DEGREE PROJECT IN COMMUNICATION SYSTEMS, SECOND LEVEL
STOCKHOLM, SWEDEN 2016

Comparison of Methods of Single Sign-On

Post authentication methods in single sign on

BARAN TOPAL

KTH ROYAL INSTITUTE OF TECHNOLOGY
INFORMATION AND COMMUNICATION TECHNOLOGY

Comparison of Methods of Single Sign-On

Post authentication methods in single sign on

Baran Topal

2016-03-02

Master's Thesis

Examiner and Academic adviser
Gerald Q. Maguire Jr.

Industrial adviser
Kent Hammarström

KTH Royal Institute of Technology
School of Information and Communication Technology (ICT)
Department of Communication Systems
SE-100 44 Stockholm, Sweden

Abstract

Single sign-on (SSO) is a session verification mechanism that allows a client to use a single password and name combination to be able to access multiple applications. The mechanism validates the client for all the applications and eliminates the need for authentication prompts when a user switches between applications within a session.

SSO mechanisms can be classified as software versus hardware or customer-requirements oriented versus server-side arrangements. The five commonly used mechanisms of Single Sign-On currently are: Web Single Sign-On, Enterprise Single Sign-On, Kerberos (or Ticket/Token Authentication), Open ID, and Federation or Federated Identity.

SSO has the main benefit of allowing a user to access many different systems without having to log on to each and every one of them separately. However, SSO introduces a security risk as once an attacker gains access to a single system, then the attacker has access to all of the systems.

This thesis describes SSO technology, the Security Assertion Markup Language, and the advantages and risks involved in using SSO. It examines authentication mechanisms and their suitability for SSO integration. The main emphasis is a description of a mechanism that ameliorates some of the disadvantages of SSO by monitoring the user behavior with respect to a template. If a user performs actions that fit the defined template behavior, then the post authentication mechanism will not get activated. If, on the other hand, a user does something unforeseen, the mechanism will not perform authentication for this user, but rather trigger manual authentication. If this manual authentication succeeds, then the user will continue to interact with the system, otherwise user session will be ended.

This behavior extension authentication mechanism is a method that eases the authentication process in which users are not expected to remember any username and password that can be forgotten easily or have a biometric attribute that can change over time. This method can be integrated to existing web application without a major risk and increase in cost.

Keywords

SSO, Single sign-on, SAML, authentication, security, behavior, risk

Sammanfattning

Single sign-on (SSO) är en sessionkontrollmekanism som gör det möjligt för en kund att använda en ett enda par av lösenord och namn för att kunna få tillgång till flera olika program. Mekanismen validerar klienten för alla anrop och eliminerar behovet av ytterligare inloggningsdialoger när en användare växlar mellan program inom en session.

SSO-mekanismer kan klassificeras enligt olika kriterier, såsom programvara kontra hårdvara eller kunder krav orienterade mot serversidan arrangemang. De fem vanligen använda mekanismerna för Single Sign-On är närvarande: Web Single Sign-On Enterprise Single Sign-On, Kerberos (eller Token autentisering), Open ID och Federation eller Federated Identity.

SSO har den stora fördelen att en användare kan få tillgång till många olika system utan att behöva logga in på vart och ett av dem separat. Men SSO inför också en säkerhetsrisk i och med att tillgång till ett enda av systemen också automatiskt innebär tillgång till samtliga.

Denna avhandling beskriver SSO-teknik, Security Assertion Markup Language, och fördelarna och riskerna med att använda SSO, samt undersöker autentiseringsmekanismer och deras lämplighet för SSO integration. Tyngdpunkten är en beskrivning av en mekanism som minskar några av nackdelarna med SSO genom att övervaka användarnas beteende med avseende på en mall. Om en användare utför åtgärder som passar det beteende som beskrivs av mallen, då den föreslagna mekanismen kommer att hantera autentiseringen automatiskt. Om, å andra sidan, en användare gör något oförutsett, kommer mekanismen inte att automatiskt utföra autentisering för den här användaren, utan utlöser manuellt autentisering. Om denna manuella autentiseringen lyckas, så kan användare fortsätta att fortsätta att interagera med systemet, annars kommer användarsessionen att avslutas.

Denna beteendebaserade utvidgning av autentiseringsmekanismen är en lovande metod som minskar behovet av att komma ihåg många namn och lösenord, utan att lämna delsystem öppna till de säkerhetsproblem som uppstår i ren SSO, och utan att vara beroende av biometriska egenskaper som kan förändras över tiden. Denna metod kan integreras med befintliga webbaserade lösningar utan ökad risk och ökade kostnader.

Nyckelord

SSO, Single sign-on, SAML, autentisering, säkerhet, beteende, risk

Acknowledgments

I would like to thank Kent Saxin Hammarström for his efforts to help me. I would like to thank also Tacton Systems that enables me to have the interviews and surveys that I needed to conduct in company premises. I am also grateful that my family and my sambo, Malin Altamirano Björn who were supporting me to finish this thesis and Minyon, our cat.

