Towards Better

Digital Identity Management

Spring 2006
Alvaro J. Gutierrez
Professor Joan Feigenbaum
Sensitive Information in a Wired World

Sensitive Information in a Wired World
CPSC 457b - Spring 2006
Alvaro J. Gutierrez

Towards Better Digital Identity Management

2. Abstract

The goal of this paper is fivefold. First, we will try to define identity, digital identity and identity management. Second, we will summarize the current status of digital identity management. Third, we will detail the specific problems and limitations of the current identity infrastructure. Fourth, we will propose a theoretical system that is designed to address the aforementioned problems. Fifth, we will present an actual framework, currently in development, which will be our final recommendation. In addition, the subject of anonymity, with respect to our recommendation, will be dealt with, but it will not be the main objective of the paper.

3. Motivation

As witnessed previously in history (e.g. during the industrial revolution), the success and continued growth of a particular technological infrastructure is never trouble free, and unfortunately the internet is no different. In particular, the internet has faced and is currently facing tremendous growing pains because of the way its foundation was built (which we will further detail later on). One of the more troublesome of these issues is identity, because of its relation to so many different aspects of the digital infrastructure (from low-level protocols to biometrics to the law).

Well, all of this having been said, the internet is out there, and it is “live.” As a whole, it is a “production” environment. It is used by literally billions of people every day. It would be difficult if not impossible to attempt to radically alter its low-level foundation to better suit some of our purposes. It might even be dangerous to do so, since it could stifle its growth and general property of openness.

A better solution would be to use current technologies, and to transition slowly towards a better system. The ultimate goal of this paper will be to describe one such solution.

4. Scope

The subject of identity, even just digital identity, and its management, is vast. It would be unrealistic to attempt to cover all of it without losing focus. Therefore, we will concentrate our efforts on certain problems and solutions.

4.1 Things out of Scope:

Specifically, there are three things which we will not give a lot of attention to, and they are:

1. Physical Authentication: What this means is essentially the process by which a terminal node in a network (practically, a *local machine*) determines that its user is the entity s/he says they are.

   Normally this process ranges from mere location (e.g. proximity and access to the machine), to simple passwords, to complex biometric schemes and smart cards and other hardware tokens [1]. Typically the effectiveness of these schemes similarly ranges from being useless to being severely flawed [2] and easily compromised [3] (e.g., with the help of gummy bears).

   The idea is to be agnostic towards this aspect of the process of identification. That is not to say that authentication is not an issue by itself (far from it), but it can be neatly separated so that the network doesn't have to deal with it.

2. Trusted Computing: This is a movement and ideology that tries to give a notion of trustworthiness to nodes in a network, often trying to do so all the way to the users' local machines.

   These kinds of initiatives have met with various kinds of resistance, from privacy advocates [4] to people concerned with efficiency [5], to people that find it convoluted and difficult to use [6]. The most prominent example of it is Palladium [7] by Microsoft, which seems to have failed miserably at the moment.

   The inroads that this kind of technology will make, especially at the fringe of the internet (i.e. the mainstream) is questionable at best. Its

success necessitates the coordination of so many parties (including hardware and software vendors) that relying on it would make for an unrealistic dependency.

3. Legislation: The reason why we will not go into the legislative and legal aspects of identity is that it is mostly ineffective. The law is slow and non-technical; therefore it can not target the problems that we will outline in this paper. The solution must be comprised mostly of technological components and must not count on the legal system.

   Additionally, the law, internationally, is complicated and often incompatible. Trying to deal with all of these incompatibilities would be untenable and probably in vain. The best that we can hope to obtain from the law is help legislating against very specific problems.

   For instance, in the United States, a good body of legislation, especially at the state level, just passed into law (in 2005) that will help curving the problem of identity theft [8]. At the federal level, there is also the Identity Theft and Assumption Deterrence Act (of 1998) [9], which at least acknowledges the problem. Once again though, this will not cause general improvement, but rather limit the impact of this one specific issue (Identity Theft). Thus we will not make use of legal solutions in this paper.

