure 3-6 A process declassifies Variable1 by creating three copies with few restrictions.


{Sal: Dave, Sam, tom, Hugo;
Jill: Dave, Hugo, Jen, Anne}
{Jill: Dave, Hugo, Jen, Anne}
variable1
copy
variable2
copy
copy
variable4
{}
variable3
{Sal: Dave, Sam, Tom,
Hugo, Jen, Anne;
Jill: Dave, Hugo, Jen, Anne}

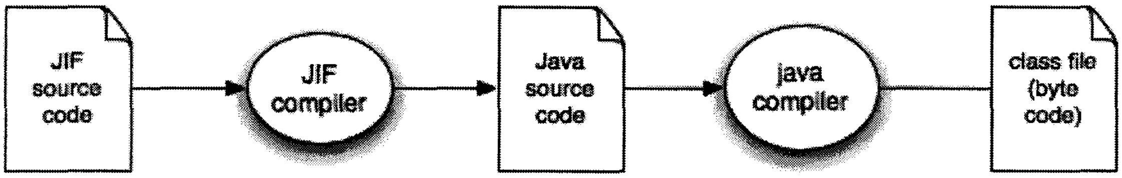
From figure 3, we can see that since the process runs on the authority of Sal and Jill, it can effect any policy that is owned by these two principals by making copies and adjusting the labels. All of the new variables, Varible2, Variable3, and Variable4, have fewer restrictions than Variable1.

3.3.3 JFlow

JFlow (formally known as JIF) is an extension to the Java language that incorporates the decentralized label model. The information flows are very fine-grained because the language allows all data to be annotated with decentralized labels [24]. The dynamic performance of these programs does not greatly decrease because the annotations can be checked statically, reducing a lot of run-time overhead, treating it as an extended kind of type checking (often referred to as label-checking).

Once a program is written in JIF, the JIF compiler then translates the JIF source into Java source, which can be compiled using any Java compiler. The JIF compiler also produces another file with information regarding the labels within the JIF source code. This file is used by the bytecode file that is produced by the Java compiler. The bytecode can also be used by the JIF compiler to compile other JIF files, as is shown by the dotted line. The figure below illustrates the process.

Figure 3-7 JIF compiler.



During the translation from JIF source code to Java source code, the label annotations are checked to see if there are any potential data leaks. In this way, it prevents a lot of dynamic label (type) checking, thereby reducing runtime overhead.

4 RELATED TECHNOLOGIES

4.1 Web Services

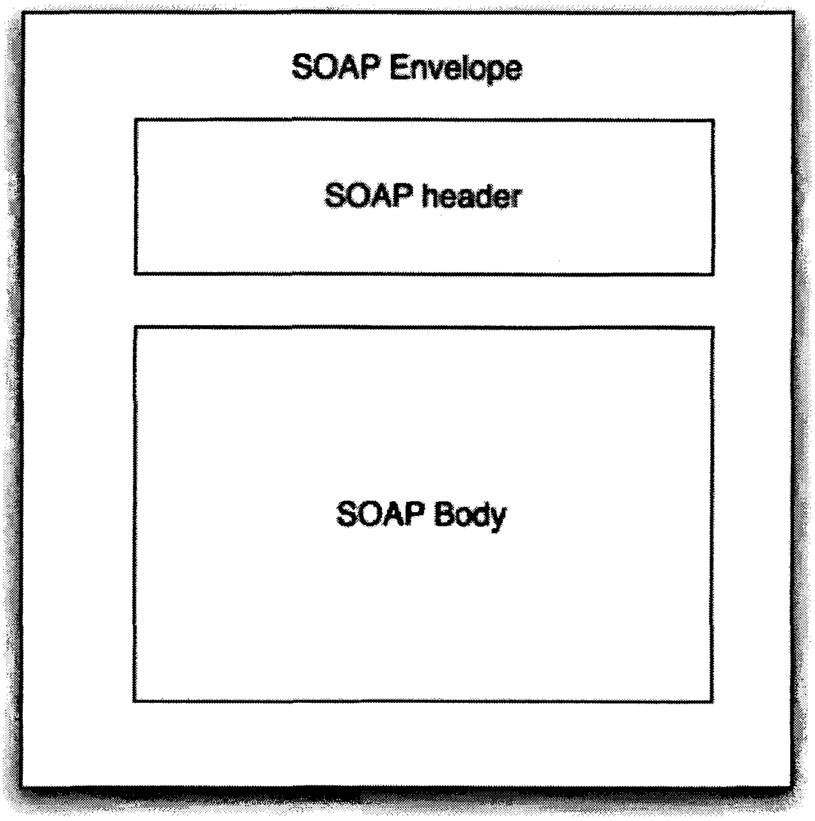
4.1.1 Introduction

Web services provide a standard means of communication between different software applications, running on a variety of platforms and/or frameworks. Web services are loosely coupled endpoints which provide services, usually in "request/response" scenarios. Messages are in XML format, sent in SOAP envelopes (see below).

4.1.2 SOAP

SOAP is a W3C recommendation which provides the definition of XML-based information which can be used for exchanging structured and typed information between peers in a decentralized, distributed environment [42]. Although SOAP is primarily a one-way communication paradigm, complex communication, such as request/response can be achieved by combining one-way exchanges within an underlying protocol. Since XML is becoming the de facto standard for transferring data between parties, SOAP is ideal for communication between disparate systems.

Figure 4-1 High level structure of a SOAP envelope

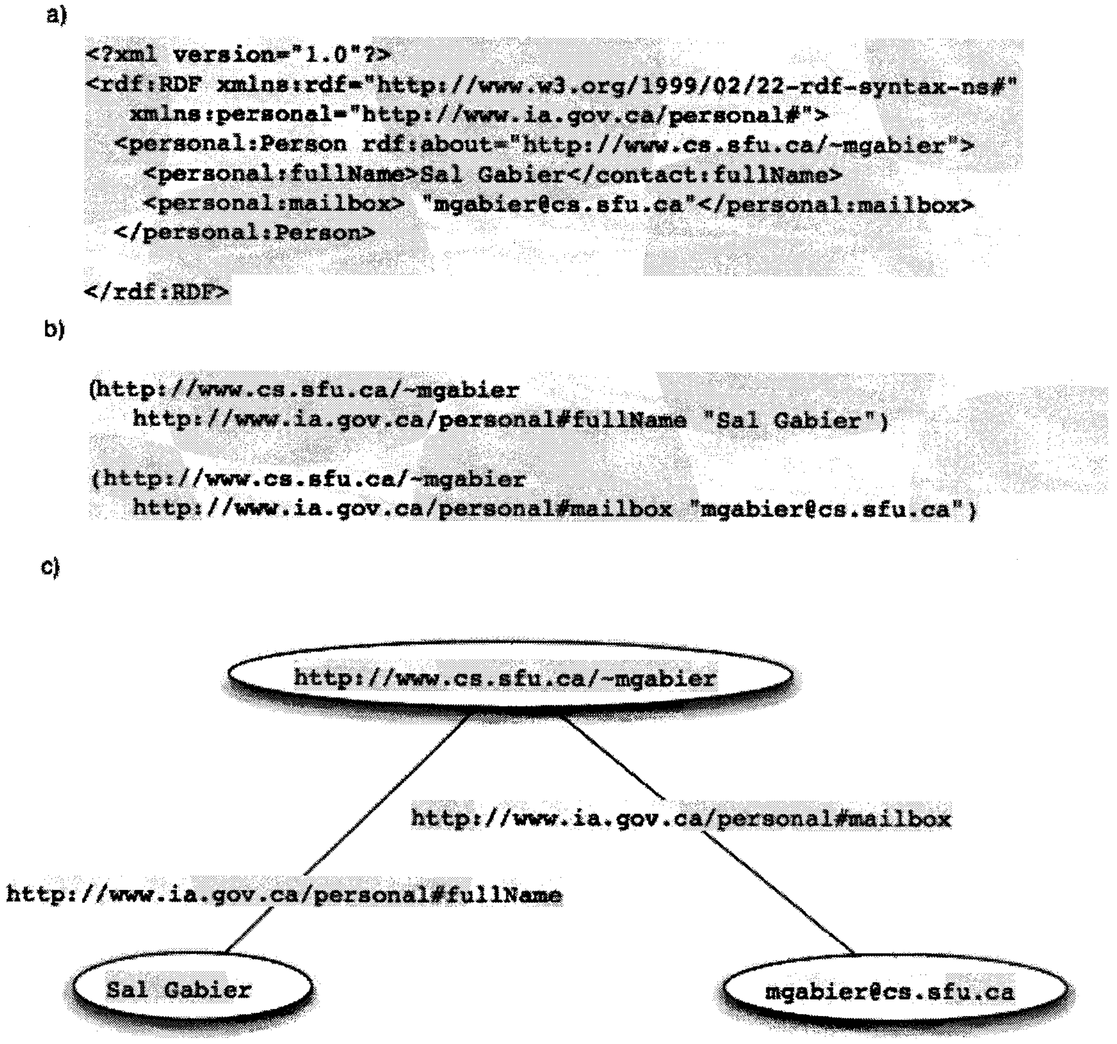


4.2 RDF

4.2.1 Introduction

The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. RDF is a XML language which consists of sets of triples which represent semantic webs. Each triple consists of a subject, predicate and object. The subject and predicate must be URIs (Uniform Resource Identifier), while the object can be a URI or a constant [41]. The figure below illustrates a very simple semantic web.

