ANALYSING POWER CONSUMPTION OF DIFFERENT BROWSERS & IDENTITY MANAGEMENT SYSTEMS IN MOBILE PHONES

Md. Sadek Ferdous[1] and Ron Poet[2]

[1]School of Computing Science, University of Glasgow, Scotland
m.ferdous.1@research.gla.ac.uk
[2] School of Computing Science, University of Glasgow, Scotland
ron@dcs.gla.ac.uk

ABSTRACT

Currently there are many different Identity Management Systems which differ in their architectures as well as use different protocols and serve different purposes and are extensively used by organisations to provide online services. With the remarkable growth of mobile phones in the last few years, both in number and computational power, more and more users are accessing an array of online services using their mobile phones. One of the major concerns for the user of mobile phones is the battery life which is limited and tends to run out quickly. Hence, efficiency in power consumption is a crucial factor for any system when it is accessed using a mobile phone. In this paper, we analyse the efficiency, in terms of power consumption, of different browsers in mobile phones and different Identity Management Systems when the mobile phones are used to access online services protected by those Identity Management Systems.

KEYWORDS

Identity Management System, Power Consumption, Mobile Phone Browser, Android

1. INTRODUCTION

Currently there are literally thousands of websites around the world providing a plethora of different services via the Internet. Originally, the protocols for digital communication were mainly designed to exchange information efficiently and reliably and the web and web-based services were not foreseen in its current form. At that budding stage, the identities of communicating parties could be assumed, and there was no need to verify it formally. It led to the omission of an Identity Layer which could be used for formal verification of Identity [1]. As the web and web-based services started to evolve, verification of identity became a crucial part as Service Providers (SP, in short; the administrative body that offers and provides any service) need to identify users, to provide correct services and only to the authorised users. To adjust the situation, the process of authentication was subsequently added to verify the correctness of claimed identities. The authentication process requires users to register to generate or retrieve required identities which are usually accompanied with a security token or credential. A credential, in the context of an Identity Management System, is a shared secret between a user and a credential provider and is usually used by the user to assert as the legal holder of the corresponding identity. Users need to supply their identities along with the credential to access any service. Soon such authentication processes became an integrated part of the web service. With the tremendous expansion of the Internet during 90s, web-based services started to become a part and parcel of our day to day life and it allowed mushrooming of thousands of websites and web services. As the number of web-services as well as the user-base was expanding rapidly, more and more identities and credentials were issued, and soon their

DOI : 10.5121/ijdps.2012.3203

management became challenging, both for service providers and for users. Identity Management (denoted IdM thereafter) was introduced initially by the industry to facilitate online management of user identities. Different groups worked on their own projects resulting in various incompatible IdM solutions. Additionally, several research initiatives in Academia were undertaken leading to different prototype IdM solutions.

Simultaneously, we have experienced another trend expanding in a remarkable rate especially in the last few years: the consumption and the usage of mobile phones. There are currently 5.3 billion mobile subscribers around the world which happens to be 77% of the world population [2]. In an average, there were 1388.2 million handsets sold worldwide in 2010 with smartphones displaying the strongest growth. Currently they represent 13% of the total mobile phones worldwide [3], however, it has been predicted that there will be 631 million smartphones sold by 2015 [2]. Almost all these smartphones are being used to access the Internet which contributed to increase the global mobile data traffic up to 159% in 2010 and the forecast is that the global mobile data traffic will increase 26-fold between 2010 to 2015 [3]. Another compelling fact is that smartphones are not only growing in numbers but also in capacity in terms of processing power and memory with capacitive touch-sensitive High Definition screens getting and bigger and brighter every year. With all these advanced hardware and user-friendly software, it can be safely stated that people will be using their mobile to access an array of different online services more and more in near future.

However, there remains a bottleneck. All these powerful pieces of hardware need significant amount of power that comes from the battery which is limited in size and capacity, severely restricted to reduce the overall size and weight of the mobile phones. Therefore, an efficient management of power is extremely crucial for any applications that are being used in these smartphones. This brings us to the theme of this paper. Since more and more smartphones will be used to access a range of different online services and all these services will utilise an Identity Management System, we want to study and investigate the impact of using different Identity Management Systems on the power consumption of a mobile phone when it is used to access such services by different browsers. We are particularly interested to find out if using a certain mobile browser will have its impact on the power consumption and how it differs quantitatively in using different Identity Management Systems to access the same set of services.

With that said, the paper is organised as follows. We will provide a short introduction to Identity and Identity Management System in Section 2. We will briefly describe our chosen Identity Management Systems in Section 3. We will explain our experiment in Section 4 and present our data and analysis in Section 5. We will discuss our findings in Section 6 and related works in Section 7. We will conclude with some future research directions in Section 8.

2. Identity, Identity Management & Other Related Topics

2.1 Identity

Different disciplines (Philosophy, Social Science, etc.) interpret identity in different ways. There are also different definitions of identity which can be quite complex to understand and sometimes even contradictory. Putting aside the contradictory arguments, a simple but intuitive analytical definition can be provided [4]: Identity is the fundamental property of any entity (a physical or logical object which has a separate distinctive existence either in a physical or a logical sense [5, 6]) that declares the uniqueness or sameness of itself and makes it distinctive from other entities in a certain context (encompassing domain or environment under which an entity exists and/or operates). Every entity has a set of attributes (representation of a trait, feature or characteristic of an entity within a context [7]) related to itself. Some of these attributes are self-created or self-assigned and others are given by third parties (e.g. Family, friends, Government and other authoritative bodies). When a subset of this set is sufficient

enough from an observer's point of view to uniquely distinguish an entity from other entities in a specific context, that subset can be loosely (and commonly) defined as the identity of the entity within that context. However, *Partial Identity* would be a more appropriate term for such a subset. That is, Partial Identity can be defined as the subset containing the minimum number of attributes which can be used to uniquely identify an entity within a context from an observer's point of view. Then, the Identity of an entity is the union of all partial identities covering all possible contexts. It is important to note here that, from this point on, when we will mention the term identity within a context, we will actually imply partial identity.

2.2 Identifier

An Identifier is an attribute or a set of attributes that can be used to uniquely identify an entity within a context [8]. An entity may have many attributes associated with itself. However, the most representative attribute or set of attributes among those can be defined as the Identifier of that entity. It is crucial to note the degree of confusion between identity and identifier in common language usage. The term "identity" and "identifier" are often used interchangeably; however, a fine line of separation exists between them in reality. We will use the terms with their distinctive meanings throughout this paper.

2.3 Identity Management

A large number of online services combined with a large number of users that access each service results in an even larger number of digital identifiers with their corresponding credentials that need to be managed. Formally, Identity Management consists of technologies and policies for representing and recognising entities with their digital identities [9]. A system that is used for managing identity is called an Identity Management System (IdMS, in short).

2.3.1 Entities in Identity Management System

Each Identity Management System has several parties involved which are:

- **Service Provider.** A service provider usually provides services to the clients or to the other service providers. It is also known as *Relying Party* (RP, in short).
- **Identity Provider.** An identity provider (IdP, in short) provides digital identity with related digital identifier to the clients to enable them to receive services from a service provider.
- **Client/User.** A client/user receives services from a service provider. To receive any service, the client usually needs a digital identity and related credential to authenticate her as a valid user of that service.

2.3.2 Identity Management Model

There are currently different types of Identity Management Systems available. These systems belong to different Identity Management Models. Figure 1 illustrates a high-level taxonomy of these models [10].