5. Introduction

In order to have a coherent discussion of identity and its digital treatment, the following three questions must be answered. The answers provided are not universal truths; they are simply local definitions pertinent to the goal of this paper.

5.1 *What is Identity?*

Identity is a concept that comes to most people very easily intuitively. Yet it is at the same time very difficult to formalize. There are also complex philosophical ramifications. It is interesting to note that identity is often defined [10] as a comparison method (i.e. in plural terms), and not as an actual distinct quality or entity. For instance, it is often said that two or more objects may be *identical*. That is to say, they share *one* identity; they are the *same*.

From this notion (or rather, its opposite), we derive the notion of human identity: two humans may not be identical, they must not share a single identity. Notice that what is absent is the formal specification of identity itself. So we got around this issue by abstracting from it and using comparison instead. One might be tempted to say certain physical traits fully determine identity (i.e. DNA), but there is the pesky issue of twins (and, very soon, clones [11]), so it's not that simple.

What this goes to show is that even in real life, there is no clear cut method for determining the identity of a person. Practically, what is often done is to use a combination of techniques; for instance, physical ID (DNA, appearance), legal ID (birth certificate, driver's license), social ID (peers, family, and friends), etc. One thing *is* clear, though: Given that this is already a problem in the real world, it will be an even bigger problem online, where many of these techniques are simply unavailable.

5.2 *What is Digital Identity?*

Briefly put: there is no such thing; at least not yet. The idea that the 'identity' of a person transfers cleanly into the digital realm is shaky at best and dangerous at worst. Instead, we define digital identity as a subset of the characteristics that define a person in the real world. The digital part comes from these characteristics being stored and possibly transferred digitally. In addition,

further subsets of this set of characteristics can be extracted, so as to meet certain identity requirements.

For instance, age can be one such subset. If a site allows only people 13 years or older to access certain material, then that site does not need to identify one as a *person*. They only need to identify a subset of this information (the characteristics), namely, the person's *age*. This is a more realistic approach to identity because it doesn't need a formalization of *complete identity* to function. We realize that this definition may be non-standard.

5.3 *What is Digital Identity Management?*

Digital ID management is very broad (especially in terms of large organizations), so for the purposes of this paper we will define it to be the interaction between a user (a human), and his or her identity information stored and maintained digitally. In other words, we will be dealing exclusively with *personal* identity management.

6. Current Problems

In order to attack the problematic aspects of digital identity, we must first classify them into digestible chunks. Some work has been done regarding this respect. For instance, Daniel J. Solove created a taxonomy of privacy [12]. However, we feel that particular framework is somewhat overwrought. Therefore, for our purposes we present a more succinct list of problems. These problems are directly related to the goal of this paper, since they will be either fully or at least partially addressed by the solution(s) proposed afterwards.

The problems are (in no particular order):

- 1. Unreliability.
- 2. Inconvenience.
- 3. Inconsistency.
- 4. Impermanence.
- 5. Insecurity.
- 6. Propagation.
- 7. Intrusion.
- 8. Intransitivity.

6.1 Unreliability

The current identity infrastructure is very unreliable. The root of this unreliability is basically the way the core technology behind the internet (and other networks) operates. The primitives that are used include machines (hosts) and packets (data). Networks work by getting data from host to host. People are not part of the picture. The reason why these primitives were used is so that the network would be as efficient and scalable as possible. Notice that the goal has been attained. The internet in particular continues to grow at an impressive rate [13] globally with basically no efficiency problems.

Nevertheless, this goal left online identity in limbo, since the best one could hope for was to correctly identify that a particular packet came from a particular machine. People are still not in the picture. Even this identification of packets (data) is problematic, because of protocol spoofing. In particular, one-way protocols such as SMTP (the de-facto outgoing mail protocol) can be trivially spoofed [14], a practice that has spawned its own crisis, nay industry, known as ‘phishing.’ [15].

In the end the result is that people can not reliably identify other people across a network, especially not across the internet. This in turn leads to poor management of identity information because machines are unsure of what information to send were. The effect is that either too much or too little information is sent; mostly too much. The worst part is that people don't even know when this information is being sent and to whom it's being sent to.