Figure 4-2 a) RDF infoset describing Sal Gabier semantic web. b) Triples representation of the Sal Gabier semantic web. c) Graphical representation of the Sal Gabier semantic web



a)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:personal="http://www.ia.gov.ca/personal#">
  <personal:Person rdf:about="http://www.cs.sfu.ca/~mgabier">
    <personal:fullName>Sal Gabier</contact:fullName>
    <personal:mailbox> "mgabier@cs.sfu.ca"</personal:mailbox>
  </personal:Person>

</rdf:RDF>
```

b)

```
(http://www.cs.sfu.ca/~mgabier
   http://www.ia.gov.ca/personal#fullName "Sal Gabier")

(http://www.cs.sfu.ca/~mgabier
   http://www.ia.gov.ca/personal#mailbox "mgabier@cs.sfu.ca")
```

c)

http://www.cs.sfu.ca/~mgabier
http://www.ia.gov.ca/personal#mailbox
http://www.ia.gov.ca/personal#fullName
Sal Gabier
mgabier@cs.sfu.ca

RDF is particularly intended for representing metadata about Web resources, which are identified with URIs. However, since URI's are universal and can be used to identify virtually anything, RDF can create semantic webs that describe real world entities.

RDF is intended for communications between *machines* without loss of meaning, not necessarily people. For complicated RDF infosets, it is often useful to create graphical images of a semantic web, making it easier for people to understand, such as figure 4.2.c. Since it is a common framework, application designers can leverage the availability of common RDF parsers and processing tools. The ability to exchange information between different applications means that the information may be made available to applications other than those for which it was originally created.

4.2.2 JENA

Jena is an open source Java framework developed at HP Labs Semantic Web Programme for building Semantic Web applications. Jena is distributed as a Jar archive file which can easily be plugged into any Java application. Jena can read any well-formed RDF infoset and save the resultant semantic web as composite java objects. As discussed above, RDF is ideal as a common language for transferring information between heterogeneous information systems. Jena eases the manipulation and parsing of RDF.

5 INFORMATION AGENTS

5.1 Introduction

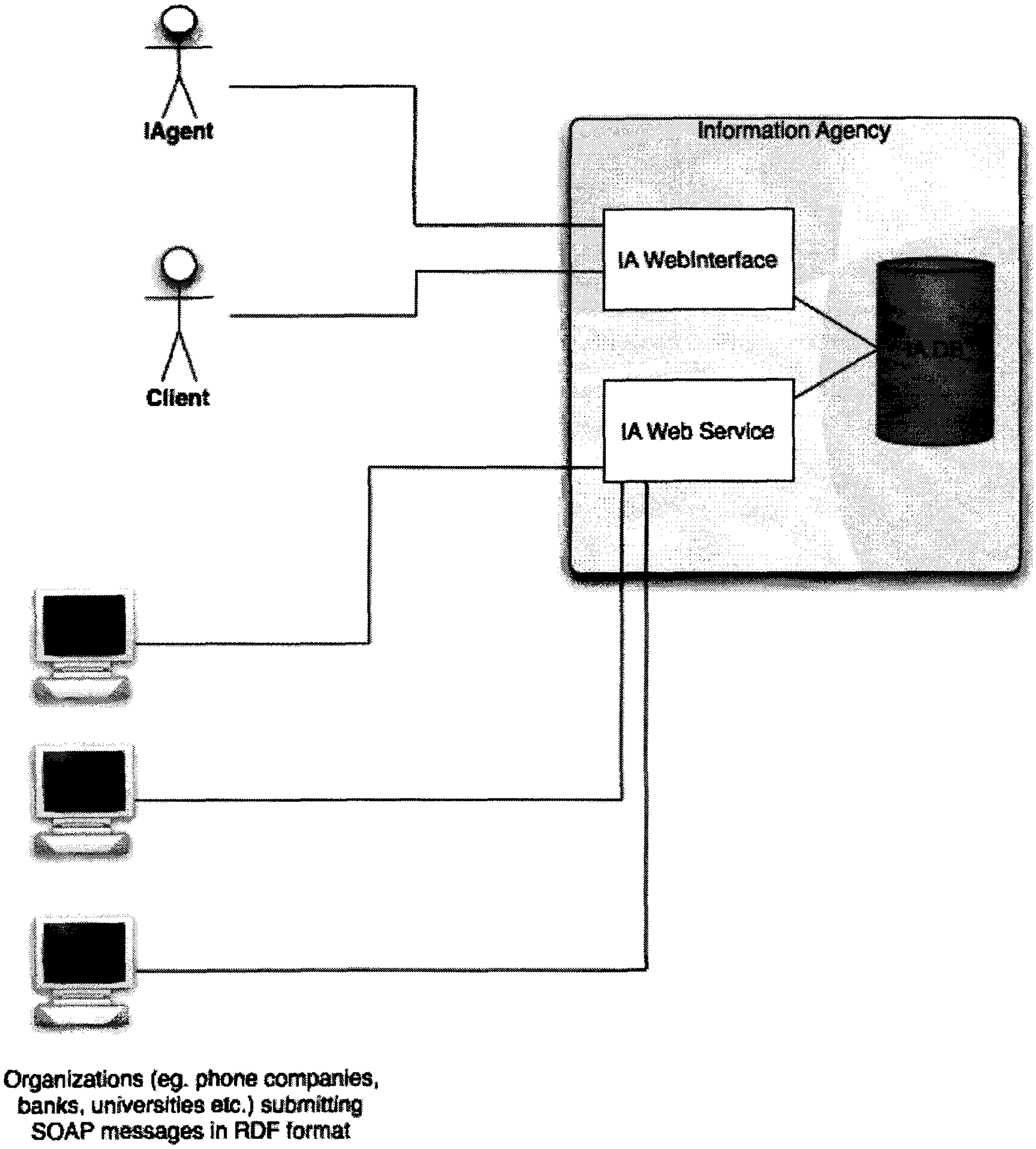
Information Agents are essential for an active information model, where organizations are required to inform individuals through electronic information updates when their information is edited, created, received, or transmitted. In order to establish an active information policy (discussed in section 5.2), the public must be able to cope with the influx of information as organizations bombard the public with information updates. Information Agents can act as the intermediary that assists the public. An Information Agent will have to be more than just a layer of software, but a professional who provides information services that may be impossible for the general public to perform. Information Agents shall be responsible for maintaining and cataloguing information as it is received, ensuring that the information is accurate, and vigilantly detecting anomalies which would indicate criminal activity.

It is easy to imagine that if organizations are required to send information updates when an individual's information is edited, these updates can be overwhelming. Each time an individual makes a purchase with a credit or ATM card, an information update will be sent to that individual. Each time we make a phone call or browse the Internet (assuming one's ISP records browsing histories), an information update will be sent. This chapter tackles the following questions:

- How can an individual cope with the influx of information updates?
- How exactly will electronic information updates be formatted?

- If Information Agents act as an intermediary, how can we establish the necessary trust in these agents?

Figure 5-1 An overview of an active security model. Organization such as phone companies, banks, universities, etc. submit information updates to Information Agent each time the organization creates, updates, or transmits personal information. The clients of information agencies can view their information, as well as which organizations have their personal information.


IAgent
Client
Information Agency
IA WebInterface
IA Web Service
IA DB
Organizations (eg. phone companies, banks, universities etc.) submitting SOAP messages in RDF format

5.2 Active Information Policy

5.2.1 Overview

An active information policy requires all organizations to inform a particular person of a creation, update, reception or transmission of his/her personal information. For instance, the phone company maintains its client's billing information as well as the client's incoming and outgoing calls. Each time a client makes or receives a phone call, the phone company logs that call. An active information policy would require the phone company to inform the client that a call has been logged. This may seem trivial, but such a policy is particularly useful when an organization sells personal information to other companies. Often, individuals do not know about the transaction, nor do they know which companies have their personal information. This would also limit the damage of an identity thief. For instance, if an identity thief opened a credit account in another person's name, the credit company would immediately inform the individual whose name was used to open the account. Identity thieves rely on the victim's ignorance of their existence in order for the thieves to continue fraudulently obtaining goods and services.