Stockholm, February 2016
Baran Topal

Table of contents

List of Figures

List of Tables

List of acronyms and abbreviations

AAA	Authentication, Authorization, and Accounting
ACS	Annual Cost of the Safeguard
ALE	Annualized Loss Expectancy
ARE	Asymptotic Relative Efficiency
ARO	Annual Rate of Occurrence
AD	Active Directory
AJAX	Asynchronous JavaScript and XML
CHAP	Challenge Handshake Authentication Protocol
CIA	Confidentiality, Integrity, and Availability
COBIT	Control Objectives for Information Technology
CRA	Challenge-Response Authentication
CRC	Cyclic Redundancy Check
Df	Degrees of Freedom
DMZ	Demilitarized Zone
DNS	Domain Name System
EF	Exposure Factor
FAR	False Accept Rate
FRAP	Facilitated Risk Analysis
FRR	False Reject Rate
FTP	File Transfer Protocol
HTML	Hyper-Text Markup Language
HTTP	Hyper-Text Transfer Protocol
ID	Identifier
IdP	Identity Provider
IEC	International Electrotechnical Commission
IIS	Internet Information Service
IS	International Organization for Standardization
ISMS	Information Security Management System
ISP	Internet Service Provider
IP	Internet Protocol
IT	Information Technology
JSON	JavaScript Object Notation
JQuery	JavaScript Query
NIST	National Institute of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
OCTAVE	Operationally Critical Threat, Asset, and Vulnerability Evaluation
PAP	Password Authentication Protocol
PC	Personal Computer
PKI	Public Key Infrastructure
PPP	Point-to-point
RADIUS	Remote Authentication Dial-in Service
RP	Relying Party
SAML	Security Assertion Markup Language
SASL	Simple Authentication and Security Layer
SCRAM	Salted Challenge Response Authentication Mechanism
SD	Standard Deviation
SHA	Secure Hash Algorithm
SLE	Single Loss Expectancy

SOAP	Simple Object Access Protocol
SP	Service Provider
SPAP	Shiva Password Authentication Protocol
SSL	Secure Socket Layer
SSO	Single sign on
TACACS+	Terminal Access Controller Access Control System
TLS	Transport Layer Security
UI	User Interface
URI	Uniform resource identifier
VM	Virtual Machine
WAMS	Web Access Management Systems
WAYF	Where Are You From
WWW	World Wide Web
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
XSS	Cross-Site Scripting

1 Introduction

Today users face the issue of frequently juggling passwords and username combinations. When working with disparate platforms and applications, they are prompted to log in to each platform and application. Single Sign-On (SSO) technology is a method of alleviating this problem. SSO is continuously evolving to address the issues faced by information technology workers and their applications. This thesis presents some of the SSO techniques in use today and examines the potential of post-authentication mechanisms that can exploit knowledge of behavioral attributes of the logged in user in order to reduce the burden on the user of explicitly having to login again and again.

SSO is not a single concept, but rather is an umbrella term used for all those techniques and devices or combinations that aim to lessen the burden of multiple login instances within a session. It is an arrangement that helps end users of systems to access their work seamlessly with the added benefit for enterprises of security and consistency. Securing an array of activities ranging from Internet browsing and navigation through various platforms and legacy applications is a challenging task. Each of these activities has its own set of issues, requirements, and security parameters that may vary greatly from one to the other.

However, SSO has an obvious drawback: Unauthorized access to just a single SSO user-password combination could be disastrous for the entire organization and the ramifications could be far-reaching. This inherent security predicament can be circumvented by designing the SSO mechanism to make it difficult for unauthorized individuals to exploit their access to part of the organization's domain in order to gain access to another part of the organization's domain. This thesis presents one such design and then analyzes it. However, there are various potential designs and each organization needs to look at its own requirements closely and then evaluate relevant designs. There are many factors to consider in this evaluation, such as design, deployment, and usage.

1.1 Background

SSO simplifies the complexities of a heterogeneous security architecture. A system that utilizes SSO should handle resource management, information services, and data management. Moreover, security must be provided end-to-end within the security architecture. Security Associations (SAs) utilize different techniques and approaches to address this requirement. Each SA defines the protocols, keys, and encryption algorithms that are to be utilized to provide this end-to-end security. If there is a disruption in the establishment of an SA, this causes a disruption in the operation of this part of the organization's system. Disruptions can be caused by key and protocol mismatch. The performance of the SA will depend upon the encryption algorithm and hashing algorithm that are utilized. Some algorithms, such as public key encryption algorithms, are computationally expensive due to large cryptographic key sizes, the need for two cryptographic keys instead of one, and with the introduction of a certificate authority extra domain name system look-ups and validation of the certificates[*]. For all of these reasons, utilization of a Public Key Infrastructure (PKI) increases security at the cost of increased server response times [2].