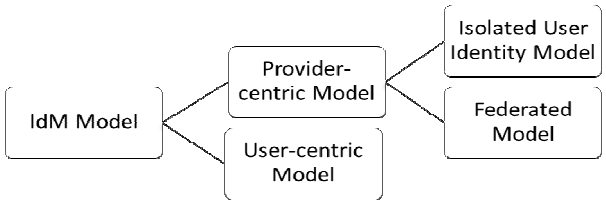


Figure 1: High-level taxonomy of Identity Management Models

We explain briefly each of these models with examples of corresponding systems below:
Provider-centric Model: In a Provider-centric model, service providers and their designated identity providers control the flow of the data. It is mainly used to manage user identities at the SP side. This is the traditional approach in which service providers want to manage their user

bases and users have no control over their data. We can further classify the Provider Centric Model into two classes: Isolated User Identity Model and Federated Model.

- **Isolated User Identity Model** (a.k.a. Silo Model). This model represents the most common and the simplest IdM model. There are only two parties involved in the scenario: service provider and its clients. Each service provider provides the identifier and the corresponding credential to the clients who wish to receive its services. That means that each service provider has its own identity domain, i.e. has its own IdP and manages its own namespace locally. Though this model represents the simplest form of IdM from the service provider's perspective, it does not allow interoperability, meaning, users from one service provider cannot access service from other service providers as there is no common protocol to establish inter-organisation communication and creates immense burden for the clients as they need to remember the corresponding pair of user-id and password for each single SP and soon those become unmanageable. Currently, all major and leading online service providers such as Google, Yahoo, Amazon, EBay, Facebook, etc. usually follow this model, however, trends are changing that allow Identity Federation (see below).
- **Federated Model**. A federation with respect to the Identity Management is a business model in which a group of two or more trusted partners legally bind themselves with a business and technical contract that allows a user from one partner to seamlessly and securely access restricted resources from other partners. The system that manages Identity Federation is commonly known as the Federated Identity Management (or FIM) System and the model of such identity management is known as the Federated Model. There are different types of Federated Systems currently exist such as Shibboleth [11], Liberty Alliance Architecture [12], etc. Security Assertion Markup Language (SAML, in short) is the most widely used protocol that is used in this setting to allow cross-connectivity among different disparate organisations.

User-centric Model: A User-centric model allows the user to take advantage of the federated model as well as puts users in the centre of identity management so that they can have more control over their data in a privacy-friendly way and allow them to control the flow of their identity data. These emerging models are the latest addition in the Identity Management model and have gained considerable attention in the research community and industry alike due to their long list of advantages and extended flexibilities. Applications based on this model allow the providers to create a federation just like any federated approach. Additionally, users can control their data flow as such applications allow them to decide which personal data to release to which service providers. OpenID [13], Windows CardSpace [14], BrowserID [15], PRIME Architecture developed in the PRIME Project [16], etc. are examples of the User-centric model.

3. Selected Identity Management Systems

For the experiment, demo web-services have been installed locally. The web-services utilise different Identity Management Systems representing different models described previously. Selected Identity Management Systems are the Username-password System representing the Silo Model, the System based on SAML representing the Federated Model, the OpenID and BrowserID Systems representing the User-centric Model. A brief introduction on each system is given below:

3.1 Username-password System

Username-password (denoted as U/P hereafter) represents the SILO model of the Identity Management and is currently the most popular Identity Management System in the web. Each SP maintains its own identity domain and restricts its services using identifiers from that domain. Users of each SP are given the respective identifier and the related credential which the user needs to submit before accessing any services. Usually, such identifiers are incompatible among SPs meaning users from one SP cannot reuse their identifiers in another SP.

3.2 Federated System based on SAML

Security Assertion Markup Language (SAML, in short) from OASIS Security Service Technical Committee is the most mature and dominant Federated Protocol which is used by different FIM Systems [17]. SAML is an XML-based standard for exchanging authentication and authorisation information between different autonomous security domains. It is based on the request/response protocol in which one party (generally SPs) requests for a particular identity information about a user and the other party (usually, IdPs) then responds with the information. The protocol flow is quite simple in nature [17]: the user wants to access a resource protected by a SAML-enabled SP. Assuming, the user is not known to the SP, she will either be redirected to a hard-coded SAML IdP or she will have chance to select her preferred SAML IdP from a list using the so called Where Are You From (WAYF) service and be redirected to the preferred IdP. If the user is not already authenticated at the IdP, she will authenticate there. Upon a successful authentication, she will return to the SP with a security token (known as the SAML Assertion) which needs to be validated. After a successful validation of the token, the user will be allowed to access the service. There are different FIM Systems based on SAML, however we have selected an open source initiative called SimpleSAMLphp [18]. SimpleSAMLphp is an award winning application written in PHP and is based on SAML that can be used to provide federated services.

3.3 OpenID

OpenID is a decentralised Identity Management System which provides Single Sign On (SSO) solution for web services over the Internet [13]. It is a User-Centric technology that aims to assist users by alleviating the need for maintaining and using different accounts to access different web services as well as aid SPs to get rid of their ad-hoc systems for managing their user bases. Currently it has a more than 1 billion user-base and is being used by many big web service providers such as AOL, BBC, Google, IBM, MySpace, Orange, PayPal, Verisign, LiveJournal, Yahoo, etc [13]. The OpenID protocol has three different actors: Users, OpenID Provider and Relying Party. A user is an entity that wants to access a protected service provided by an RP. Assuming the user is not known to the RP, she provides her OpenID identifier (also known as just OpenID). Based on the OpenID, the RP discovers her OpenID provider and redirects the user to the respective provider. The Provider authenticates the user and redirects the user back to the RP with a security assertion. Before the assertion is trusted by the RP, the RP needs to validate it and so it is sent back to the Provider once again. The provider does the validation and the result is sent back to the RP. Upon receiving a positive result, the user can access the service at the RP. As the user is now authenticated at the provider, she can use the corresponding OpenID for accessing any OpenID enabled services without needing any authentication at the provider.

3.2 BrowserID

BrowserID, a Mozilla Lab initiative, is the latest addition into the Identity Management landscape that is aiming to provide a simplified, consistent and decentralised identity management solution for web-enabled services [15, 19]. It's been released in July 2011 and is still evolving heavily. It is based on the Verified Email Protocol (VEP) [20] and has strong support for security and privacy as well as aims to provide a usable one-click login mechanism by reducing the reliance on password as much as possible. The protocol consists of three actors [21]: i) Primary Identity Authority (PIA or simply Primary) - essentially the IdP which provides the user an identifier in the form of an email address. This can be the traditional email providers like Yahoo, Gmail, Hotmail, etc. ii) Relying Parties (RP) - essentially the SP which uses BrowserID to authenticate users and iii) Implementation Provider (IP) – which will provide proxy services for the PIA or browsers in case they do not have native supports for BrowserID. Currently, browserid.org fills in this role. The protocol flow starts with the Certificate Provisioning step in which the user authenticates at the PIA to receive a certificate containing

the user email address and a public key, generated by the user agent. Upon receiving the certificate from the IdP, the browser stores the certificate along with the public-private key pair in its key-ring. The second step, called the Assertion Generation, starts when the user tries to access a service protected by the BrowserID RP. The browser generates an assertion including the certificate received from the PIA, signs it using the private key and then sends it to RP. At this point, the third step, known as Assertion Verification, starts where the RP validates the certificate and retrieves the email address and the public key from the certificate and validates the signature in the assertion. A successful verification of the signature signifies that the user indeed is the owner of the email address and then she is allowed to access the service.