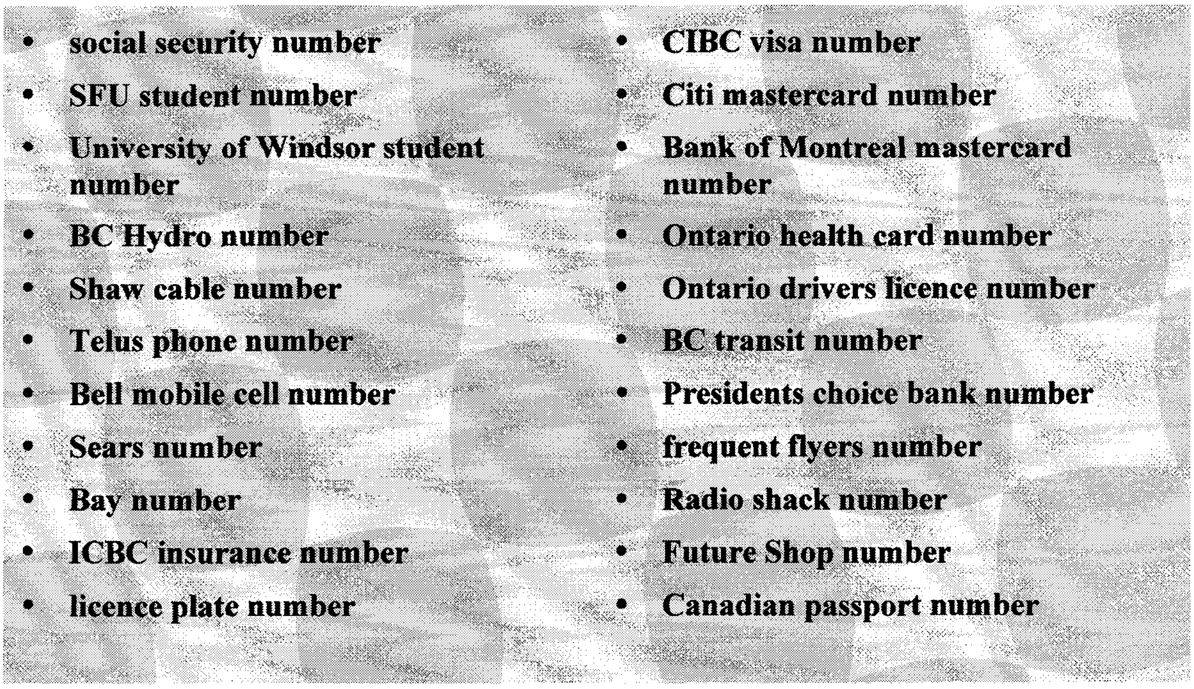
North America employs a "passive" information policy. Although companies are required to submit to an individual his/her personal information when requested, the public is not aware of all the companies that have their personal information. Put simply, an individual cannot request his/her personal information from a company if he/she is not aware that the company has his/her information. Companies are not required to actively inform us if they are storing our information. There are numerous companies that secretly obtain our personal information and sell it to whomever is willing to pay.

5.2.2 Information Overload

It is difficult to measure how many organizations have an individual's personal information. Companies often sell personal information, and other companies boast about their ability to illegally obtain personal information [19, 35]. We can look at particular companies and count how many records of individuals it has, but we can't look at a particular individual and discover how many organizations have his/her personal information. Because of this, it is difficult to accurately estimate how many information updates an individual will receive in any time frame if an active information policy is implemented. We can, however, imagine that there would be numerous updates each month, perhaps, each day. Every time we make a purchase with a debit or credit card, make a phone call, or pay a bill, an information update will be sent to us. It becomes quite obvious that if our society employed an active information policy, the amount of information that will be sent to individuals from companies will be abundant.

An interesting exercise to illustrate how many companies have our personal information is to take ten minutes and brainstorm the number of unique identifiers, such as a social security number, bank account number, or user ID, that organizations use for your identification. Usually these identifiers identify records with our personal information.

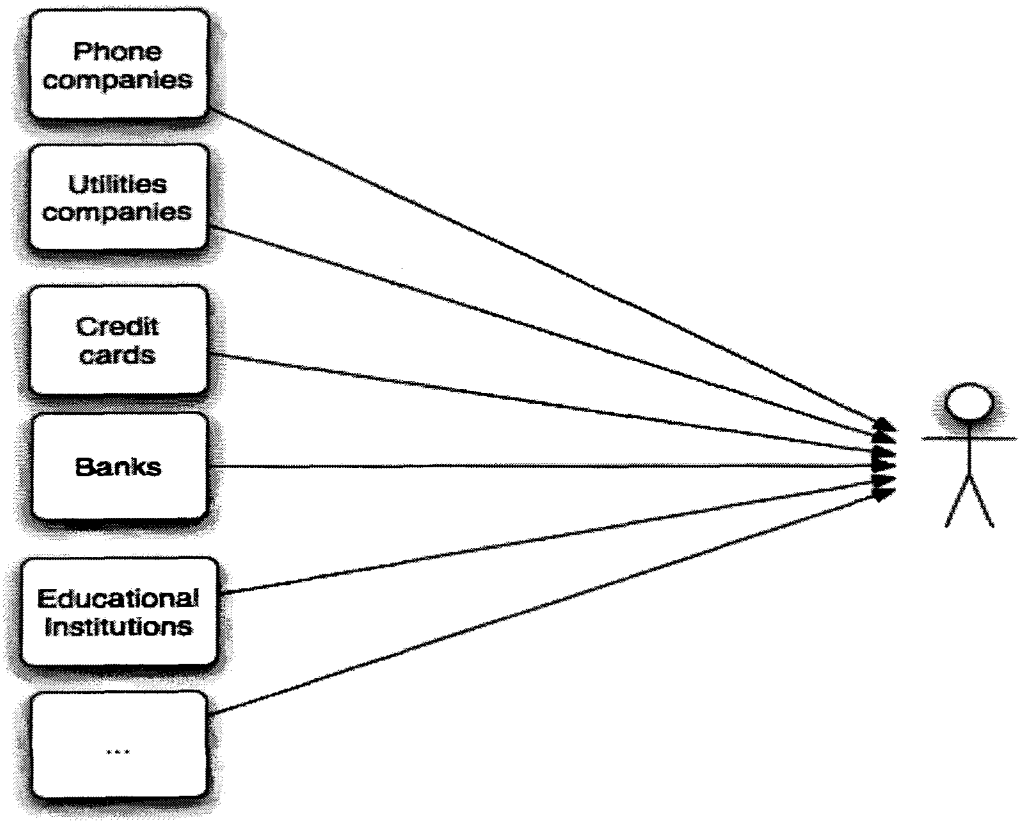
Figure 5-2 Below is a list of my unique identifiers. The list is the result of ten minutes of brain storming and is not exhaustive.



- **social security number**
- **SFU student number**
- **University of Windsor student number**
- **BC Hydro number**
- **Shaw cable number**
- **Telus phone number**
- **Bell mobile cell number**
- **Sears number**
- **Bay number**
- **ICBC insurance number**
- **licence plate number**
- **CIBC visa number**
- **Citi mastercard number**
- **Bank of Montreal mastercard number**
- **Ontario health card number**
- **Ontario drivers licence number**
- **BC transit number**
- **Presidents choice bank number**
- **frequent flyers number**
- **Radio shack number**
- **Future Shop number**
- **Canadian passport number**

If we take into account that there exist companies such as National Demographics and Lifestyles (NDL) that possess 25 million personal dossiers which they sell to over 130 companies with more than 300 brands [10], it is clear that the average citizen will have trouble coping with the almost daily information updates that would surely be sent. This will lead to an information overload (see figure 5-3). Information agents will act as the intermediary and provide the necessary information services for individuals to cope with incoming updates.

Figure 5-3 Organizations sending information updates each time records are created, updated, transferred or received can lead to information overload.



5.3 Role of the Information Agent

An Information Agent is a professional whose purpose is to provide informational services to his/her clients that relate to the management the client's personal information. The Information Agent will act as the intermediary between clients and the organizations that send information updates. The responsibilities of the Information Agent include the following:

- Acting as the intermediary between the client and organizations electronically sending information updates when organizations create, edit, transfer, and receive personal information regarding the client.
- Ensuring that all the information that external organizations have on their clients are current and accurate,
- Ensuring external organizations are not abusing their clients information,
- Recognize companies that have a pattern of abuse or poor information handling which lead to incorrect information and reporting them to the authorities.

Since the Information Agent will have to cope with electronic updates, he/she will need to be technologically savvy. Appendix A describes the IA Server, a tool that can receive information updates in the form of semantic webs and store this information in a meaningful way, enabling the Information Agent to easily perform his/her tasks. Using existing technologies, a tool can be created that can:

- Receive electronic information updates in the form of semantic webs and store the information in a meaningful format enabling management of the information.
- Keep track of all companies that have personal information regarding the client and monitor their use of the information.