[*] Additionally, these domain name lookups should be using Domain Name System Security Extensions (DNSSEC) [1].

Different types of SSOs can be utilized, such as web-based and non-web based (legacy) SSO. Web based SSO introduces a web based architecture in which clients may need to login to different web systems, but they only need to do so once. Such an SSO can be Internet or intranet facing, but the fundamental goal is that each user logins in only once (per session[*]).

SSOs use tokens/cookies and Security Assertion Markup Language (SAML) deployments to provide security [3]. A validation token is transmitted over a secure channel, customarily, Secure Socket Layer (SSL). SSL makes use of server certificates. The SSL cryptographic functions are provided to clients via browsers. These tokens are sent through secure channels to other security systems in order to provide verification of the user's identity [3]. Once authentication is successful, such tokens are removed and identification signatures are transmitted to the system that initiated the transaction to enable the client system to access data or other resources [3].

1.2 Problem definition

The advantages of SSO are also its disadvantages. The advantage of SSO is that it replaces the individual authentication mechanisms of a group of systems with a single authentication mechanism. This is also the disadvantage, since after the initial authentication there is no subsequent check of whether the previously authenticated user is actually who he/she claims to be or if it is still this same user. This risks providing a user access to an entire system with many subsystems, when this user should **not** have been authenticated in the first place or who should only have been able to access a subset of these systems.

1.3 Purpose

The purpose of this thesis project is to evaluate a tradeoff between a SSO with multiple independent points of failure versus a SSO with a single point of failure. The proposed solution will affect the following aspects of SSO:

- End users

  End users avoid the hassle of multiple login procedures. (This thesis will describe how an organization can utilize SSO so that the user's login procedure will be seamless.)

- IT Security personnel
  - The proposed solution will ease IT security personnel's duties by reducing security concerns after an end user has authenticated via SSO.
  - The security concerns introduced with vanilla SSO are reduced, while reducing the number of problem tickets which the security personnel must handle.

- Corporations
  - Ease problems of multiple logins, while reducing costs
  - Providing a stronger authentication mechanism
  - A cheaper solution that the alternative of multiple authentications

[*] This session can be limited to a fixed maximum period of time or to the period of time that the browser maintains a connection with the server.

1.4 Goals

The aim of this thesis is to outline the technical details of the deployment of the proposed SSO. A number of different post authentication mechanisms that improve the confidentiality when SSO is employed will be assessed in terms of their advantages and disadvantages. This thesis investigates how behavioral patterns of an already authenticated user could be exploited by SSO.

The objective of this thesis project is to analyze commercially available SSO applications. However, this thesis project will *not* consider an exhaustive list of SSO solutions. The purpose is rather to examine the potential of post authentication methods. The popular uses of cryptography in security measures are also investigated. However, hands-on work on each of these security measures will *not* be done.

This thesis:

- Describes SSO implementations and solutions, along with their advantages and disadvantages,
- Presents a practical implementation using Shibboleth (described in Section 2.3.6),
- Describes known authentication methods and their shortcomings,
- Presents a rationale for introducing post-authentication methods,
- Introduces the idea of using Real Time User Behavior Attribution of a user already authenticated by SSO,
- Proposes a failover authentication mechanism for false positives, and
- Implements and evaluates the proposed solution using behavioral patterns and a failover authentication mechanism in SSO.

1.5 Research Methodology

In this research, I adopt design science research methodology as it is well suited for a research project aiming to develop an artifact [4]. This research methodology follows certain steps in order to make scientific contributions to information sciences. In this thesis, I critically analyze a particular SSO system and formulate my hypothesis (that standard SSO is as good as or better than my proposed solution). This thesis concerns “artifact development”, which means that the end goal of this thesis is to propose, *implement,* and *evaluate* a solution, which falsifies my hypothesis.

1.6 Delimitations

The delimitations of this project are:

- Comparing all SSO solutions is out of scope, due to insufficient resources for setting up all of these SSO environments.
- The implementation described in this thesis illustrates only one SSO solution (specifically Web SSO – see Section 2.3.5) and one authentication mechanism for failover (Challenge Handshake Authentication Protocol (CHAP) see Section 2.7).
- The literature study is limited to public information and sources available via KTH’s library.
- This thesis does not cover PKI infrastructure, the interested reader is referred to [5–7].

1.7 Structure of the thesis

Chapter 2 presents relevant background information about SSO. Chapter 3 presents the methodology and method used to solve the problem. Chapter 4 presents the analysis and major results. The last chapter presents the conclusion and future work.

2 Background

Authentication, authorization, and accounting (AAA) are three vital functions sought by web service providers [8]. Single sign-on offers all these functions and gives users the ability to access more than one protected resource with a single authentication schema.