4. Experiment

To measure the power consumption, we conducted our experiment in order to collect quantitative data which will be used as metrics for comparison. Our aim was to simulate real-world usage scenarios that a user would typically perform with her mobile phone and gather some quantitative data during her usages. At the initial stage of the experiment we had the following goals in mind: i) to find out if it differs in power consumption by a browser when the same (similar) service is accessed using different Identity Management Systems and ii) the impact, in terms of power consumption, of choosing a particular browser in the mobile phone to access a service protected by the same Identity Management System. Before we present our findings, it is essential to explain the setup, the data collection methods and other related issues. We discuss each issue in the following sub-sections.

4.1 Setup

To simulate a typical service access scenario using Identity Management Systems, we set up eight different SPs providing eight different yet similar services where each service contained three php pages (index.php, Page1.php and Page2.php). These SPs were identified by eight different URLs: sp1, sp2, saml1, saml2, openid1, openid2, bid1 and bid2 and placed into the document root of an Apache Server which was running on a Windows 7 machine in a local network. Figure 2 illustrates the document structure in the Apache Server. All these services utilised a self-signed certificate to secure connections between the browser and the Apache server. Among these eight services, sp1 and sp2 were protected using username-password and represented the SILO Model.

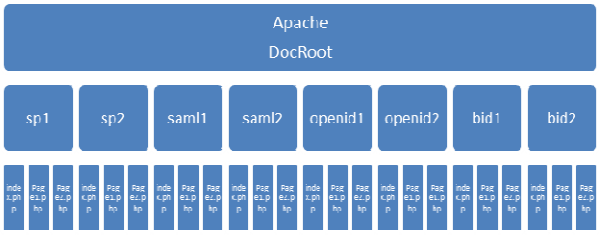


Figure 2: Document Structure in the local Apache Server

saml1 and saml2 represented services that utilised the Federated Model. We used the SimpleSAMLphp library to configure two local SAML SPs for saml1 and saml2. The Feide OpenIdP which is a SAML 2.0 IdP, was used as the SAML IdP [22]. Similarly, openid1 and openid2 services utilised the OpenID System whereas bid1 and bid2 utilised BrowserID System. We used the PHP OpenID library by JanRain, Inc. [23] to configure our services. BrowserID library has been used to configure BrowserID set of services.

A general scenario for accessing these services would be the following: when the user visited one of the first SPs (e.g. saml1), based on the system that SP used, she either would need to provide her username-password (in sp1) or she would be redirected to the respective IdP where

she would need to provide authentication information. Once authenticated, she would not need to provide her authentication information to access the second SPs except in sp2 where she would need to provide her username-password again to access the service.

4.2 Mobile Phones and Browsers

We choose two mobile phones for our study: HTC Hero and Samsung Galaxy SII (SGSII in short, also known as Samsung i9100). The specifications for SGSII are: Dual-core ARM Cortex with 1.2 GHz CPU, Mali-400MP GPU, 1 GB Ram, 4.3 inches Super AMOLED+ capacitive touchscreen with 16M colours and 480 x 800 pixels, at 217 ppi and the OS is Android v2.3 (Gingerbread) [24]. The HTC Hero specs are: ARM 11 with 52b MHz CPU, Adreno 130 GPU, 288 MB of RAM, 3.2 inches TFT capacitive touchscreen with 65k colours and 320 x 480 pixels at 180 ppi and the OS version is Android OS, v1.5 (Cupcake) [25]. However, our phone had an upgraded version of the OS: v2.1. These two phones represent the two generations of Android phones and have been hailed for their builds and performances during the time of their releases. We used the SGSII to represent the high-end spectrum and the Hero to represent the lower end, in terms of hardware as well as software. Choosing two different phones that vary significantly in their specifications would allow us to examine and compare the findings in one phone with respect to another.

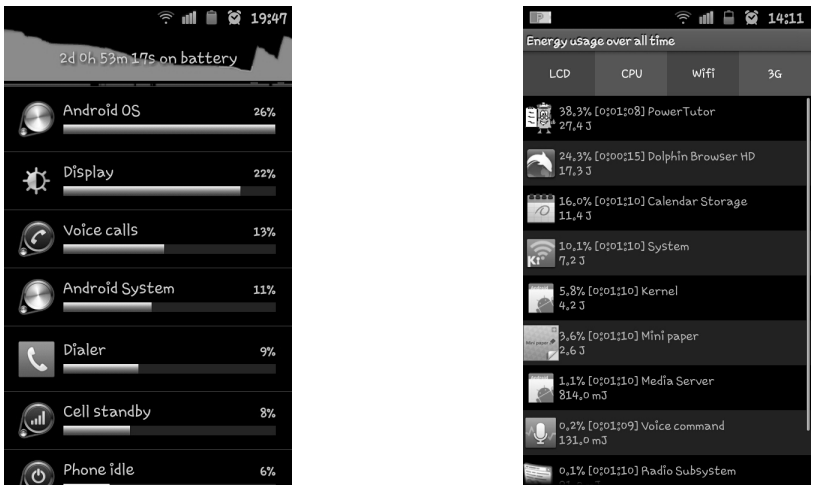
Then, we had to choose browsers for our phones. We selected four browsers: the Android browser (that comes bundled with the OS), Firefox, Opera Mobile and the Dolphin Browser HD. These are the top four browsers downloaded in the Android Platform [26] and that is why they were chosen. We were able to install all of them on the SGSII; unfortunately, the Firefox could not be installed on the Hero as it did not meet the system requirements for Firefox [27].

4.3 PowerTutor App

The heart of this experiment is to compare the battery consumption of different browsers. Unfortunately, Android does not offer any built-in application that can be used to record fine-grained battery consumption at the application level. There is one Battery Usage option under the Setting in Android, that provides the percentage of battery usage of different applications over a long period of time and selecting one of the applications from that list would only reveal the amount of CPU Usage and data sent/received by that particular application (Figure 3a). We needed more detailed statistics on the usage. Our last resort was to look for an app that does the job. There are many apps in the android market that can be used to retrieve statistics regarding power usage. However none seemed to provide the fine-grained stats at the application level that we were looking for except one app called PowerTutor [28]. This app is the result of a joint collaboration between the University of Michigan and Google and is based on a novel online power model called PowerBooter [29]. This model uses built-in battery voltage sensors found in modern smartphones and the knowledge of battery discharge behaviour to record accurate, real-time power consumption estimates of different components of an Android phone such as CPU, LCD, GPS, Audio, Wi-Fi and cellular radio interface.

The PowerTutor app provides an intuitive graphical interface that can be used to read the power consumption estimates by any application for any period of time. Once the app is started, the user needs to click the Start Profiler button so that it can keep running in the background and collect different statistics. Then, the user will have to interact with the respective application and use it like she would normally do. Once she is done using the application, she has to invoke the PowerTutor again. From the main screen of the PowerTutor, the user will need to select the Application Viewer option and this will display a list showing all applications that ran since the Profiler was started. From that list, the user can find the battery consumption estimates (in Joules) of the corresponding application, how long that application has been running as well as power consumption in terms of percentage (Figure 3b). Additionally, the user can individually choose different mobile phone components (such as LCD; CPU, Wi-Fi and Cellular Radio

Interface) from this screen and instruct the PowerTutor to measure the power consumption estimates of only those components that the user has selected. For our experiments, we selected all available components.