- Check for consistency with internal records and the information of external organizations.
- Provide security that hides personal information from untrusted parties. Clients should be able to control the access to their personal records.

5.4 Achieving Trust

Achieving trust may be the most difficult task in an active information model. Trust can only be obtained by decentralizing Information Agents, i.e. there should be no central private or government agency with stockpiles of everyone's personal information. In order to gain trust, citizens should be able to choose their own Information Agent the way we can choose our own lawyers or doctors. Individuals should also have the option of being their own Information Agent. All that would be required is a layer of software that can cope with the influx of electronic information updates. It would then be up to that particular person to maintain the information.

5.5 Format of Information

Information updates will be sent as semantic webs (see Section 4.2). Semantic webs can be represented as a RDF infoset. Since RDF is an XML language, companies can use a host of tools in many different languages to construct the updates and send them as SOAP messages. SOAP is becoming the *de facto* standard for communications between heterogeneous systems which is ideal for an active information culture because it does not require companies to store information in any special structure or format, nor does it require any specific programming language. Different information Agents can develop IA tools/servers with web services which can accept these information updates over a secure connection.

Each web service needs a URL so that clients have an address with which to send messages. IA tools/servers are no different. Clients have to therefore be able to resolve an individual's unique ID (in North America we use social security numbers) to a URI in order to send the electronic information updates. Resolving social security numbers to URIs is beyond the scope of this project.

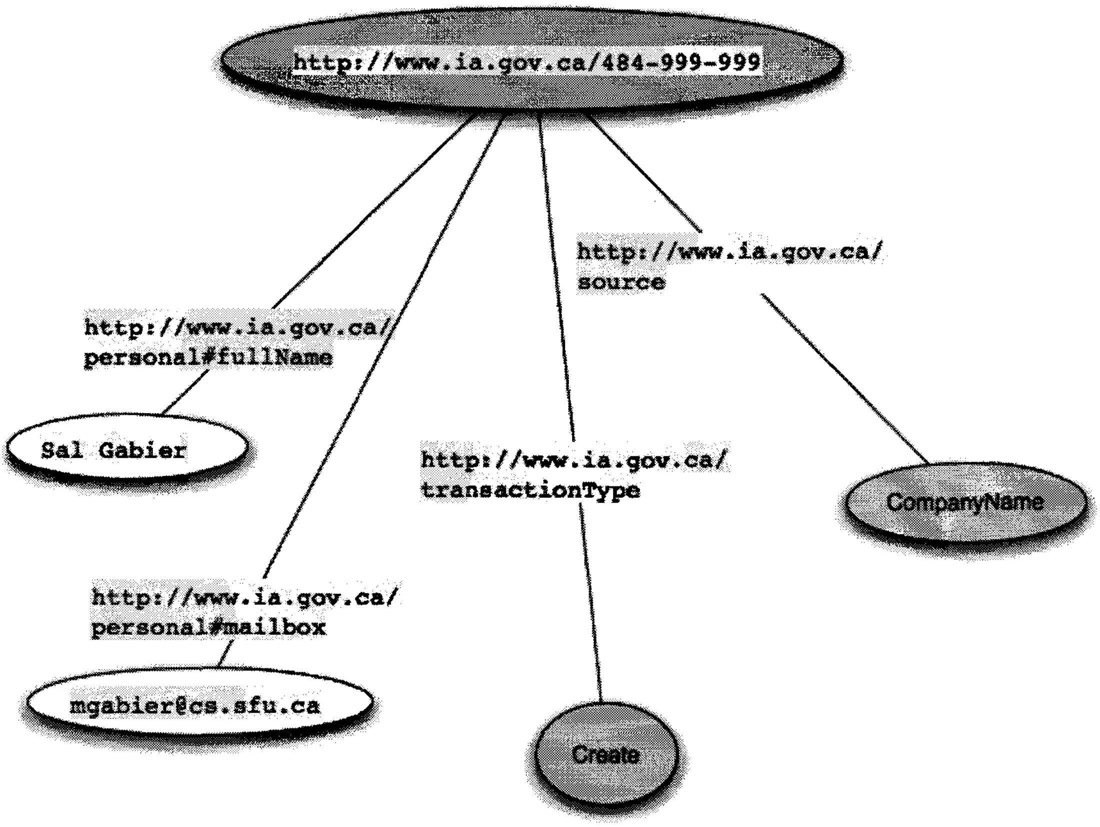
Although semantic webs offer a great deal of flexibility, an active information policy must enforce certain rules regarding the structure of information updates. The rules are as follows:

- The base of the semantic web must contain the client's social security number. This is to uniquely identify the client. Since the root is a subject node and must be a URL (see section 4-2), the following form shall be used – www.ia.gov.ca/999-999-999, where the last nine numbers are the client's social security number.
- Each semantic web must contain a predicate and subject indicating the type of information update – create, update, transfer, or receive. The predicate must be labelled www.ia.gov.ca/transactionType, and its attached object will describe the transaction.

- Each semantic web must contain a predicate and subject indicating the organization's name/ID sending the information update. The predicate must be labelled www.ia.gov.ca/source, and its attached object must uniquely identify the name of the source.

The figure below illustrates a semantic web which conforms to the standards described above.

Figure 5-4 Semantic web representing an information update conforming to Information Agent rules. The required objects are filled in grey.


http://www.ia.gov.ca/484-999-999
http://www.ia.gov.ca/source
http://www.ia.gov.ca/personal#fullName
Sal Gabier
http://www.ia.gov.ca/transactionType
CompanyName
http://www.ia.gov.ca/personal#mailbox
mgabier@cs.sfu.ca
Create

6 CONCLUSION

6.1 Summary

The necessity of an active information policy is clear when we consider the fact that millions of people in North America are victims of identity theft each year. Criminals assume another persons identity in order to fraudulently perform actions or crimes in another person's name. These actions range from opening checking accounts to renting apartments. Although there have been a number of advancements in network and database security, identity theft often occurs through very unsophisticated means, such as stealing one's mail. The problem is hard to combat because the public is unaware of who has their personal information. This project therefore dealt with trying to limit the damage of an identity thief. Research has been heavily biased towards preventing identity theft and neglecting how to limit the damage and cope with the problem once the crime has already been committed.

Although there are still deep questions regarding the social impact of Information Agents, I believe this project has demonstrated that an active information policy is technically viable. RDF semantic webs can describe many different types of information and is used as the language of Information Agents. The most important result of an active information policy is not that an individual will know what pieces of personal information organizations currently have, but *who* has the information.

6.2 Future Work

Below is a list of possible directions of research.

1. Investigate the social and economic impact of an active information policy. Can trust be achieved when our personal information is being transmitted over a network to us. Even if the information is encrypted, the public may not be comfortable. There is a possibility that requiring companies to transmit information as it is manipulated will put a great strain on companies. Will our current networks be saturated with constant updates?