SSO enables a system manager to restrict websites and applications by defining what resources they want to protect and defining rules for access to each resource. These rules are policies and together they implemented the following three functions:

Authentication	Authentication determines whether a user is who he or she claims to be. In SSO, a user is authenticated based upon the user's response to a challenge which is generated when the user initiates a session and requests access to a resource.
Authorization	Authorization determines if a given user has the right to access the requested resource.
Accounting	Accounting provides information that can subsequently be used to audit access to a set of assets. Accounting records: "What was done?". "Who performed the action?", and "Where/When did an access/operation occur?"

By incorporating the above functions, SSO technology empowers clients and domains to safely access different services using a single key.

Currently, nearly all services provided over the Internet via the Internet Protocol (IP) validate user identity through username-password combinations [9]. Internet Service Providers (ISPs) administer this via systems, such as Remote Authentication Dial-In Service (RADIUS) and Terminal Access Controller Access Control System (TACACS+) that are designed to authorize access as well indicate which services a client or user is allowed to access. The client or user generally needs different passwords to access different services. However, in practice this leads to users using a single password, because this is easiest for them to remember. Moreover, in TACACS+, the username and password are sent in plaintext [10p. 16], whereas in RADIUS the password is hashed, hence computing a plaintext password corresponding to a hashed password requires cryptanalysis [11p. 70] Utilizing a username and password makes a weak system from a security point of view [12]. As more and more value based services are offered on the Internet, it has become important to overcome this weakness. PKI solutions are one mechanism to reduce this weakness, but this thesis does not go into details of PKI.

2.1 Single Sign-On (SSO)

Among authentication frameworks, SSO is one of the most desired frameworks [13]. SSO is the process by which the user authenticates only once to gain access to various services *without* any further need to verify his/her credentials [14]. Following deployment of SSO, the user's working procedures become much simpler, as recurring requests for credentials are removed. Therefore, a client requires only one logon and there is only one username-password pair that each client has to deal with. This improves the efficiency of service consumption. An indication of this is that 46% of Facebook users use Facebook's SSO to login to other services [15].

Given a SSO infrastructure, an organization gains the following advantages [16]:

- Security administrators and support staff have a smaller database of user credentials to deal with, making it easier to track the access of individual users to services and resources.
- All of the authentication data is under central control and the tools, techniques, and routines of controlling and manipulating this data are identical, thus simplifying the task of the security administrators.

However, these advantages can also be vulnerability, as gaining access to one of the systems guarded by SSO enables the intruder to access to all of the other systems guarded by this SSO [17].

SSO is advantageous because of its simplicity of organization and utilization. Additionally, SSO is advantageous from a security point of view as centralization facilitates uniformity in applying security policies across the organization. A decentralized structure is more difficult to monitor and maintain than a centralized one. As a result, deployment of SSO techniques improves the security of an organization. If users need to remember multiple username-password combinations, they are liable to expose them. The chances of compromise are proportional to the number of combinations each user has to struggle with [9]. A preferred SSO arrangement is one that is independent of the application or platform where it is used; hence, it shields its procedures and techniques from the clients, while providing the feedback required by the centralized security administration [17].

A possible argument against SSO is that SSO qualifications are the pathway or key to the kingdom [18]. Fortunately, introducing additional security measures may diminish this danger. According to Clerq and Grillenmeier, these security measures are knowledge based (as in passwords); attribute based (typically biometrics, such as fingerprints), and possession based (such as a smart card or cryptographic token) [17]. These additional measures are considered in Section 2.4 below.

2.2 Security Considerations of SSO

SSO brings a certain level of ease, hence attackers can exploit this ease of use in several ways [19p. 17]. A common type of attack is known as a Denial of Service Attack, this type of attack causes the system to be unable to respond to legitimate requests due to being overloaded with requests [19p. 18]. Processing of a Security Assertion Markup Language (SAML) request consists of parsing the message, evaluating the request, and performing input/output operations. Parsing the message could take a significant amount of time. A denial of service attack against SAML may be reduced by requiring a unique origination of the request. The probability of a successful message forgery can be diminished by requiring that the message together with a timestamp be signed. Using too short of a time frame for the validity of the timestamp can be a problem, as a valid request may be falsely detected as a forgery because the request was delayed *en route* due to network congestion.

SAML's Simple Object Access Protocol (SOAP) binding does *not* require message integrity and confidentiality, as these properties are optional in SAML requests and SAML responses. SOAP over HTTP has some known problems [19p. 18]. These problems are further described in the following subsections.

An attacker can acquire both a SAML request and response by intercepting these messages (for example, by eavesdropping). Then an attacker can modify the assertions (carried in these messages) of another party known to the hijacked party. Encryption of the eXtensible Markup Language (XML) used to encode the SAML messages can prevent such attacks. However, research has shown that even with XML encryption, an AES-128 encrypted block can be decrypted with only 162 requests in less than one second [20]. Moreover, it is feasible to decrypt larger cipher texts. For example, decrypting 16,000 bytes takes only 17 minutes [20]. However, this may not be as large a

problem as it might seem, as limiting the request rate of a single origin makes this type of attack infeasible.