6.2 Inconvenience

This one goes almost without saying, as most users will tell you. In order to support even rudimentary member services, sites and organizations must each and all implement some form of registration scheme. People must in turn register at each and every of these sites. Efforts to centralize and automate this process, such as Microsoft Passport [16] have largely failed [17].

In addition, in most versions users must generate and "remember" a cryptographically secure (i.e. long and complicated) password in order to thwart basic brute force attacks, where the adversary systematically tries every password combination given their available resources. These username/password/email triples must be stored somewhere, which is a hassle in itself. Usually this information ends up in a post it note on a monitor, viewable by all.

Some sites require the use of CAPTCHAS [18] to stem the tide of spam. These are – "Completely Automated Public Turing tests to tell Computers and Humans Apart." The result is more pain and inconvenience for users, since they must complete these fun little tests for mostly every non-idempotent action.

6.3 Inconsistency

In addition to the inconvenience, there is also the issue of inconsistency. Because there is no standard for developers to follow, registration and identification schemes vary wildly. The differences can be simply cosmetic (e.g. GUI colors), or fundamental (e.g. deletion of one's account).

Usually, every language / server pair has its own "framework" for identifying users (e.g. PHP's sessions [19]). The functionality available from frameworks is the subject of numerous articles (usually aimed at developers), so it is troublesome for the user to be expected to be conversant about the features available here or there. Greater consistency is greatly needed.

6.4 Impermanence

By impermanence we mean that identity, even within one site or organization, does not usually last more than a fixed period of time. The time-related functionality mainly varies across frameworks. Some let the user set how long the user will remain "logged-in," generally the options do not exceed 2 weeks (e.g. eBay [20]). Others do not allow this kind of functionality and limit the time to a hardcoded duration (e.g. Yahoo Mail [21]). Yet others will not allow for permanence of any kind and will log users out immediately upon the closure of the browser session (e.g. Yale Webmail/CAS [22]).

The common element is that identity at all of these systems *expires*. It is not *permanent*. This notion is unintuitive in real life, since whatever we define identity as, it tends to either not expire at all (e.g. Birth Certificates) or expires after very long periods of time (e.g. Drivers' Licenses). Digital ID, on the other hand, is very ephemeral and can be lost merely by closing a window at times. Ideally, we would like to be able to move away from this set up, which is the result mostly of previous security failures.

6.5 Insecurity

Basically, the current infrastructure is insecure. The main reason is that, even with a lot of variation, systems typically rely on browser 'features' such as cookies, basic HTTP auth and query strings to function, most of which are insecure. For instance, cookies are vulnerable to XSS attacks.

Moreover, even when there is the possibility of security, there is a lot of human error. People choose weak passwords, people lose those passwords, people compromise their local machines (Trojans or Keyloggers, or other viruses/spyware/malware). Examples of large-scale insecurity abound: The Yale SSN fiasco [23], the Lexis-Nexis leaks, etc.

On top of that, many systems are static and can become stale. It's no surprise that protocols and ciphers become outdated and breakable. For instance, SSL1, MD4 (and possibly MD5), etc. If a system has no update mechanism, it will eventually be vulnerable to attacks via obsolete technology.

6.6 Propagation

The current identity infrastructure is set up to allow for vast propagation of sensitive information. In the best case this simply leads to the discomfort of not knowing who possesses what information. In the worst case, serious information leaks occur which can in turn lead to crimes such as identity theft.

Therefore, because the system permits such easy propagation, it is also very prone to leaking. We can look at the identification process more or less as a pipeline, with the point of entry being the original party and the point of exit (if there is one) being the last party concerned with any part of the identity information. This kind of linear structure is vulnerable to weakest link attacks.

If any of the parties along the pipeline does not treat the information carefully, then the information protection of the whole system fails. There are many reasons why this may happen. Parties may be incompetent, without proper security measures, or they may be actively malicious, trying to funnel in as much private information as possible for illegal purposes.