2. Investigate whether it is possible for individuals to "sell" their personal information via Information Agents. Individuals may mark certain items as readable so that companies can buy this data. This may eliminate the market for companies who spy on individuals and sell personal information.
3. The prototype IA has used social security numbers as unique identifiers. Organizations must be able to use these social security numbers and map them to an IA in order to send the updates. This may be done in a similar fashion to the way a message is routed using a URI. There may also be a need to use a different unique identifier.
4. Investigate the ethical issues revolving around privacy as it relates to ownership of information. For instance, if a company obtains personal information about an individual, does the company own the information, or does the individual own it because the information is about him/her. Ownership of information is a central topic when discussing an active information policy.

APPENDIX A IA SERVER FUNCTIONAL REQUIREMENTS

A.1 Introduction

This section gives a high-level description of the IAT (Information Agent Tool), a system which accepts information updates from external organizations and catalogues the information. Information Agents and their clients can then ensure the correctness of the received information.

A.2 Problem Description

An active information policy requires organizations to actively inform an individual when their personal information is created, edited, transferred, or received. The IAT acts as an intermediary between external organizations and individuals. Organizations are required to electronically send information updates formatted as RDF infosets describing the information that was created, edited, transferred, or received. The IAT catalogues the information as it is received and gives limited access to Information Agents and complete access to individuals who own the information. It is important to note that neither the Information Agent nor the individual can edit any catalogued information. If incorrect information is received, it is the responsibility of the Information Agent and the individual to correct the information of the external organization, not the catalogued information stored by the IAT.

The access rights of the Information Agent for any personal information pertaining to an individual are determined by the access rules which are created by that particular individual. Individuals can set up rules for the incoming information such as the following:

- Any information relating to money can only be accessed by the individual himself/herself.

- Medical information can only be accessed by the individual himself/herself.
- Any information containing the string “charity” can only be accessed by the individual himself/herself.
- Flag any information that is received from companies not on my trusted list.
- Flag any information that contains the string “medical”, “financial”, or “dollars”.

Based on these rules, the IA software will store the incoming information in a secure way and only allow the appropriate people/organizations to access the data.

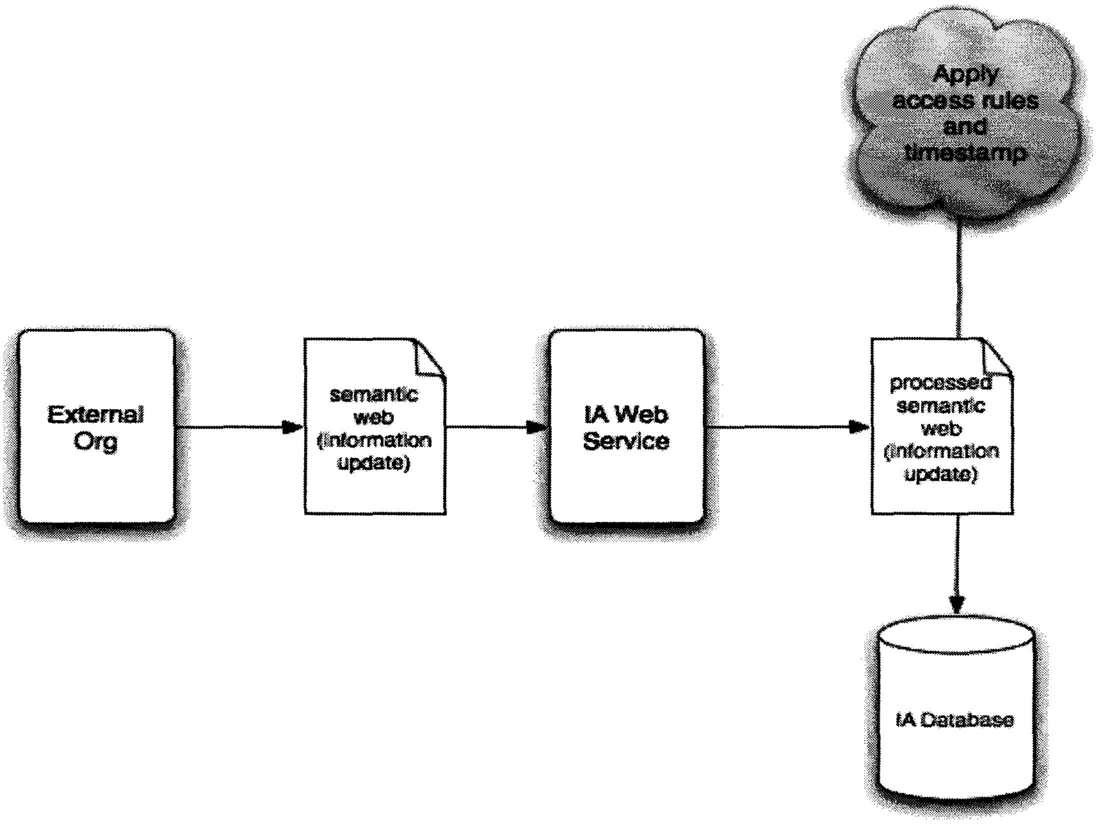
A.3 IAT Structure

The IAT consists of two main parts: a web service which interfaces with external organizations and stores the information in a secure database and a web interface which is accessed by Information Agents and clients. Both the web service and web interface are deployed on a JBoss server.

A.3.1 IAT Web Service

The IAT Web Service is build on top of the Apache Axis 1.2 framework. It provides an interface for external organizations to send information updates consisting of semantic webs detailing the manipulation of an individual’s personal information. The messages are sent over a secure HTTPS connection. As the semantic webs are received, they are parsed into a set of triples using the Jena parser (see Section 4.2). The Web Service then applies the predefined access rules and appends a permission level to each piece of information. For example, if one of the triples contains the string “medical”, a rule may specify that the triple is only viewable by the client himself/herself, however, triples containing the string “address” is viewable by both the Information Agent and the individual. The accessibility is stored as a field called the access or flag value and is determined by the access rules. An access value of ‘3’ means that only the client can view the triple, a value of ‘2’ means the Information Agent and the individual can view the triple, and a value of ‘1’ means that the triple is open to the public. There is currently no interface to the public. The triples are then time stamped and catalogued in a secure MySQL database.

Figure A. 1 – External organizations send semantic webs representing an individual's personal information to the IA Web Service. The IA Web Service parses, applies access rules, and time stamps the semantic web and enters the information into the IA database.


Apply access rules and timestamp
External Org
semantic web (information update)
IA Web Service
processed semantic web (information update)
IA Database

A.3.2 IAT Web Interface

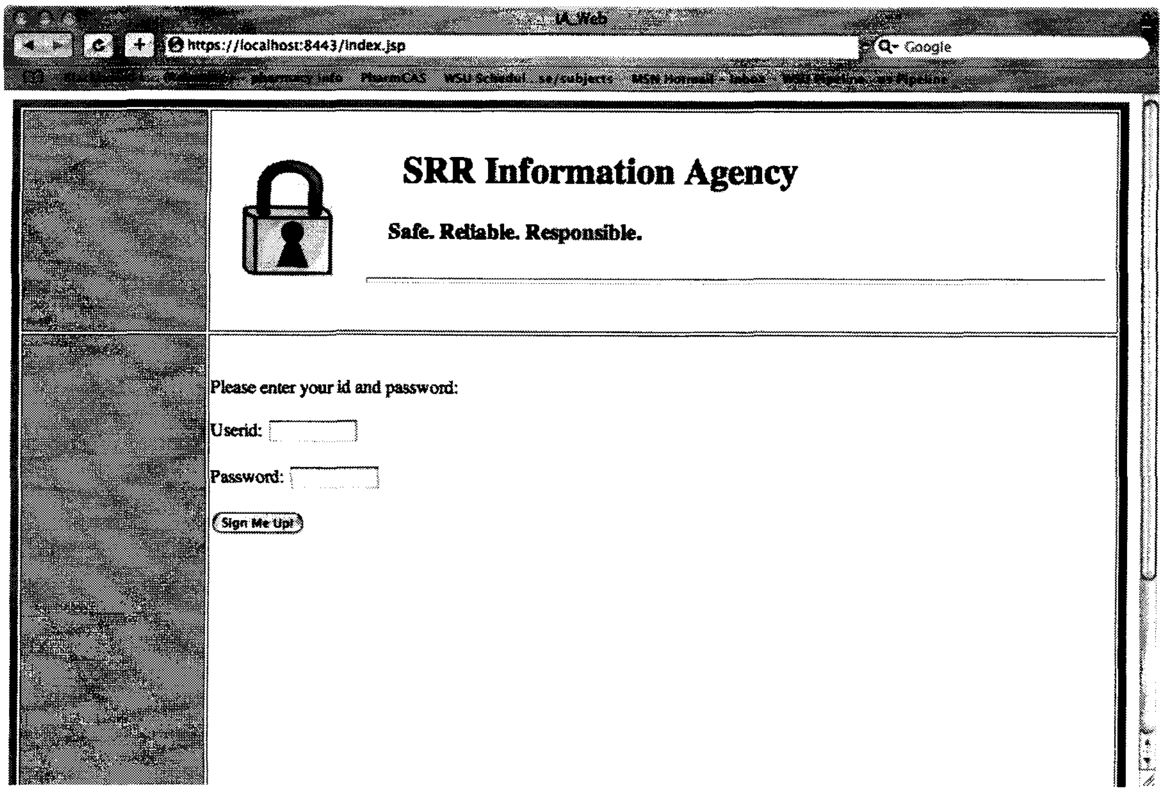