(a) (b)

Figure 3: Power consumption provided by the Android (a) and PowerTutor (b)

4.4 Data Collection Methods

Primary data for this experiment have been collected by recording the power consumption estimates provided by the PowerTutor app while going through a series of use-cases using one of the browsers in one of the mobile phones. A general flow of each use-case for an Identity Management model is given below:

i) Assuming we would like to record estimates for the silo model, this use-case will involve visiting different web-pages in two services (sp1 and sp2) that represent the Silo Model. Additionally, the phone will be used in Portrait mode and the login credential (username/password) will be entered each time.
ii) At the beginning of each use-case, the PowerTutor app will be initiated and the Start Profiler button will be clicked to enable the app running in background.
iii) The user will chose one of the browsers.
iv) The user will enter the URL of the first service (https://192.163.0.3/sp1).
v) It will bring the user to the home page (index.php) of sp1. Then the user will visit other two pages (Page1.php and Page2.php) of the service.
vi) The users will enter the URL of the second service (https://192.163.0.3/sp2). Depending on the service, the user may need to login again (U/P) or the user may need to provide her OpenID (OpenID Service), select her BrowserID (BrowserID Service) or do nothing (SAML Service) and the same sequence will take place.
vii) Once all the pages are visited, the PowerTutor app will be invoked again. By clicking the Application View button, the power estimates of that browser will be read and recorded for further analysis.

However, we have found that even though the same service is accessed using the same browser, the power estimates from the PowerTutor tend to vary. This is because the PowerBooter is itself a statistical model and is accurate to within 4.1% of measured values. For this reason, we repeated the flow 5 times for each use-case and took an average. That is, the primary data used for our analysis are average values with a maximum standard deviation of 5.28 Joules.

The same browser will be used in landscape mode to record the power estimates while going through the same flow. At the second stage of data collection for each browser, the Save Password feature will be used to save the username-password at the start of each use-case. Then the power estimates will be recorded for each browser while using a service first in portrait mode and then in landscape mode. We will repeatedly follow this same pattern of use-cases for each service using each browser.

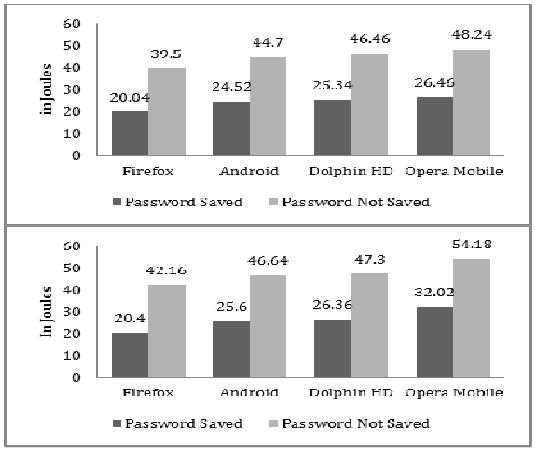
During the data collection period, we found that the user interaction inside each browser varied significantly which affected the way a service could be accessed. For example, sometimes the URLS were copied and pasted into the address bar while other times the URLs could be selected from the address bar. Sometimes the username, OpenID, etc. were saved as Form data and so it was not required to type in the username, other times (especially with Opera), the username could not be saved as Form data and therefore we had to type in the username each time even though the Save Form data option was enabled.

5. Data Presentation & Analysis

This section presents our collected data along with the analysis on them. We have grouped our data according to different Identity Models and within each Model, they have been further grouped according to the mobile phone on which they were collected. Now, we have four different sets of data for every browser in each model using one mobile phone: portrait with password not saved, portrait with password saved, landscape with password not saved and landscape with password saved. The data sets will be used to plot two graphs to represent power consumption in a specific model in which the first graph (denoted by (a)) will represent the comparison between password saved and password not saved states in portrait mode and the second graph (denoted by (b)) will represent the same comparison in landscape mode. These two graphs will enable us to compare, side-by-side, the performance of each browser in a specific model.

5.1 Silo Model (Using U/P):

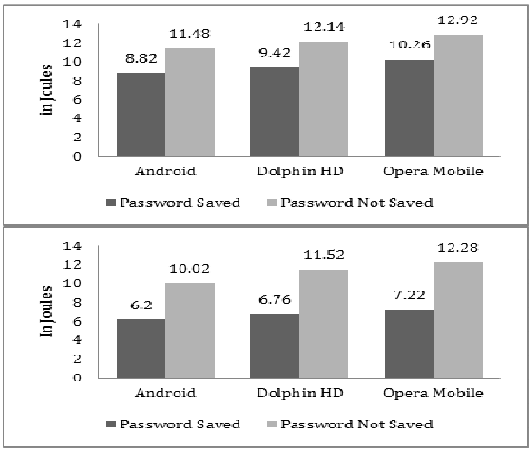
5.1.1 SGSII



(a) In portrait mode b) In landscape mode

Figure 4: Power consumption while using U/P in SGSII

5.1.2 HTC Hero



(a) In portrait mode (b) In landscape mode

Figure 5: Power consumption while using U/P in Hero

5.1.3 Analysis

The first thing to note in these graphs is the difference in power consumption in SGSII and HTC Hero. This difference is due to the fact that both phones differ significantly both in hardware and software. A powerful dual-core processor, more RAM and significantly larger high definition SUPER AMOLED+ screen caused the SGSII to consume considerably more power than its counterpart. This difference makes it meaningless to compare the readings of a specific model in between these phones. However, our goal is to find a pattern of power consumption in one phone and compare it with the other phone. Another thing to note here that all readings presented below have been rounded to whole numbers for clarity.

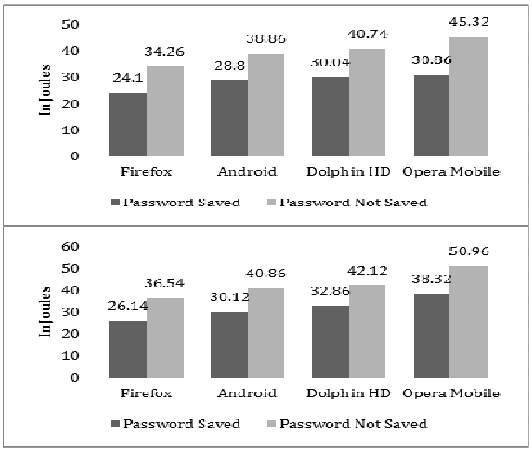
Figure 4 clearly reveals the fact that Firefox was the most economical in power consumption; the Android browser was the second best, then Dolphin and Opera being the least economical in both modes. Opera consumed 32% and 57% more power than Firefox in portrait and landscape mode respectively when the password was not saved and 22% and 29% more when the password was saved. Another revealing fact from these graphs is that all browsers performed well when the password was saved. For example, when the password was not saved, Firefox, the Android browser, Dolphin and Opera consumed 97%, 82%, 83% and 82% respectively more than the scenario when the password was saved in portrait mode and 107%, 82%, 79% and 69% respectively more in landscape mode. The main reason is not very hard to guess. When the password was saved, the user needed minimum interaction time to complete a particular sequence of actions for accessing services. On the other hand, when the password was not saved, the user had to provide the username and password for both services which took a significantly longer time and the browser had to use different hardware components especially screen, processor and the Wi-Fi and software components such as virtual keyboard and touch sensitive user input. All these contributed towards significantly higher consumption readings for all browsers when the password was not saved.

Now, Figure 5 represents the same graph sets for the Hero. The graphs here clearly reveal the same pattern of power consumption as in SGS2 where, Firefox being absent, the Android browser was the most economical, then Dolphin HD and Opera again was the least economical. Opera consumed 16% more power than the Android browser both in portrait and landscape mode when the password was saved and 13% and 23% respectively more when the password was not saved. Like in SGSII, all browsers performed well when the password was saved. For example, when the password was not saved, the Android browser, Dolphin and Opera

consumed 30%, 29%, and 26% respectively more than the scenario when the password was saved in portrait mode and 62%, 70% and 70% respectively more in landscape mode.