Message insertion involves inserting a message either in the request or in the response [19], while message modification alters the original request and the system can be compromised by an unexpected result due to such an alteration [19]. In contrast, message deletion of responses results in inappropriate action of the original requester. The action would be regarded as inappropriate as it would be an action that the requester does not intend to perform. The deletion of a response may result in no action at all [19].

Signing the message or transporting the message via SSL/TLS over HTTP would provide some additional security to the messages. However, this does not change the fact that there are still risks when either the key for signing is compromised or valid messages are simply recorded and subsequently replayed [19p. 19].

2.3 Security Assertion Markup Language (SAML)

Security Assertion Markup Language (SAML) is an accepted standard used in validation and authentication of data transmitted between domains. SAML has been specified by the non-profit Organization for the Advancement of Structured Information Standards (OASIS) [21]. SAML was introduced by OASIS in order to exchange identity assertions [22]. The SAML standards were introduced in 2002 (v1.0), 2003 (v1.1), and 2005 (v2.0) [23]. Through the use of this language, security can be realized via authentication between clients, domains, and devices that are called subjects.

It is important to understand the underlying domain concept used in SAML. A domain represents a service having identification methods for clients and acts as an identity provider. The domain needs to provide this identification method to third party service providers.

The following sections describe each of the concepts that OASIS defined in terms of relationships between profiles, protocols, assertions, and bindings. These relationships are illustrated in Profiles. The top-level concept in SAML is a *profile*, which defines how assertions and bindings are used in different SSO setups. Structurally a profile contains a set of bindings. The SAML profiles are listed in Table 2-1.

Table 2-1: Complete list of SAML profiles [18]

Profile	Description
Web Browser SSO Profile	This profile specifies how SAML messages can be used for web single sign-on with web browsers via the SAML bindings "HTTP Redirect", "HTTP POST", and "HTTP Artifact bindings".
Enhanced Client and Proxy Profile	In this profile, the Web Browser SSO Profile is enhanced to use SOAP and reverse-SOAP (PAOS).
Identity Provider Discovery Profile	This profile specifies a mechanism that a service provider can find a previously visited identity provider by a user.
Single Logout Profile	This profile specifies a single logout from multiple identity and service providers. It uses the SOAP, HTTP Redirect, HTTP POST, and HTTP Artifact bindings.
Assertion Query/Request Profile	This profile specifies a mechanism for querying an existing assertion from an asserting party. It uses SAML SOAP binding.
Artifact Resolution Profile	This profile specifies a dereferencing of an artifact over a SAML binding.
Name Identifier Management Profile	This profile implements the details on how to modify the value or format of a name identifier (ID) that identifies a subject. In this profile, the service provider and/or identity provider can issue such a request with the use of the SOAP, HTTP Redirect, HTTP POST, or HTTP Artifact bindings.
Name Identifier Mapping Profile	This profile specifies a mechanism for mapping one name identifier to another. This profile uses synchronous bindings such as SOAP.

2.3.1 Bindings

Bindings in SAML are defined as the transport protocols over which SAML messages can be sent [22]. The defined bindings are shown in Table 2-2.

Table 2-2: SAML defined bindings [22]

Binding	Description
HTTP Redirect Binding	This binding specifies the HTTP 302 redirection so as to send base64-encoded SAML-messages.
HTTP POST Binding	This binding specifies the base64-encoded SAML messages that can be sent in HTML forms via HTTP POST requests.
HTTP Artifact Binding	This binding specifies in what way artifacts can be sent from a message sender to a message receiver either in HTML-form data or URL-request parameters.
SAML SOAP Binding	This binding specifies how to send SAML-messages over SOAP.
Reverse SOAP (PAOS) Binding	This binding specifies whether an HTTP client acts as a SOAP responder.
SAML URI Binding	This binding specifies how to fetch an existing SAML assertion by resolving a uniform resource identifier (URI).

2.3.2 Protocols

According to OASIS, a protocol can be one of those shown in Table 2-3.

Table 2-3: SAML protocols [22]

Protocol	Description
Authentication Request Protocol	This is a protocol that specifies how an entity can request assertions from an assertion-issuing entity. In the Web Browser SSO context, this protocol is used to initiate the sign-on process where the service provider issues authentication request for the identity provider.
Single Logout Protocol	This protocol enables a user to be simultaneously logged out of the already signed on services. This process can be launched by the user, any of the service providers, or from the identity provider.
Assertion Query and Request Protocol	This describes the protocol used for querying or requesting assertions over an unspecified binding.
Artifact Resolution Protocol	This describes a protocol used for resolving artifacts, which are later used for retrieving assertions. An artifact is defined as a small fixed-length value. The party that receives the artifact uses the Artifact Resolution Protocol to ask the creator of the artifact to dereference the artifact in order to receive the actual message, which is typically an assertion.
Name Identifier Management Protocol	This specifies the protocol for communicating changes in value or format of a name identifier used to refer to a subject.
Name Identifier Mapping Protocol	This specifies the protocol to map a previously agreed upon (between two or more parties) SAML name identifier into a new name identifier.