Here's an example of the pipeline, using Amazon.com [24], a reputable online retailer. This is a case where the user utilizes Amazon's Z-Shops feature [25], which allows third-party vendors to access the Amazon purchasing and inventory frameworks.

[User] → [Amazon] → [Third-Party Vendor] → [CC Proc.] → [Bank] → [Shipper]

→→→ (Private/Sensitive Information) →→→

Notice that Amazon itself need not be malicious nor incompetent for leaks to happen, but they are simply the point of entry. Any party afterwards could be, and in the case of third-party vendors, that's likely to be the case. In our example, it is quite possible for the original information to have propagated fully to all of the parties along the pipeline, something that is completely unnecessary.

In essence, the infrastructure relies on "corrective" measures. The paradigm is "assume there will be no leaks", then "secure the leaks if there are any." A better paradigm might rely on "preventive" measures instead, so that leaks are less damaging if and when they happen (and easier to track down).

6.7 Intrusion

Intrusion is, in a few words, the effect of involuntary actions upon the user. Intrusion in the real world can come to encompass a cornucopia of things, but in our context, it generally just means unwanted communication.

The opposite of intrusion (its solution, in a sense), is *participation*. If a user is allowed to choose whether s/he wants to *participate* in a given communication, then there is no intrusion.

The best example of intrusion is email spam. Spam is very problematic because its sources are so varied. Spam can be not only commercial, but also political or religious. In its worse form, spam is ambiguous and/or anonymous, so that the user can't even vaguely determine who the intruder is.

Digital intrusion can result in privacy violations or even severe security failures. Spam, in HTML form, can contain hidden HTTP requests that can track a user to see if they have opened the email message. This is a form of a privacy violation. More malicious versions of spam, exploiting common security vulnerabilities [26] in email clients (such as Outlook [27]), can install all sorts of malware (spyware, adware, etc.) that can completely surrender a user's machine or even local network to the adversary; clearly this would entail a security failure.

6.8 Intransitivity

What we are referring to in this case is the lack of a connection between the ID of a user at location A and their ID at location B, where locations can be web sites, or online video games, etc. Usually, after going through all of the hassle of creating an account or identity at one location, it is completely useless everywhere else. We can hence call these IDs intransitive. They do not transfer across domains, games or sites, except for those within the same organization (e.g. Google's Gmail [28] and Answers [29]).

Therefore, whatever characteristics a user inputs or accrues in one place are not meaningful anywhere else. This leads to either redundancy (having to input the same thing more than once), or frustration (my character in game A won't work in game B). It would be a great improvement to allow for characteristics to transfer, even if not fully. For instance, if 'Reputation' is a common characteristic of two games, but not 'Experience' nor 'Wealth,' then the games could discard the former two and make use of the latter. This kind of transitivity would vastly improve the user experience.

7. A Theoretical Solution

7.1 The Client

As we have demonstrated previously, the concept of a person, a user, does not feature prominently in the current identity infrastructure. Typically the things that matter are the server, the language used, the site, the host, the speed, or some other peripheral factor. This is a big problem and leads to various online ills, which we have also detailed. In order to remedy the situation, we must put the user, whom we will call a “client,” at the very center.

It is the user who must be in control, as much as possible, of the flow of information. It is in this spirit that we introduce the idea of Client-Side Transactions, and later, Client-Side Personas. The use of Client-Side means that control is given to the Client whenever possible. The concept is not tied to the use of a *local machine*, since this would be a serious limitation, as these can be problematic themselves. Notably, a local machine could go offline while a transaction needs to be completed, or, a machine might not be available to its user when the user is not physically next to the machine, for example.

In order to get around these last issues, we also introduce the idea of a *naïve server*. This is simply a way to replicate local machine functionality across servers. The information would be encrypted, so the servers would have no idea what the information they are relaying is. Servers use bandwidth, and bandwidth costs money, so this functionality would either have to be paid for, or donated by gracious philanthropists or governments.

7.2 Client-Side Transactions

The philosophy behind client-side transactions (CST) is to minimize information required, of a client, to complete a transaction. Typically, when a user needs to participate in a given transaction, they send all information required by all parties concerned to a single party (and point of failure). This party then propagates the information as it sees fit to the other parties.