The IAT Web Interface is used by the Information Agent and the client to access the personal information received by external organizations. The website is password protected. Users log in using a secure HTTPS connection and can browse and search through all the personal information to which the user has access. Clients can access all of their personal information while Information Agents can access only selected information based upon the predetermined access rules. The rules are determined by the client and stored in a secure MySQL database.

The screenshots below illustrate the general navigation path of the website. There are two basic scenarios, logging in as a client and logging in as an Information Agent

Scenario 1: Logging in as a client.

The website requires users to login with their social security number and password.

Figure A. 2 – Users are required to login with their social security number and password.



SRR Information Agency

Safe. Reliable. Responsible.

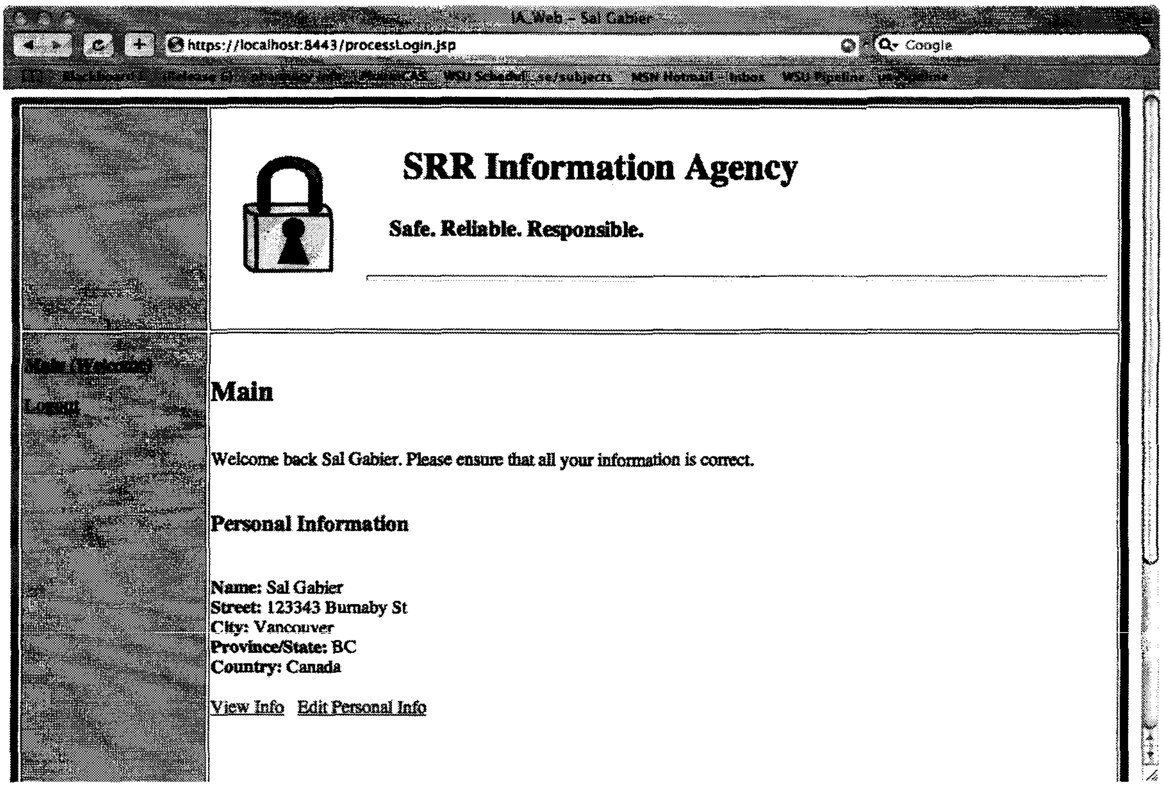
Please enter your id and password:

Userid:

Password:

After the social security number and password have been verified, the user is taken to the main page (see below). The main page displays the clients name and address, and gives the user two options, to click on "View Info" or "Edit Personal Info". The personal information that is displayed is the internal client information, not information received from external organizations. The "View Info" link takes the user to a search page that the user can use to query the catalogues of personal information received from external organizations. The "Edit Personal Info" link takes the user to a page that can be used to update internal information.

Figure A. 3 – The main page displays the client's personal information and has two links, "View Info" and "Edit Personal Info".



IA_Web – Sal Gabier
https://localhost:8443/processLogin.jsp
Google
Blackboard 6 (Release 6) MyMedCAS WSU Schedule ...se/subjects MSN Hotmail - Inbox WSU Pipeline
Main (Welcome)
Logout

SRR Information Agency

Safe. Reliable. Responsible.

Main

Welcome back Sal Gabier. Please ensure that all your information is correct.

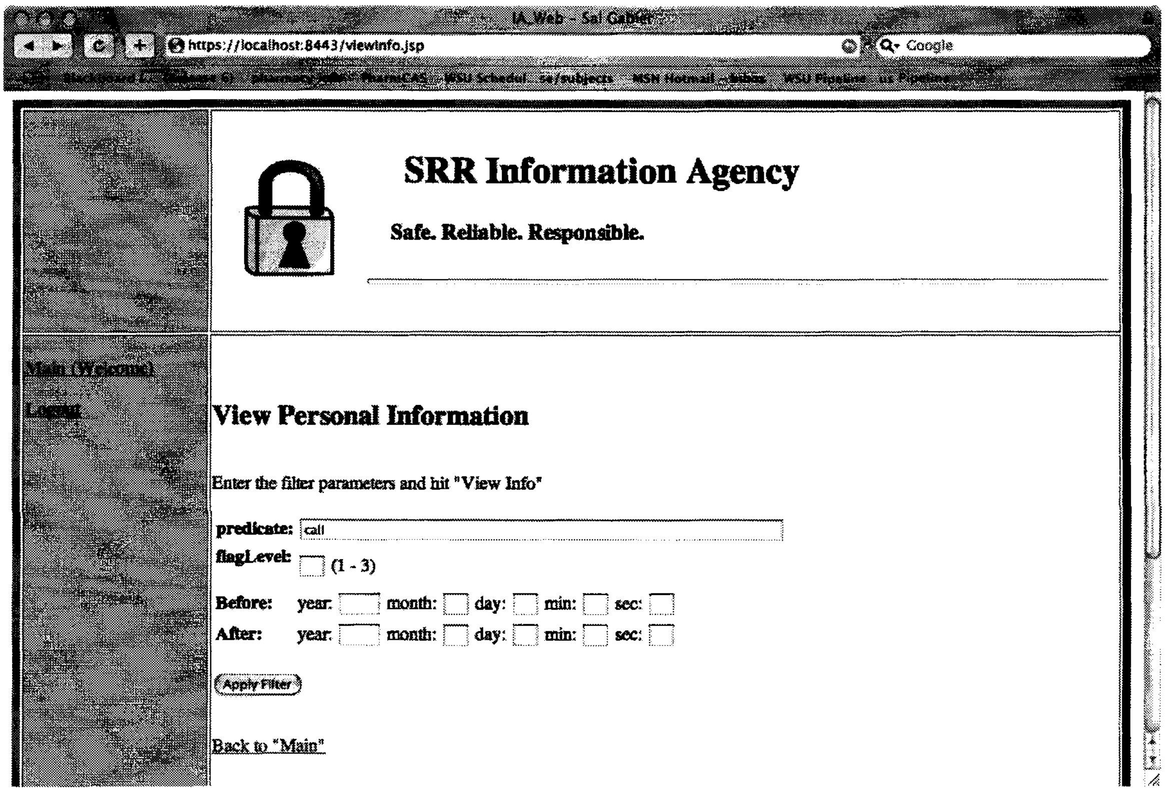
Personal Information

Name: Sal Gabier
Street: 123343 Burnaby St
City: Vancouver
Province/State: BC
Country: Canada

View Info Edit Personal Info

If the user selects "View Info", the user is taken to the search page displayed below. The user can search for information using the predicate (the second value of semantic web triples), flag (the access value determined by the access rules), and date parameters.

Figure A. 4 – View Personal Information Search Page. This page allows the user to query the catalogues of information received from external organizations



SRR Information Agency

Safe. Reliable. Responsible.

View Personal Information

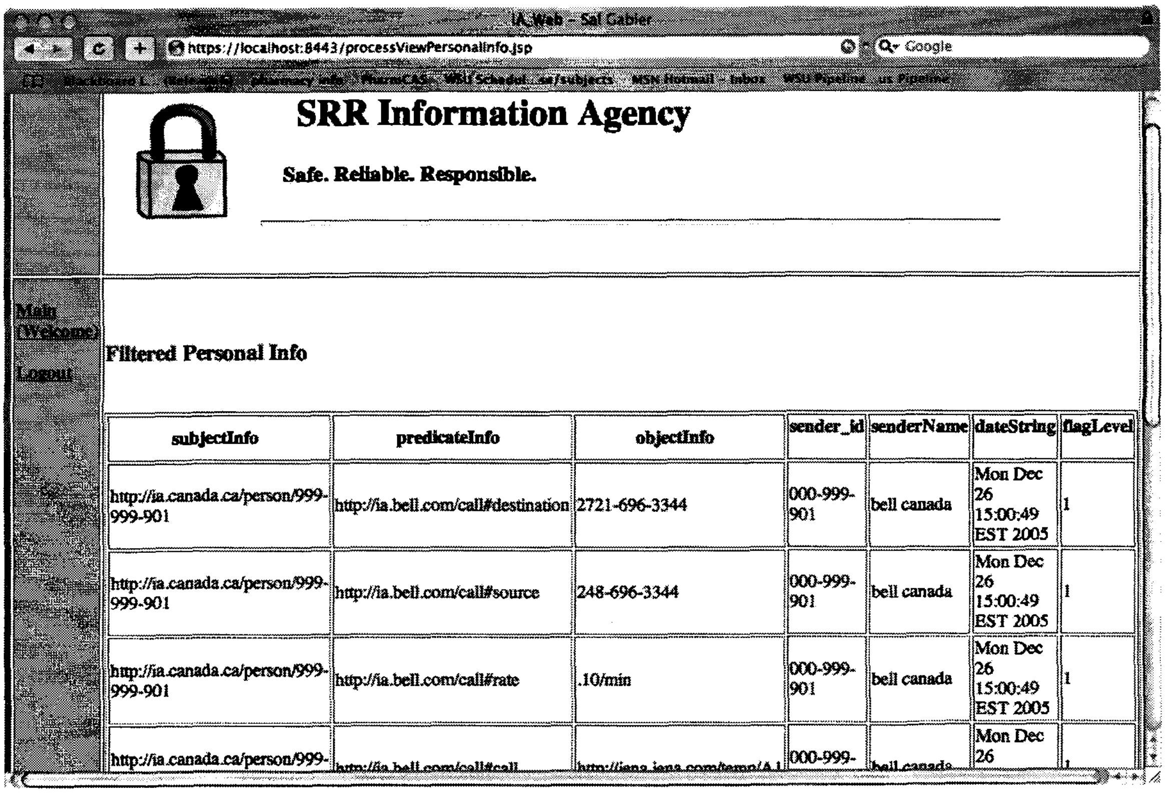
Enter the filter parameters and hit "View Info"

predicate: call
flagLevel: □ (1 - 3)
Before: year: □ month: □ day: □ min: □ sec: □
After: year: □ month: □ day: □ min: □ sec: □

Back to "Main"

When the user enters the search criteria and hits the "Apply Filter" button, the user is taken to the results page, which displays the personal information based on the search criteria.

Figure A. 5 – Search page displays the results of the search query.



SRR Information Agency

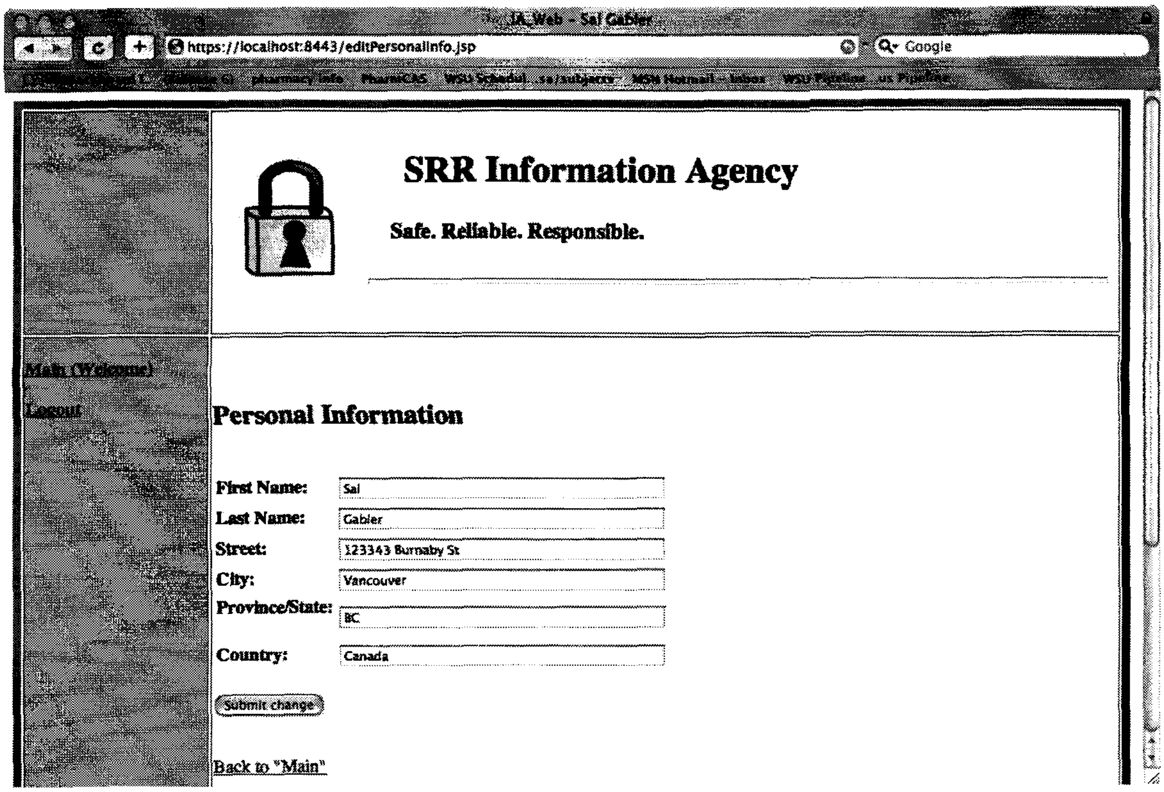
Safe. Reliable. Responsible.

Filtered Personal Info

subjectInfo	predicateInfo	objectInfo	sender_id	senderName	dateString	flagLevel
http://ia.canada.ca/person/999-999-901	http://ia.bell.com/call#destination	2721-696-3344	000-999-901	bell canada	Mon Dec 26 15:00:49 EST 2005	1
http://ia.canada.ca/person/999-999-901	http://ia.bell.com/call#source	248-696-3344	000-999-901	bell canada	Mon Dec 26 15:00:49 EST 2005	1
http://ia.canada.ca/person/999-999-901	http://ia.bell.com/call#rate	.10/min	000-999-901	bell canada	Mon Dec 26 15:00:49 EST 2005	1
http://ia.canada.ca/person/999-	http://ia.bell.com/call#call	http://iana.iana.com/temp/A.1	000-999-	bell canada	Mon Dec 26	1

The user can edit his/her internal personal information (at this point, only the address is stored) by clicking on the "Edit Personal Info" link. This will take the user to a page which allows the user to edit the information and submit the results which will update the database.

Figure A. 6 The Update Personal Information Page allows the user to update his/her internal personal information.



SRR Information Agency

Safe. Reliable. Responsible.

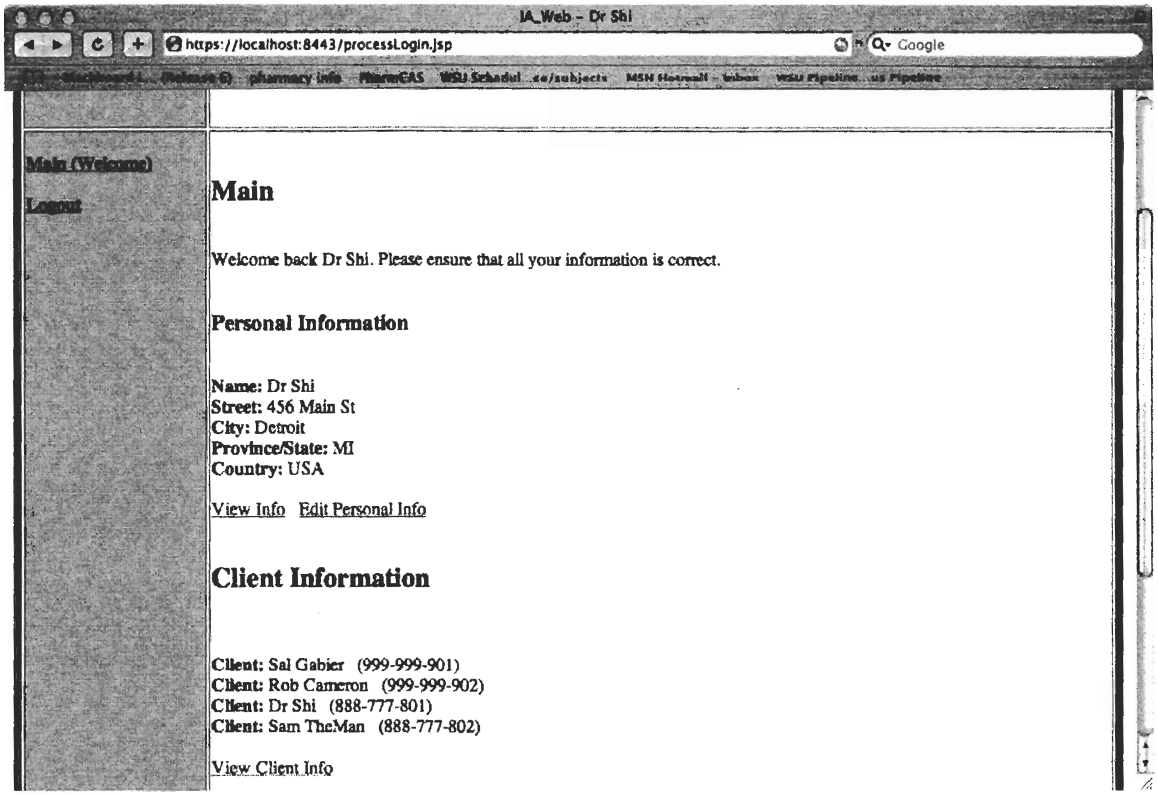
Personal Information

First Name:	Sal
Last Name:	Gabler
Street:	123343 Burnaby St
City:	Vancouver
Province/State:	BC
Country:	Canada

Scenario 2: Logging in as an Information Agent.

Information Agents login to the same page as clients (see figure A.1), and are taken to the main page. The main page differs in that it contains information about the Information Agent's clients as well as the Information Agent's personal information.

Figure A. 7 – The main page for Information Agents displays the internal personal information of the Information Agent, as well as the names of the clients. There is an additional link to search client's personal information of his/her clients.



Main

Welcome back Dr Shi. Please ensure that all your information is correct.

Personal Information

Name: Dr Shi
Street: 456 Main St
City: Detroit
Province/State: MI
Country: USA

View Info Edit Personal Info