2.3.3 Assertions

SAML assertions carry statements about a subject that an asserting party (the issuer of the SAML document) claims to be true [22]. SAML XML contains a set of assertion statements. Such a set allows the asserting party to assert security information about a subject. The three types of assertion statements are: Authentication, Authorization, and Attribution [22].

Authentication is the verification of the identity of the subject. This information is created by the entity that authenticates the requesting party. This authentication can be achieved via a hardware token, a password, or an X509 public domain key [23, p. 13].

An authorization is an assertion that assigns access rights to data or a channel to a subject. The issuing party decides which actions this user is authorized to perform [22].

An attribution is an assertion of an attribute of a subject that describes the property of that subject in question [24, p. 11]. An attribute can be user information or membership in a domain.

2.3.4 Example SAML request and response

A SAML assertion contains at least one statement about the user and is carried between parties in a SAML protocol response message, which itself must be transmitted using a transport protocol [22].

Figure 2-1 shows the relationship between SAML components, namely SAML protocol and SAML message structure.

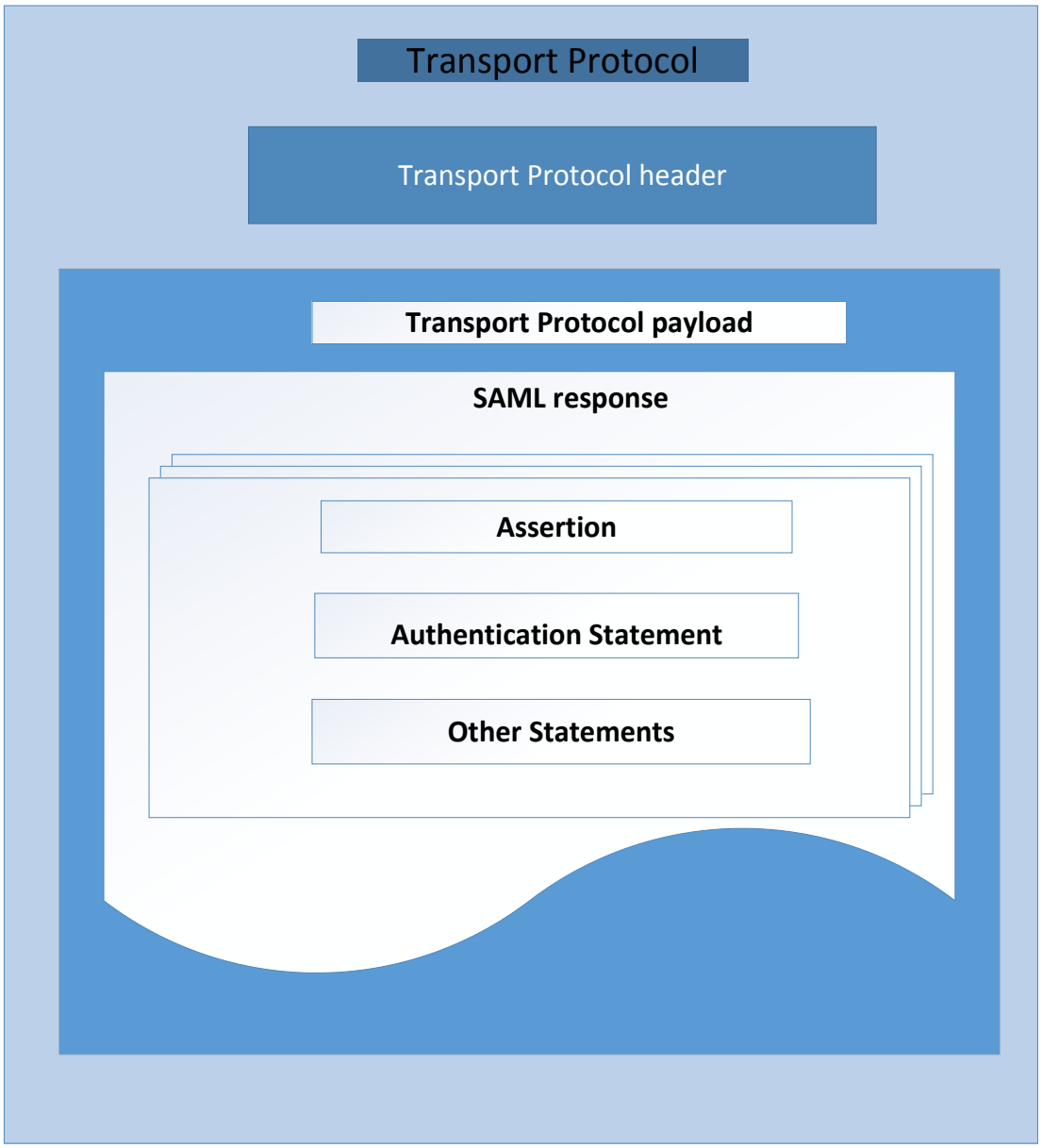


Figure 2-1: Relationship between SAML components (Adapted from Figure 5 of [21])