Instead, client side transactions let the user interact directly with all parties concerned, send each one the minimal set of information needed (thus segregating it), and lets the parties then communicate pseudonymously (of the user).

For instance, in a typical scenario a user will buy something from a vendor. The user sends the vendor shopping cart information, shipping information and financial information (to purchase, ship, and pay for products, respectively). The vendor then contacts the payment system with the financial details, and acquires funds for the transaction.

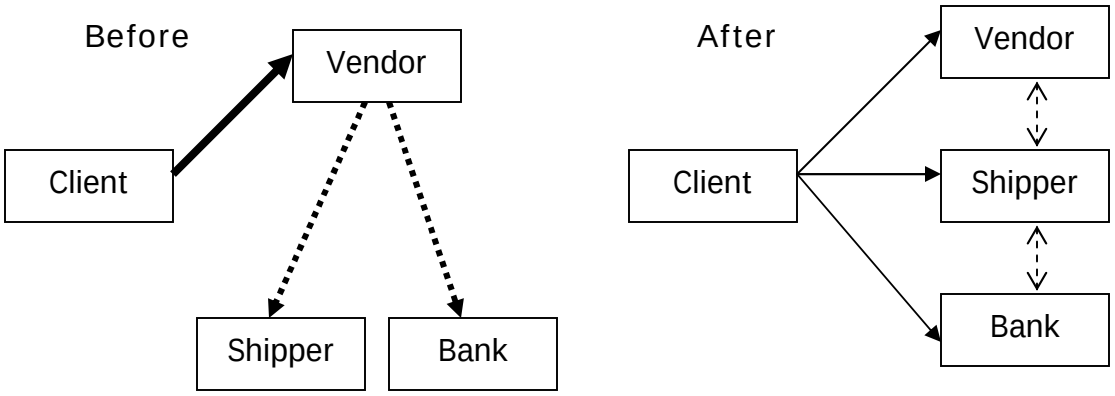
Then it uses to the shopping cart information to select items from its inventory and set them up for delivery. Finally, it uses the shipping information to tell the shipper where to send the items purchased. The single point of failure here is the vendor. If it decided to misuse the user's data in any way, it already has access to all of it by default, so it can do a lot of damage.

In the CST model, the user contacts the shipper, financial institution and vendor separately and arranges for the three to coordinate their actions based solely on unique identifiers (GUIDs [30], for instance). These identifiers, which are randomly generated, serve merely as pseudonyms. They contain no other personal information.

So the vendor receives a payment GUID and a shipping GUID from the financial institution (e.g. a bank, or credit card company), and shipper, respectively. It doesn't even need to receive the buyer's name for the transaction to go through. The vendor extracts funds from the payment GUID and slaps the shipping GUID on the packaging. It knows nothing of the user. The shipper knows nothing of the payment or of the shopping cart. The financial institution knows nothing of the shopping cart or of the shipping location. The information has been segregated. Instead of a single point of failure, the vendor can now only do minimal damage with a minimal amount of information (namely, a list of items).

One could argue that this is a lot of work on the part of the user, but in practice the whole thing could be easily automated locally via the GUI. Automatic pseudonym generation can be easily achieved [31], and routing can be done via email, for instance, so all the user would need to do is "approve" or "deny" a particular transaction, or a "set" of transactions for a complete purchase.

A picture is worth 1024 words, so...