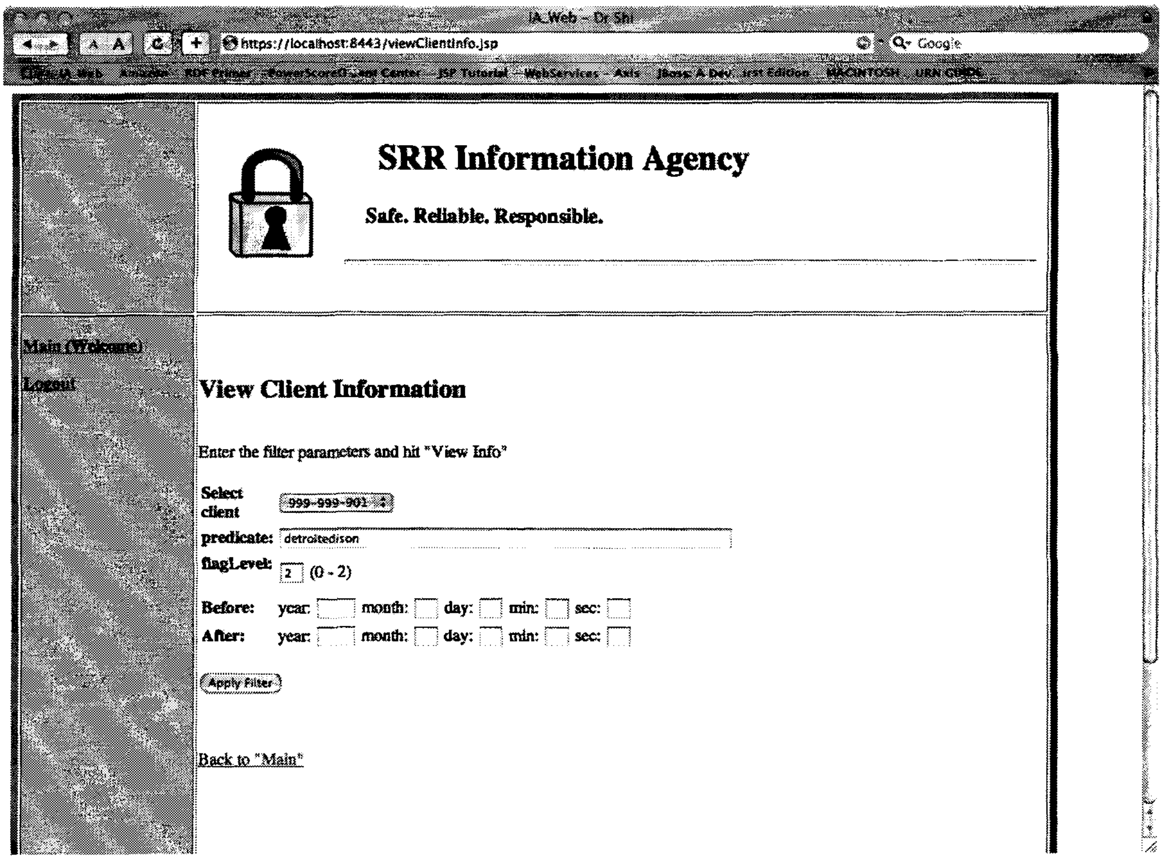
Client Information

Client: Sal Gabier (999-999-901)
Client: Rob Cameron (999-999-902)
Client: Dr Shi (888-777-801)
Client: Sam TheMan (888-777-802)

View Client Info

The "View Info" and "Edit Personal Info" take the Information Agent to the same pages as the links of the main page of the client login (see figures A.3 and figure A.5). If the Information Agent clicks "View Client Info", he/she is taken to a page where he/she can query the personal information of his/her clients. It is important to note, the Information Agent's query will never return any semantic web triples of access value '3', because those values can only be viewed by the individual himself.

Figure A. 8 - View Personal Information Search Page for Information Agents. This page allows the Information Agents to query the catalogues of their client's personal information received from external organizations. The query will not return information whose access level is '3', which indicates the information is viewable only by the individual himself/herself.



SRR Information Agency

Safe. Reliable. Responsible.

View Client Information

Enter the filter parameters and hit "View Info"

Select client: 999-999-901

predicate: detroitedison

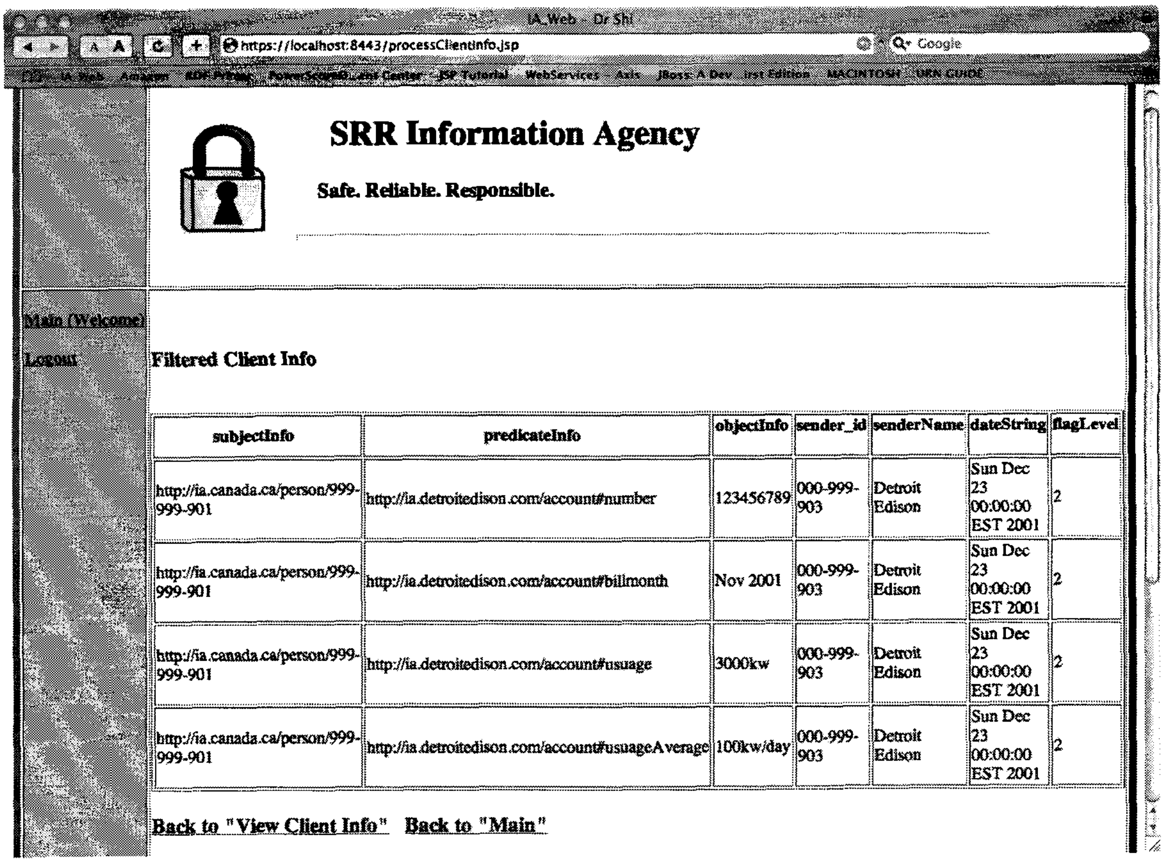
flagLevel: 2 (0 - 2)

Before: year: □ month: □ day: □ min: □ sec: □

After: year: □ month: □ day: □ min: □ sec: □

When the Information Agent enters the search criteria and hits the "Apply Filter" button, the Information Agent is taken to the results page, which displays the personal information which matches the search criteria.

Figure A. 9 - Search page displays the results of the search query.



SRR Information Agency

Safe. Reliable. Responsible.

Filtered Client Info

subjectInfo	predicateInfo	objectInfo	sender_id	senderName	dateString	flagLevel
http://ia.canada.ca/person/999-999-901	http://ia.detroitedison.com/account#number	123456789	000-999-903	Detroit Edison	Sun Dec 23 00:00:00 EST 2001	2
http://ia.canada.ca/person/999-999-901	http://ia.detroitedison.com/account#billmonth	Nov 2001	000-999-903	Detroit Edison	Sun Dec 23 00:00:00 EST 2001	2
http://ia.canada.ca/person/999-999-901	http://ia.detroitedison.com/account#usuage	3000kw	000-999-903	Detroit Edison	Sun Dec 23 00:00:00 EST 2001	2
http://ia.canada.ca/person/999-999-901	http://ia.detroitedison.com/account#usuageAverage	100kw/day	000-999-903	Detroit Edison	Sun Dec 23 00:00:00 EST 2001	2

Back to "View Client Info" Back to "Main"

A.4 Tools and Frameworks

The IA Tool was designed using a number of languages, tools, and frameworks. The list below describes those technologies.

- JBoss – open source application web server which hosts the website and web service, which both use a secure connection with a self-generated certificate
- MySQL – open source database which stores the personal information and access rules. All passwords are encrypted.
- Java 1.4 – free platform independent programming language. The entire program is written in java.
- Jakarta Ant – open source build tool used to automate compiling, archiving, and deploying the IAT.

- Jena – open source RDF (resource description framework) parser developed at HP labs, used for parsing incoming messages.
- Apache Axis – open source SOAP engine.
- Eclipse – Open source integrated development environment.

A.5 Security

Because the information that is stored is extremely sensitive, it was necessary to provide security for the IAT. Below is a list of security measures implemented for this project.

- The IAT Website and Web Service use a SSL connection with a self-generated certificate.
- The user needs to login with a password to access any information.
- All passwords are stored in MySQL and are encrypted/hashed. When the user types in his password, it is hashed and compared with the hashed value stored in the Database.
- The database can only be accessed locally. This prevents people from trying to remotely access the database.
- The database only has one user account, the application account 'iaserver'. This means that users must go through the application, instead of having direct access to the database.
- The information is only accessible by those with access rights. There is a finely grained security model which gives a security level for each piece of information. Only those of a particular level can access sensitive data.

BIBLIOGRAPHY

[1] S Ajmani. *A Trusted Execution Platform for Multiparty Computation.* MIT Masters Thesis, Cambridge, MA, July 2000

[2] E Bertino, S De Capitani Di Vimercati, E Ferrari, P Samarati. "Exception-based information flow control in object-oriented systems" in *ACM Transactions on Information and System Security (TISSEC).* Volume 1, Issue 1 November 1998

[3] A Clement. "Considering Privacy in the Development of Multimedia Communications" in *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[4] D Dean, E W Felten, D S Wallach. "Java Security: From HotJava to Netscape and Beyond" in *1996 IEEE Symposium on Security and Privacy.* May 1996

[5] H M Deitel. *Java How to Program.* fourth edition, Prentice Hall, 2002

[6] H M Deitel, P J Deitel, S E Santry. *Advanced Java 2 Platform.* Prentice Hall, 2002

[7] H M Dietel, P J Dietel, B Dewaldt, L K Trees. *Web Services: A Technical Introduction.* Prentice Hall, 2002

[8] J Dublin. *The Little Black Book of Computer Security.* Penton Technology Media, 2005