5.2 SAML Model:

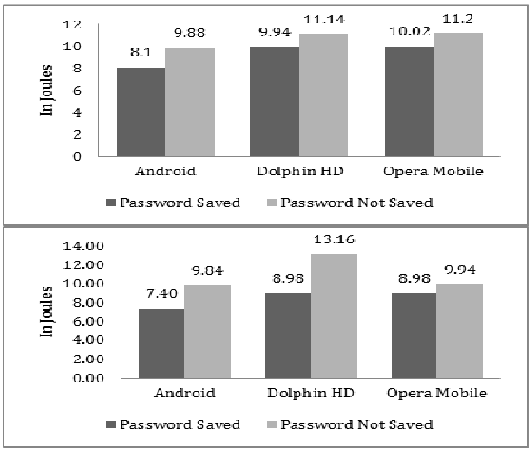
5.2.1 SGSII:



(a) In portrait mode (b) In landscape mode

Figure 6: Power consumption while using SAML in SGSII

HTC Hero:



(a) In portrait mode (b) In landscape mode

Figure 7: Power consumption while using SAML in Hero

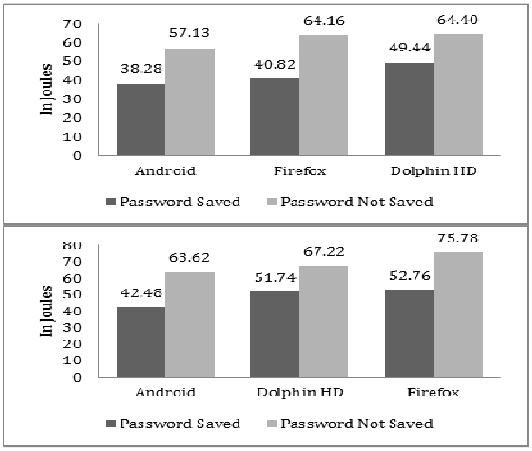
5.2.3 Analysis:

For SAML services, graphs in SGSII (Figure 6) expose the same traits like before where Firefox was the most economical in power consumption, the Android browser being the second best, then Dolphin HD and the Opera was the least economical in both modes. Here, Opera consumed 32% and 39% more than Firefox in portrait and landscape mode respectively when the password was not saved and 28% and 47% more when the password was saved. When the password was not saved, Firefox, the Android browser, Dolphin and Opera consumed 42%, 35%, 36% and 47% respectively more than the scenario when the password was saved in portrait mode and 40%, 36%, 28% and 33% respectively more in landscape mode.

Graphs in HTC Hero (Figure 7) illustrate that the Android browser was the most economical, then Dolphin HD and Opera again was the least economical. Here, Opera consumed 24% and 21% more power than the Android browser in portrait and landscape mode respectively when the password was saved and 13% and 1% respectively when the password was not saved. Like in SGSII, all browsers performed well when the password was saved. When the password was saved, Android, Dolphin and Opera consumed 22%, 12% and 12% respectively more than the scenario when the password was not saved in portrait mode and 33%, 47% and 11% respectively more in landscape mode.

5.3 BrowserID

5.3.1 SGSII:



(a) In portrait mode (b) In landscape mode

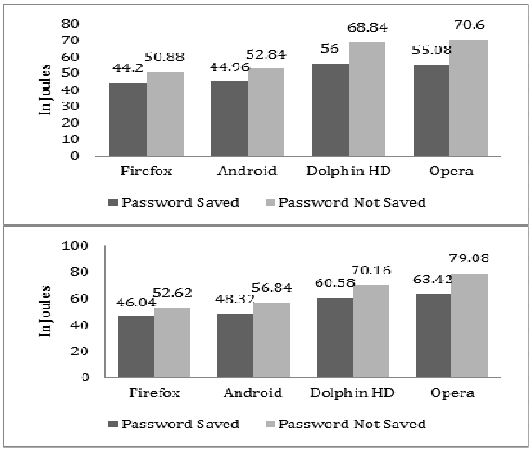
Figure 8: Power consumption while using BrowserID in SGSII

5.3.2 Analysis

We had only one mobile phone (SGSII) and three browsers to consider for BrowserID services as Opera in SGSII and all browsers in HTC Hero could not be used to access services with BrowserID. Graphs in Figure 8 reveal some interesting shift in places in different modes where the Android browser was the most economical in power consumption in both modes. Firefox was the second best in portrait mode meaning that Dolphin was the least economical in portrait mode and the Dolphin being the second best in the landscape mode, Firefox was the least economical. As expected, all browsers performed well when the password was saved. For example, when the password was saved, the Android browser, Firefox and Dolphin consumed 49%, 57% and 30% respectively more than the scenario when the password was not saved in portrait mode and 50%, 44% and 30% respectively more in landscape mode.

5.4 OpenID

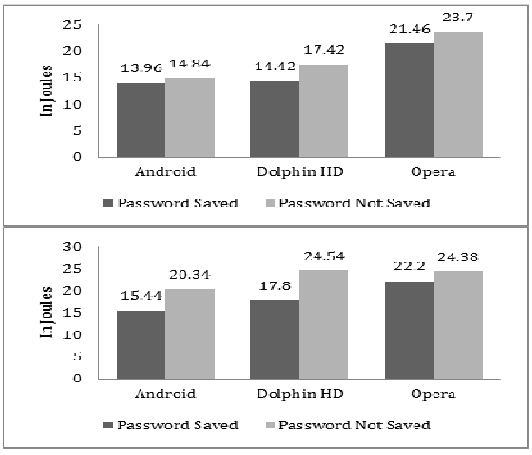
5.4.1 SGSII



(a) In portrait mode (b) In landscape mode

Figure 9: Power consumption while using OpenID in SGSII

5.4.2 HTC Hero



(a) In portrait mode (b) In landscape mode

Figure 10: Power consumption while using OpenID in Hero

5.4.3 Analysis

Graphs in SGSII (Figure 9) expose that Firefox, yet again, was the most economical in power consumption, the Android browser was the second best, Dolphin 3rd (except one case in portrait mode with password saved where Dolphin was the least economical and Opera took the third place) and the Opera was the least economical in both modes. Opera consumed 39% and 50% more than Firefox in portrait and landscape mode respectively when the password was not saved and 25% and 38% when the password was saved. As expected, all browsers performed well when the password was saved. For example, when the password was saved, Firefox, the Android browser, Dolphin and Opera consumed 15%, 18%, 23% and 28% respectively more than the scenario when the password was not saved in portrait mode and 14%, 18%, 16% and 25% respectively more in landscape mode.

The usual pattern of power consumption is evident from the graphs of HTC Hero (Figure 10) where Android was the most economical, then Dolphin and Opera was the least economical. Opera consumed 44% and 54% more power than Firefox both in portrait and landscape mode when the password was saved and 20% and 60% respectively more when the password was not saved. Like in SGSII, all browsers performed well when the password was saved. When the password was saved, the Android browser, Dolphin and Opera consumed 32%, 18%, and 10% respectively more than the scenario when the password was not saved in portrait mode and 6%, 21% and 10% respectively more in landscape mode.

5.5 Comparison of Systems

Graphs in the previous section along with the accompanied analyses could be used to compare the consumption of power by different browsers in a specific model. However, those graphs could not reveal the head-to-head comparison between different models. In this section, we will present some graphs that have been compiled using the same data from the previous graphs to illustrate the head-to-head comparison between different identity systems.