7.3 Client-Side Personas

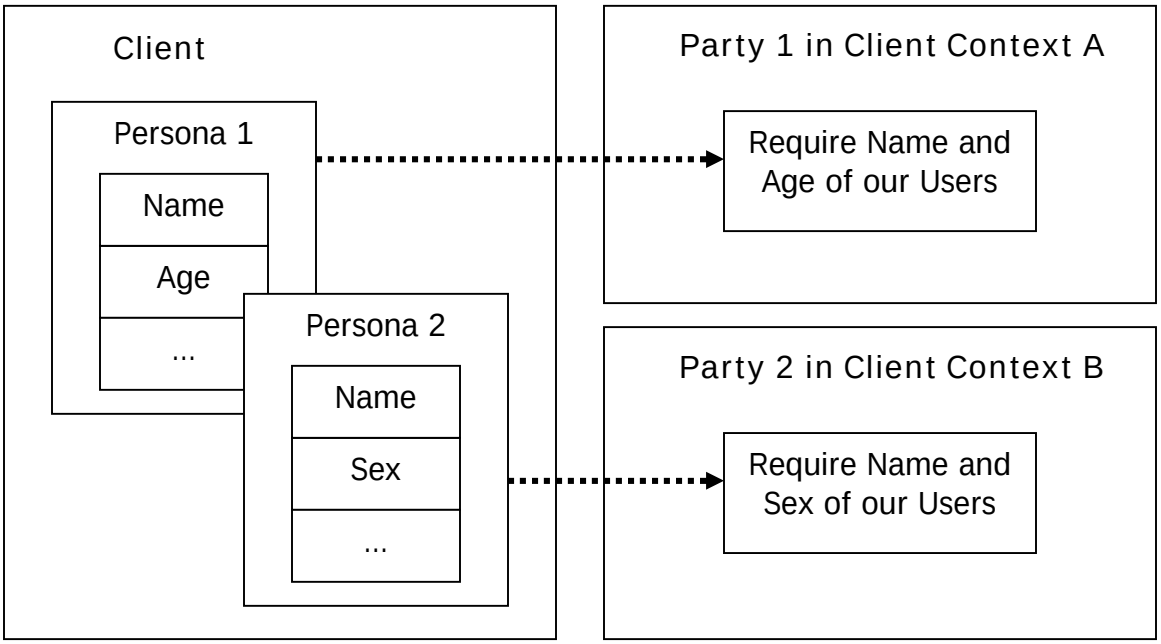
Personas are merely an extension of the concept of a transaction to allow for maintaining state locally. Transactions so far are stateless. That is, a transaction has no idea of any other transaction or any other piece of data to which it is not directly related via a particular pseudonym.

What we can do is to allow users to store identity characteristics client-side as well (at the local machine, or in one of the naïve replicating servers). The set of these characteristics, together, can be said to be a “persona” and can be grouped via the interface as such.

This would allow clients to maintain one or more personas with useful information, so that whenever a transaction needs to occur, the client doesn’t need to re-introduce the same information into the system. In addition, a persona can be kept for each different context (e.g. social context) of the client, thereby further separating the amount of information that could potentially be linked together.

The final piece of the puzzle is to allow the client to create a persona with no characteristics at all, which would serve as an anonymous entity, with which the user could participate in online activity revealing no personal information save for his or her IP address; this address can itself be hidden by using proxies or umbrella (DHCP/NAT) IPs (such as free wireless access), or internet cafes.

Here’s another example diagram:


Client
Persona 1
Name
Age
...
Persona 2
Name
Sex
...
Party 1 in Client Context A
Require Name and Age of our Users
Party 2 in Client Context B
Require Name and Sex of our Users

8. Recommendation: Microsoft InfoCard

Microsoft InfoCard [32] is a new set of technologies that aim to fix (some of) the current problems with digital identity management. InfoCard is very new in fact, and it hasn't even reached release status for version 1. Nevertheless, we believe it combines the right functionality and security features to warrant recommendation.

In short, InfoCard is Microsoft's implementation of the W3C's [33] WS-* specification. This spec is based completely on open standards, and could be evaluated as such, but we will focus on InfoCard because it is the most full-featured (and arguably working) client implementation at the moment. It is expected and encouraged [34] for other OS and browser platforms to implement the spec so that the system succeeds.

The technical details behind WS-* are outside of the scope of this paper. They will be covered by Will Tsui's own paper on the subject [35]. We will deal here only with InfoCard's benefits, limitations, possible extensions and its relation to anonymity.

8.1. Problems Solved by InfoCard

We believe InfoCard solves at least partially at the problems mentioned before.