[9] Federal Trade Commission. *Identity Theft Survey Report.* October 2003. Last cited June 2004, http://www.ftc.gov/os/2003/09/synovatereport.pdf

[10] D Hatch. "Privacy: How Much Data Do Direct Marketers Really Need" in *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[11] R Helton. *Java Security Solutions.* Wiley, 2002

[12] P Herrmann. "Information Flow Analysis of Component-Structured Applications" in *Proceedings of the 17th Annual Computer Security Applications Conference (*ACSAC'2001), *ACM SIGSAC,* pages 45-54. New Orleans, IEEE Computer Society Press, December 2001

[13] C Hibbert. "What to Do When They Ask for Your Social Security Number" in *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[14] T Horowitz. "Mr. Edens Profits from Watching His Workers' Every Move" in *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[15] HP Semantic Web Programme. *An Introduction to RDF and the Jena RDF API.* Last cited Jan 2006, http://jena.sourceforge.net/tutorial/RDF_API/index.html

[16] R Kling, J P Allen. *How the Marriage of Management and Computing Intensifies the Struggle for Personal Privacy.* Last cited December 2005, http://rkcsi.indiana.edu/archive/kling/pubs/marriage.htm

[17] R P Kusserow. "The Government Needs Computer Matching to Root Out Waste and Fraud" In *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[18] K Laudon. "Markets and Privacy" in *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[19] D F Linowes. "Your Personal Information Has Gone Public" in *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[20] G T Marx. "The Case of the Omniscient Organization" in *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[21] A. C. Myers, B. Liskov. “A Decentralized Model for Information Flow Control” in *Proceedings of the 16th ACM Symposium on Operating Systems Principles*. October 1997.

[22] A C Myers, B. Liskov. “Protecting privacy using the decentralized label model” in *ACM Transactions on Software Engineering and Methodology*. Volume 9, Issue 4,October 2000

[23] A C Myers, B. Liskov. “Complete Flow with Decentralized Labels” in *Proceedings of IEEE S&P'98*. May 1998

[24] A C Myers. “JFlow: Practical Static Information Flow Control” in *Proceedings of the 26th ACM Symposium on Principles of Programming Languages (POPL '99)*. January 1999

[25] A C Myers. *Mostly-Static Decentralized Information Flow Control*. MIT Ph.D. Thesis and Technical Report 783. January 1999

[26] E Newcomer. *Understanding Web Services: XML, WSDL, SOAP, and UDDI*. Addison Wesley Professional, 2002

[27] S Oaks. *Java Security*. Second Edition, O’Reilly Media, 2001

[28] C Peikari, A Chuvakin. *Security Warrior*. O’Reilly Media, 2004

[29] T Pender. *UML Bible*. Wiley Publishing Inc, 2003

[30] PhoneBusters. *Statistics on Phone Fraud, Identity Theft Complaints*, 2003. Last cited January 2006. http://www.phonebusters.com/english/statistics_E03.html

[31] PhoneBusters. *Statistics on Phone Fraud, Identity Theft Complaints*, 2004. Last cited January 2006. http://www.phonebusters.com/english/statistics_E04.html

[32] PhoneBusters. *Statistics on Phone Fraud, Identity Theft Complaints*, 2005. Last cited January 2006. http://www.phonebusters.com/english/statistics_E05.html

[33] R Posch. “Direct Marketing Is Not a Significant Privacy Threat” in *Computerization and Controversy Value conflicts and Social Choices*. Second Edition, Edited by Rob Kling, Academic Press, 1996

[34] Privacy Rights Clearing House. *How Many Identity Theft Victims Are There? What IS The Impact On Victims*. February 2006. Last cited 2006, http://www.privacyrights.org/ar/idtheftsurveys.htm#FTC

[35] ReverseRecords.org. *Instant Self-Service Online Reverse Lookup Privileges.* Last cited February 2006, http://cell.reverselookupsonline.com

[36] N Richards, S Griffith. Jr. *JBoss: A Developers Notebook.* O’Reilly, 2005

[37] D S W Rice, A W Appel, E W Felten. “SAFKASI: a security mechanism for language-based systems” in *ACM Transactions on Software Engineering and Methodology.* Volume 9, Issue 4, October 2000

[38] M Rottenberg. “Wiretapping Bill: Costly and Intrusive” in *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[39] Royal Canadian Mounted Police. *Computer Crime Prevention.* October 2003. Last cited June 2004, http://www.rcmp-grc.gc.ca/scams/ccprev_e.htm

[40] J Shattuck. “Computer Matching Is a Serious Threat” in *Computerization and Controversy Value conflicts and Social Choices.* Second Edition, Edited by Rob Kling, Academic Press, 1996

[41] W3C. *RDF Primer, W3C Recommendation.* 10 February 2004. Last cited June 2005, http://www.w3.org/TR/rdf-primer

[42] W3C. *SOAP Version 1.2 Part 0: Primer, W3C Recommendation.* 24 June 2003. Last cited June 2005, http://www.w3.org/TR/2003/REC-soap12-part0-20030624

[43] W3C. *Web Services Description Requirements.* October 2002. Last cited May 2005, http://www.w3.org/TR/ws-desc-reqs

[44] W3C. *XML Schema: Formal Description.* September 2001. Last cited May 2005, http://www.w3.org/TR/xmlschema-formal/

[45] D S Wallach, D Balfanz, D Dean, E W Felten. “Extensible Security Architectures for Java” in *Proceeding of the 16th Symposium on Operating Systems.* October 1997

[46] S Zdancewic, L Zheng, N Nystrom, A C Myers. “Untrusted hosts and confidentiality: secure program partitioning” in *Proceedings of the 18th ACM symposium on operating systems principles.* 2001