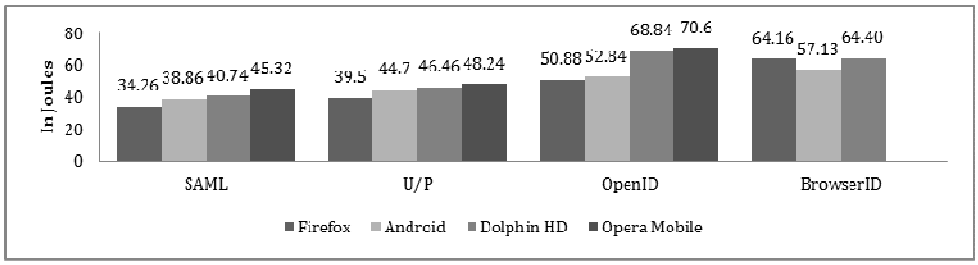


Figure 11: Comparison of different browsers in portrait mode with password not saved – SGSII

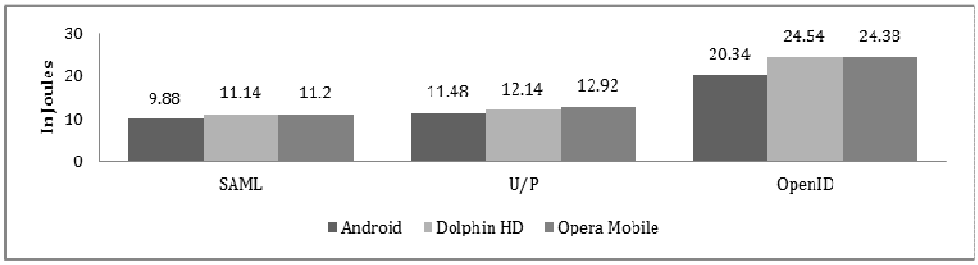


Figure 12: Comparison of different browsers in portrait mode with password not saved – HTC Hero

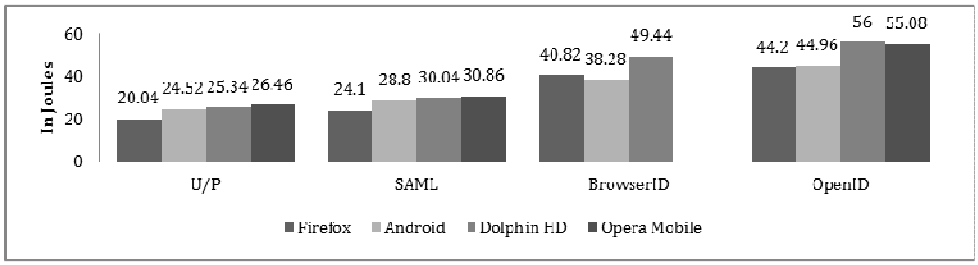


Figure 13: Comparison of different browsers in portrait mode with password saved – SGSII

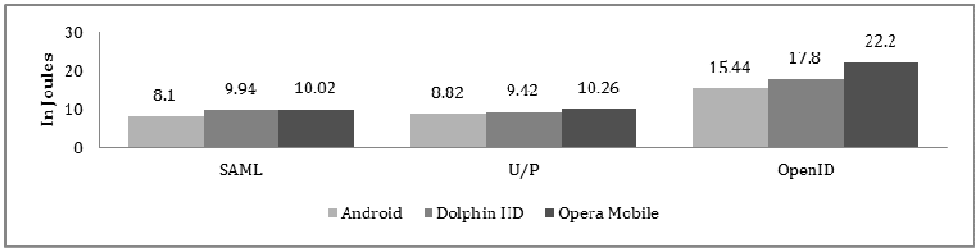


Figure 14: Comparison of different browsers in portrait mode with password saved – HTC Hero

5.5.1 Portrait mode with password not saved

For this setting, the graphs (Figure 11 and 12) clearly indicate that browsers consumed the least amount of power when they were used to access SAML services. Power consumption in U/P services was slightly higher than the SAML services. OpenID took the third position and BrowserID the last. Please note here that, Firefox and the Android browser consumed less power in OpenID than BrowserID, however, Dolphin used more power in OpenID than BrowserID. Two out of three browsers consumed more power in BrowserID and that is why it has been placed into the last place. The interesting observation from these graphs is that U/P consumed slightly more power than the SAML, even though U/P services used the local Apache server for authentication whereas SAML services used the IdP hosted in a different Internet domain. The main reason for this is that the user had to type her username/password twice to access two U/P services whereas she had to type just once to access two SAML services. Typing takes a considerable amount of time in mobile devices which certainly increases the power consumption. The OpenID protocol, on the other hand, has several stages: service discovery, authentication phase, validation phase, etc. Users needed to provide their OpenID as well username/password to access the first service and provide her OpenID only for any other subsequent services. All these contributed towards higher power consumption in OpenID services even though it provided federated services. BrowserID has several phases as well: certificate provisioning, assertion generation, assertion verification, user authentication, etc. Additionally, the protocol is a bit computationally heavy due to different cryptographic operations performed inside the browser. All these contributed towards the higher power consumption while using BrowserID.

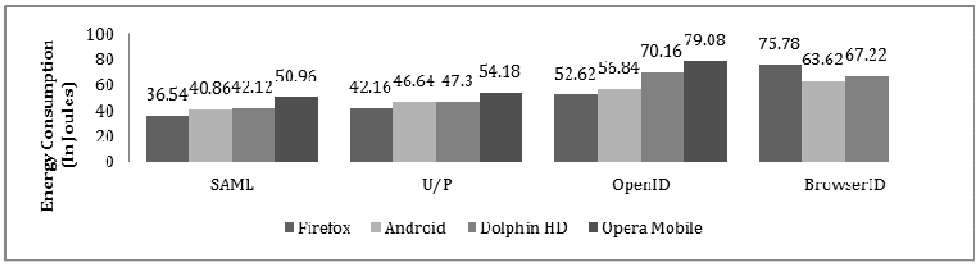


Figure 15: Comparison of different browsers in landscape with password not saved – SGSII

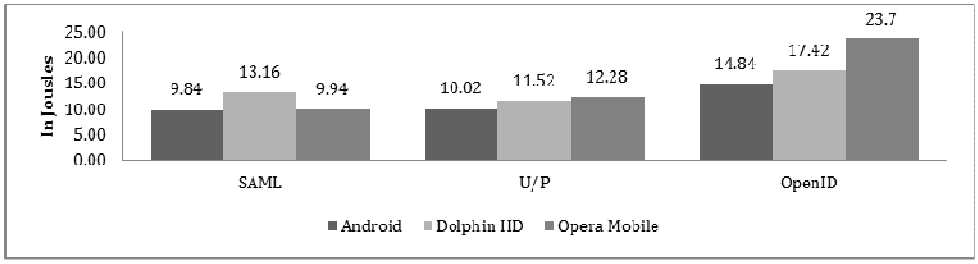


Figure 16: Comparison of different browsers in landscape with password not saved – HTC Hero

5.5.2 Portrait mode with password saved:

When the same services were accessed in portrait mode and the password was saved, we find that the position in SGSII is swapped between SAML and U/P services with U/P services being the most economical and between BrowserID and OpenID with OpenID being the least economical (Figure 13 and 14). We can identify at least one reason for this exchange of place between SAML and U/P services. As the password was saved and the user did not need to type in the username/password, the transaction time in U/P services was slightly lower than that of SAML services. However, SAML, like before, was the most economical, OpenID the least economical and the U/P was in between them in the Hero.