- 1. Unreliability: InfoCard is a reliable platform for identification and communication. It is based on open standards and if implemented well, should be pose no reliability issues. At least, not anymore than general Internet reliability. However, it still depends on a company's non-core competency (namely, Microsoft's), so it is not a panacea.

- 2. Inconvenience: InfoCard gets rid most of the hassle involved with creating accounts and remembering all of the pertinent information.

- 3. Inconsistency: There is only one InfoCard interface, and it works exactly the same across sites. No more looking around to see what works and what doesn't.

- 4. Impermanence: By bypassing the browser based session mechanism, InfoCard sessions can last as long as needed without requiring explicit re-identification. If a token expires, it can be renewed automatically behind the scenes indefinitely.

- 5. Insecurity: The WS-* system is based on cryptographically secure protocols. Moreover, because InfoCard is integrated with the OS, it can update itself and keep itself secure with little user intervention.

- 6. Propagation: InfoCard helps diminish propagation with its system of claims and assertions. If parties comply with the system and demand only as little information as needed, then there will be minimal propagation. It is still up to the relying parties to do this, though. But at least the possibility for this system to work is there.

- 7. Intrusion: InfoCard can't prevent general intrusion (e.g. spam), but it can help with promoting participation. Whenever InfoCard is used, a claim can be introduced to support the kind of checks that each person may want. For instance: political affiliation.

- 8. Intransitivity: Though not implemented yet, InfoCard supports what is known as ID federation, which will allow users to "connect" their identities from one site to another with minimal loss and minimal redundancy.

8.2. Limitations of InfoCard

One potential problem for InfoCard is that it could instill a false sense of security upon users. InfoCard does not deal with the issue of Trust directly, so it is still partly up to users to deal with it. If they ignore it, then they will be at danger just like before. The issue of Trust is important. For instance, if a user trusts an unscrupulous vender with their data, even via InfoCard, then there will likely be violations.

In addition, it might be difficult to reconcile organizational IDs (given by large ID providers) and weak, self-issued IDs. More than a technical issue, this is a semantic one, so it will be interesting to see how it plays out between users and relying parties.

Finally, InfoCard relies heavily on the OS. It is deeply integrated so that it can use protected memory, the modal interface, etc. This kind of integration has resulted in various security problems in the past (e.g. Internet Explorer), so Microsoft will have to be very careful about its implementation.

8.3. Extensions to InfoCard

First, it would be nice if InfoCard natively supported CAPTCHA claims, especially for self-issued cards. Otherwise automated attacks would be too easy to perform.

Second, InfoCard should build-in some kind of Trust infrastructure, so that users know who they are communicating with. The information should be presented clearly via the interface; the data could come from the many organizations that specialize in that type of thing: Consumer Reports, the BBB, community sites, government databases, etc. The idea is to introduce some sort of “trust provider” into the system.

Third, it would be great to see the system be peer reviewed before it is fully launched (for instance, via the use of Betas). There are many people specialize in security (for instance, Bruce Schneier [36]), who could easily spot weaknesses before they reach the general public.

8.4. InfoCard and Anonymity

InfoCard does not address anonymity. The goal of InfoCard is roughly the opposite – a good Identification system. However, InfoCard does allow a sort of anonymity. When creating so-called “self-issued” cards, there is nothing stopping users from filling in completely bogus information to satisfy whatever claims a site requires.

On the other hand, this is problematic. For instance, having one such card for all anonymous purposes is at best pseudonymous, not anonymous, since the information (bogus or not), is the same across the sites where it was used. In addition, bogus cards are inconvenient – for the users, who need to create and maintain them, as well as the sites, which need to deal with them.

Moreover, bogus cards are still traceable and can be logged at the network level (i.e. via their source IP addresses).

This does not need to be the case. InfoCard could easily support anonymity by offering simple interface improvements. The following are all independent suggestions; they do not rely on each other.

First, allow automatic client generation of bogus cards. This way the card used is completely different from any others used before, therefore two or more could never be linked together.

Second, allow sites to accept anonymous users via InfoCard, in particular those sites that accept self-issued cards. This way, users would not need to generate bogus cards for those sites. An anonymous user can be implemented simply as one without any claims (or ‘assertions’).

Third, the interface should allow InfoCard-level granularity over proxy control. The problem (as specified before) is the IP address, which doubles as a smoking-gun of sorts.

The idea for a solution is for users to be able to set different proxy settings for different InfoCards. With this capability, users could use their regular internet connections for most of their sites, but could set up certain cards to go through anonymous proxy chains or specific anonymity-oriented networks such as Tor [37] via Privoxy [38]. The point of this feature is so that this kind of use would not disrupt (or slow down) the user’s other cards that don’t require proxy support.

References

[1] Hardware Tokens:
http://www.usenix.org/publications/library/proceedings/ec96/summaries/node2.html

[2] The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments:
http://www.cs.utah.edu/flux/fluke/html/inevitability.htm

[3] Gummy bears defeat fingerprint sensors:
http://www.theregister.co.uk/2002/05/16/gummi_bears_defeat_fingerprint_sensors/

[4] Windows Security Gain or Privacy Pain?
http://news.com.com/2100-1001-938957.html

[5] Microsoft Scheme for PC Security catches flak:
http://www.eetimes.com/story/OEG20020715S0033

[6] Microsoft Expects Slow Adoption for Palladium:
http://www.networkworld.com/news/2003/0508microexpec.html

[7] Microsoft Palladium (Next Generation Secure Computing Base):
http://www.microsoft.com/resources/ngscb/default.mspx

[8] Criminal Identity Theft:
http://www.privacyrights.org/fs/fs17g-CrimIdTheft.htm

[9] IDENTITY THEFT AND ASSUMPTION DETERRENCE ACT:
http://www.ftc.gov/os/statutes/itada/itadact.htm

[10] Identity Definition:
http://dictionary.reference.com/search?q=identity

[11] How Human Cloning Will Work:
http://www.howstuffworks.com/human-cloning.htm

[12] A Taxonomy of Privacy:
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=667622

[13] Internet Growth Charts:
http://navigators.com/stats.html

[14] Spoofed/Forged Email:
http://www.cert.org/tech_tips/email_spoofing.html

[15] Phishing:
http://www.computerworld.com/securitytopics/security/story/0,10801,89096,00.html

[16] Microsoft Passport:
http://www.passport.net/

[17] Microsoft Revokes Passport Service:
http://www.theregister.co.uk/2004/12/30/ms_ends_pass/

[18] The CAPTCHA Project:
http://www.captcha.net/

[19] PHP's Sessions:
http://www.php.net/session/

[20] eBay:
http://www.ebay.com/

[21] Yahoo Mail:
http://mail.yahoo.com/

[22] Yale Webmail / CAS:
https://www.mail.yale.edu/

[23] The Yale SSN Fiasco:
http://www.yaledailynews.com/article.asp?AID=19454

[24] Amazon:
http://www.amazon.com/

[25] Amazon's Z-Shops:
http://zshops.amazon.com/

[26] Microsoft Outlook Vulnerabilities:
http://www.frsirt.com/english/product/2277

[27] Microsoft Outlook 2003:
http://office.microsoft.com/en-us/FX010857931033.aspx

[28] Google Gmail:
http://www.gmail.com/

[29] Google Answers:
http://answers.google.com/

[30] What is a GUID?
http://www.webopedia.com/TERM/G/GUID.html

[31] UUID (GUID) Generator on the Web:
http://kruithof.xs4all.nl/uuid/uuidgen

[32] WinFX Home: InfoCard
http://msdn.microsoft.com/webservices/infocard/default.aspx

[33] W3C Web Services:
http://www.w3.org/2002/ws/

[34] MIX06: InfoCard:
http://channel9.msdn.com/Showpost.aspx?postid=165297

[35] Will Tsui's Technical Paper on InfoCard
<not online currently>

[36] Bruce Schneier:
http://www.schneier.com/

[37] Tor: An anonymous Internet communication system:
http://tor.eff.org/

[38] Privoxy (A web proxy) Home:
http://www.privoxy.org/