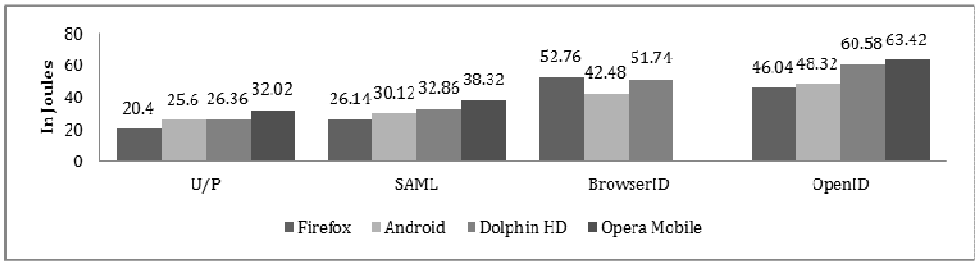


Figure 17: Comparison of different browsers in landscape with password saved – SGSII

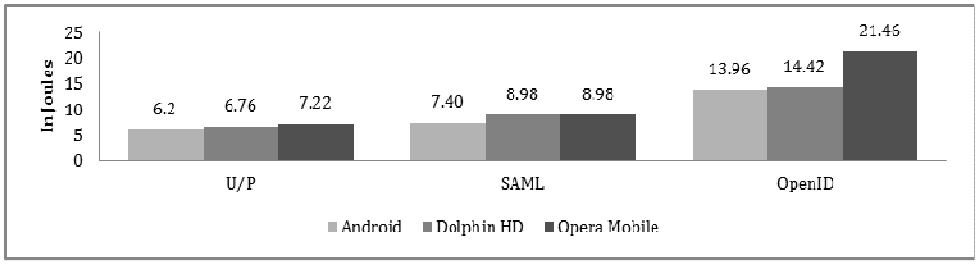


Figure 18: Comparison of different browsers in landscape with password saved – HTC Hero

5.5.3 Landscape mode with password not saved:

In this setting, the user accessed different services in landscape mode and the password was not saved (Figure 15 and 16). Once again, SAML was the most economical and BrowserID was the least economical while U/P and OpenID services held the second and third position in SGSII. The same traits followed in HTC Hero where SAML was the most economical and OpenID the least economical with U/P in-between.

5.5.4 Landscape mode with password saved:

When the services were accessed in landscape mode and the password was not saved, U/P services performed better than the SAML services in both mobile phones, BrowserID took the third place in SGSII and OpenID consumed the highest power in both mobile phones (Figure 17 and 18).

6. Discussions

From the data and graphs presented in previous sections, it is clear that neither a browser nor a service showed a fully consistent behaviour. One browser was the best performer in one mode (and scenario) while another browser took the crown in another mode (and scenario). The same thing was evident in IdM Services as well where one service performed better in one case while another service performed better in other cases. Despite this huge variation in power consumption by the browsers and in the service, a few patterns emerge. Table 2, 3, 4 and 5 can identify these patterns.

Table 1: Relative ranking of browsers in SGSII

	U/P				SAML				BrowserID				OpenID			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Portrait	F	A	D	O	F	A	D	O	A	F	D	X	F	A	D	O
Landscape	F	A	D	O	F	A	D	O	A	D	F	X	F	A	D	O

Table 2: Relative ranking of browsers in HTC Hero

	U/P			SAML			OpenID		
	1	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
Portrait	A	D	O	A	D	O	A	D	O
Landscape	A	D	O	A	D	O	A	D	O

In table 1 and 2, F represents Firefox, A represents the Android browser, D represents Dolphin HD and O represents Opera. X is used to represent the situation where there was no data available for that particular scenario. Similarly, in table 4 and 5, S represents SAML, Op is for OpenID and B for BrowserID. The respective numbers (1, 2, 3 and 4) represents the order in power consumption where 1 represents the most economical (i.e. consumed least power) and 4 represents the least economical (i.e. consumed most power).

Table 1 represents a relative ranking of browsers in both modes according to their power consumption in different systems in SGSII. Browsers exhibited a consistent power consumption patterns in U/P, SAML and OpenID where Firefox has been the most economical browser (hence taking the number 1 position), the Android browser was the second best, then Dolphin and Opera was the least economical in both modes (Portrait and Landscape). In BrowserID, however, the patterns are not consistent and tend to change in different modes. For example, Android is the most economical in both modes. Firefox and Dolphin take the 2nd and 3rd position respectively in Portrait mode and the position is reversed in landscape mode where Dolphin takes the 2nd position and Firefox the 3rd.

Table 2 represents a relative ranking of browsers in both modes according to their power consumption in different systems in HTC Hero. The patterns here are more stable where the Android browser was the most economical, Dolphin took the 2nd place and Opera was the least economical throughout.

Table 3: Relative ranking of Identity Systems in SGSII

	1	**2**	**3**	**4**
Portrait with password not saved	S	U/P	Op	B
Portrait with password saved	U/P	S	B	Op
Landscape with password not saved	S	U/P	Op	B
Landscape with password saved	U/P	S	B	Op

Table 4: Relative ranking for Identity Systems in HTC Hero

	1	**2**	**3**
Portrait with password not saved	S	U/P	Op
Portrait with password saved	S	U/P	Op
Landscape with password not saved	S	U/P	Op
Landscape with password saved	U/P	S	Op

Table 3 represents the overall ranking of Identity Systems in four different settings when used in SGSII. The table clearly illustrates a pattern. When the password is not saved, the SAML service is the most economical; U/P takes the 2nd place, OpenID takes the 3rd position whereas the BrowserID services are the least economical in both modes. The positions are pair-wise reciprocal in case the password was saved in both modes where the U/P service is the most

economical, SAML takes the 2nd place, BrowserID takes the 3rd place and OpenID is the least economical.

Table 4 represents the overall ranking of Identity Systems in four different settings when used in HTC Hero. In three settings out of four, the SAML service is the most economical and the OpenID service is the least whereas U/P services take the 2nd place. However, when HTC Hero is used in landscape mode with password saved, U/P becomes the most economical, SAML takes the second place and OpenID takes the last position.

The tables above have identified different patterns in different settings as power consumption of the browsers tends to fluctuate a lot. A few reasons behind these fluctuations could be identified which are mostly related to different usability issues:

- Different browsers, when used in the landscape mode, have different ways to display the keyboard. For example, in Opera, there is no Next button in the landscape keyboard like in Firefox. Clicking in the next button allows users to shift focus easily from one field to another (e.g. moving focus from the username field to the password field). Due to the absence of the Next button, the user, after typing in the username, requires to click the *Done* button at first which would slide down the keyboard, then select the password field once again to type in the password. All these prolong the interaction time which in turn increases the power consumption.
- Opera never saved any form data and it was required to type in or copy the username/email/OpenID into the clipboard.
- Browsers have different ways of saving and displaying the history when a URL is typed in the address bar or the username is typed into the username field.

Other than these, power consumption mainly depends on the internal mechanisms by which a browser utilises the Operating System and hence it is very difficult to track down the exact reason that causes the power consumption to vary in between different browsers.

7. Related Works

There are a few papers that have developed different power models for smartphones and in doing so measured power consumption of different hardware and software components including browsers [29, 30, 31, 32]. However, they did not provide any comparative analysis of power consumption among different available browsers in smartphones. Using a power model to measure, analyse and compare the power consumption in different browsers to access services provided by different Identity Management Systems is a novel idea and to the best our knowledge has not been explored before.

8. Conclusions

In this paper we have presented our findings and analysis on the power consumption of different browsers when they were used to access different services protected by different Identity Management Systems. The data, graphs and table clearly show that certain browsers are likely to consume more power than other browsers when used in the same settings and to access the same service. On the other hand, the same services provided by different Identity Management Systems are also likely to have their own impact on the power consumption of a browser. We have also found that it was a bit difficult to collect data due to the volatility of readings in the PowerTutor app. The app itself consumed a huge amount of power which was necessary to estimate the power consumption by all consecutively running applications. We think that if such a facility could be packaged into the OS itself as a built-in feature, it might provide more stable and realistic data than the current available applications. There are a couple of research directions from this point: the same method could be used to record the power consumption in various other smartphones and then correlate those readings with the current data set. It was found in [33] that using a particular desktop proxy server in a SAML setting, they had a

performance boost (in terms of average response) of 39.53%. The proxy server was used to outsource some of the tasks typically performed in the mobile phones. The similar approach could be used to check if a considerable gain in power consumption could be achieved in different identity systems, especially in computationally heavy systems such as OpenID and BrowserID.

REFERENCES

[1] Cameron, K. May 2005. The Laws of Identity. [Online]. Available: http://www.identityblog.com/stories/2004/12/09/thelaws.html.

[2] Global mobile statistics 2011. November 2011. [Online]: Available: http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats.

[3] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010–2015. February 2011. [Online]: Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ ns827/white_paper_c11-520862.html

[4] Thanh, D. V. and Jørstad, I. 2008. The ambiguity of identity. Telektronikk Telenor, 3/4.2007.

[5] Dixon, M. December 2005. Identity Map - Physical Identity. [Online]. Available: http://blogs.oracle.com/identity/entry/identity_map_physical_identity

[6] Wikipedia entry on Entity. Accessed on 14th June, 2011. [Online]. Available: http://en.wikipedia.org/wiki/Entity.

[7] Joosten, R., Whitehouse, D. and Duquenoy, P. 2008. Towards a Meta Model for Identity Terminology. In IFIP/FIDIS Summerschool, 2008. [Online]. Available: http://www.buslab.org/SummerSchool2008/slides/ Rieks_Joosten.pdf.

[8] Modinis Common Terminological Framework for Interoperable Electronic Identity Management. Accessed on 28th June, 2011. [Online]. Available: https://www.cosic.esat.kuleuven.be/modinis-idm/twiki/bin/view.cgi/Main/GlossaryDoc.

[9] Jøsang, A., Al, M. and Suriadi, Z. S. 2007. Usability and privacy in identity management architectures. In ACSW '07: Proceedings of the fifth Australasian symposium on ACSW frontiers', Australian Computer Society, Inc, pp. 143–152.

[10] Jøsang, A. and Pope, S. 2005. User Centric Identity Management. In Asia Pacific Information Technology Security Conference, AusCERT2005, Australia, pp. 77–89.

[11] Shibboleth Project. September, 2005. Shibboleth Architecture: Protocols and Profiles. [Online]. Available: http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-latest.pdf.

[12] LIBERTY ALLIANCE PROJECT, Liberty ID-FF Architecture Overview Version: 1.2-erratav1.0. [Online]. Available: http://projectliberty.org/liberty/content/download/318/2366/file/draft-liberty-idff-arch-overview-1.2-errata-v1.0.pdf.

[13] OpenID. [Online]. Available: http://openid.net/.

[14] Microsoft Windows CardSpace. [Online]. Available: http://www.microsoft.com/windows/products/winfamily/cardspace/default.mspx.

[15] BrowserID. [Online]. Available: https://browserid.org/.

[16] PRIME Project. [Online]. Available: https://www.prime-project.eu/.

[17] Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. 15 March 2005. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[18] SimpleSAMLphp. [Online]. Available: http://simplesamlphp.org/.

[19] Introducing browserid: A better way to sign in. July 2011. [Online]. Available: http://identity.mozilla.com /page/2.

[20] Verified email protocol. Accessed on 21 November 2011. [Online]. Available: https://wiki.mozilla.org/Labs/Identity/VerifiedEmailProtocol.

[21] Hilaiel, L. July 2011. How browserid works. [Online]. Available: http://lloyd.io/how-browserid-works.

[22] The Feide OpenIdp. [Online]. Available: https://openidp.feide.no/.

[23] PHP OpenID Library by JanRain, Inc. [Online]. Available: http://www.janrain.com/openid-enabled.

[24] Samsung Galaxy SII – Full Phone Specifications. [Online]. Available: http://www.gsmarena.com/samsung_i9100_galaxy_s_ii-3621.php.

[25] HTC Hero – Full Phone Specifications. [Online]. Available: http://www.gsmarena.com/htc_hero-2861.php.

[26] Android Market. [Online]. Available: https://market.android.com/.

[27] System requirements for Firefox Mobile. [Online]. Available: https://wiki.mozilla.org/Mobile/Platforms/Android#System_Requirements.

[28] PowerTutor App. [Online]. Available: https://market.android.com/details?id=edu.umich.PowerTutor &hl=en.

[29] Zhang, L., Tiwana, B., Qian, Z., Wang, Z., Dick, R. P., Mao, Z. M. and Yang, L. 2010. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis* (CODES/ISSS '10). ACM, New York, NY, USA, 105-114. DOI=10.1145/1878961.1878982 http://doi.acm.org/10.1145/1878961.1878982.

[30] Carroll, A. and Heiser, G. 2010. An analysis of power consumption in a smartphone. In Proceedings of the 2010 USENIX Annual Technical Conference, June 2010.

[31] Dong, M. and Zhong, L. 2010. Sesame: A self-constructive virtual power meter for battery-powered mobile systems. Tech. Rep.

[32] Gurun, S. and Krintz, C. 2006. A run-time, feedback-based energy estimation model for embedded devices. In Proc. Int. Conf. Hardware/Software Codesign and System Synthesis, Oct. 2006, pp. 28-33.

[33] Tran, T. and Wietfeld, C. 2009. Approaches for optimizing the performance of a mobile SAML-based emergency response system. Enterprise Distributed Object Computing Conference Workshops, 2009. EDOCW 2009. 13th , vol., no., pp.148-156, 1-4 Sept. 2009. doi: 10.1109/EDOCW.2009.53320.

Authors

Md. Sadek Ferdous

Short Biography:

Md. Sadek Ferdous is a second year PhD student at the School of Computing, University of Glasgow, UK where he is investigating the ways a mobile phone can be used for managing user identities. Before that he worked as a research assistant for a year at the University of Kent, UK participating in a JISC funded project (Logins4Life) where he investigated on the integration of Social Network sites into the Federated Identity Management System to allow users to use their Social Networking accounts to access restricted resources in the Academia. He holds double masters in Security and Mobile Computing from the Norwegian University of Science & Technology (NTNU), Norway and the University of Tatu in Estonia. His research interests include Identity Management, Privacy Enhancing Technologies, Trust Management, Security Usability and Petname Systems. He is the author of several publications on different aspects of Networking, Security and Identity.

Dr. Ron Poet:

Short Biography:

Dr. Ron Poet is a Lecturer in the School of Computing Science at the University of Glasgow, Scotland. He holds an MA and a PhD. in Mathematics from the University of Cambridge, UK. Before joining the University of Glasgow, he served in the University of Cardiff, UK. His current research interests include Security and Authentication, Bioinformatics, Object Oriented Technology, Dynamical Systems & Chaos Theory and Atomic Astrophysics. He is the author of several publications on Security & Authentication, Bioinformatics, Object Oriented Technology, Chaos Theory and Atomic Astrophysics. He is the co-author of a book on Chaos Theory which is to be published this year. Apart from teaching, he holds several administrative positions in the respective department.