Encapsulation of SAML-assertions in a SAML response is as follows [22]:

```
<saml:Assertion Version="2.0" ID="34234se72" IssueInstant="2005-04-01T16:58:33.173Z">
        <saml:Issuer>http://authority.example.com/</saml:Issuer>
        <ds:Signature>...</ds:Signature>
        <saml:Subject>
                <saml:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">
 jygH5F90l
 </saml:NameID>
        </saml:Subject>
        <saml:AuthnStatement AuthnInstant="2005-04-01T16:57:30.000Z">
                <saml:AuthnContext>
                        <saml:AuthnContextClassRef>
 urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
 </saml:AuthnContextClassRef>
                </saml:AuthnContext>
        </saml:AuthnStatement>
</saml:Assertion>
```

The XML tags are:

Issuer	the issuer name [Required]
Signature	an XML signature for integrity protection and authentication of the issuer [Optional]
Subject	the subject of the statements in the assertion [Optional]
Conditions	must be evaluated when using assertions [Optional]
Advice	additional info that assists in processing of assertions [Optional]

Assertion statements contain zero or more of:

AuthnStatement	an authentication statement
AuthzDecisionStatement	an authorization statement (finalized in SAML V2.0)
AttributeStatement	an attribute statement
Statement	custom statement type

An example of an Attribute Assertion is:

```
  <saml:Assertion...>
 <saml:Issuer>                    ...                    /saml:Issuer>
 <saml:Subject>...</saml:Subject>
 <saml:AttributeStatement>
   <saml:AttributeName="PaidStatus">
   <saml:AttributeValue>Paid</saml:AttributeValue>
  </saml:Attribute>
 </saml:AttributeStatement>
</saml:Assertion>
```

An example of an Authorization Assertion is:

```
  <saml:Assertion                                                    ...>
 <saml:Issuer>                    ...                    /saml:Issuer>
 <saml:Subject>...</saml:Subject>
 <saml:AuthzDecisionStatement>
   Resource="http://foo.com/doit.cgi"
   Decision="Permit">
   <saml:Action>Execute</saml:Action>
 </saml:AuthzDecisionStatement>
</saml:Assertion>
```

There are a number of request/response protocols for communicating with a SAML authority. Some of these can be used for retrieving existing assertions, requesting authentication of a principal, requesting a simultaneous logout, and requesting a name ID to be mapped into another one.

An example SAML request query encapsulated in a SOAP message is as follows [22]:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap/envelope">
        <env:Body>
                <samlp:AttributeQuery xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:samlp="urn:ossis:names:tc:SAML:2.0:protocol" ID="saf23196 -1773 -2113-474a-
fe114412ab72" Version="2.0" IssueInstant="2006-07-17T20:31:402">
                        <saml:Issuer>http://example.sp.com</saml:Issuer>
                        <saml:Subject>
                                <saml:NameID Fornat="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName"> C=US, 0=NCSA -TEST, OU=User, CN=trscavo@uluc.edu </saml:NameID>
                        </saml:Subject>
                        <saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:uri" Name="urn:oid:2.5.4.42" FriendlyName="givenName"> </saml:Attribute>
                </samlp:AttributeQuery>
        </env:Body>
</env:Envelope>
```

As seen in the example above, the attribute starts with a namespace declaration and continues with the choice of the SAML protocol and message ID. The requesting party provides the attributes, such as `givenName` and the subject [22].

A SAML response query encapsulated in a SOAP message is as follows [22]:

```
<?xml version="1.0" encoding="UTF-8"?>
<env:Envelope xmlns:env="http://schemas.xmlson.org/soap/envelope/">
        <env:Body>
                <samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Version="2.0"
ID="i92f8bE230dc04d73e93095719d191915fdc67d5e" IssueInstant="2006-07-17T20:31:412"
InResponseTo="aaf23196 -1773 -2113-474a -fe114412ab72 ">
                        <saml:Issuer>http://idp.example.org</saml:Issuer>
                        <samlp:Status>
                                <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
                        </samlp:Status> ...SAML assertion... </samlp:Response>
        </env:Body>
</env:Envelope>
```

As seen above, similar to a SOAP request, the response starts with the namespace declaration and continues with SAML protocol with message ID [22]. The `ResponseTo` XML attribute indicates the request that the asserting party is responding to with a status code [22]. In the example shown above, the response indicates a success.

A SAML response may have different status codes depending on the result [26]:

urn:oasis:names:tc:SAML:2.0:status:Success	The request succeeded.
urn:oasis:names:tc:SAML:2.0:status:Requester	The request could not be performed due to an error on the part of the requester.
urn:oasis:names:tc:SAML:2.0:status:Responder	The request could not be performed due to an error on the part of the SAML responder or SAML authority.
urn:oasis:names:tc:SAML:2.0:status:VersionMismatch	The SAML responder could not process the request because the version of the request message was incorrect.

If the individual system requires more information than the minimum that SAML requires to fully leverage the security mechanisms that it has available, then it is possible to add the information that the system requires [25p. 14]. Thus, SAML is sufficiently flexible to allow each domain to assert its own security strategy in the existing frameworks *without* introducing difficulties for those security measures employed by other entities. Examples of these frameworks are Web Services Security (WS-Security) which employs security protection for SOAP messages and eXtensible Access Control Markup Language (XACML) for defining access control [22]. This flexibility enables SSO implementations to be easily deployed in web environments.

2.3.5 Web SSO and flow of authentication

A number of SSO implementations treat SSO differently depending on whether the target applications are web solutions or not. When an application needs to be integrated within a Web SSO framework, the target application might be a Web application. As a general rule, this implies a multi-layered architecture front-end of the Web application. Web Access Management Systems and many other systems utilize SSO techniques and solutions for security [27]. There are (so far) four major open Web SSO specifications: SAML Web Browser SSO Profile, InfoCard, OpenID, and OAuth.

Figure 2-2 depicts the workflow of Web SSO from an end user's perspective [23]. This figure illustrates a signup/login flow when a visitor attempts to login to a Relying Party (RP) Web site using one of her/his Identity provider (IdP) accounts [23]. This flow happens behind the scene and the user is presented with a form similar to that shown in Figure 2-3.

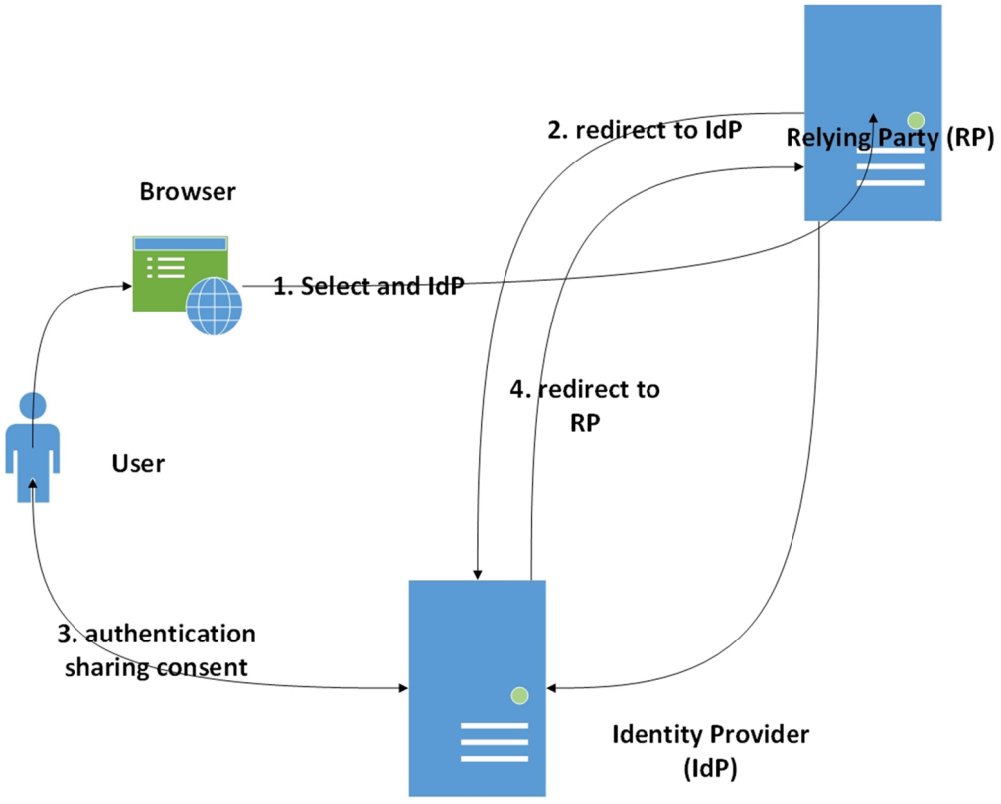


Figure 2-2: Sign up/ Login Flow in SSO (Adapted from Figure 5 of [23])

The steps involved in Figure 2-2 are [23]:

1. In the login form, the user selects an Identity provider from those presented by a Relying Party. A Web SSO-integrated login form typically combines password authentication with a list of IdP options that the user can choose from.
2. The RP forwards the user to the IdP so that he/she can be authenticated.
3. The user authenticates to their IdP by entering his/her username and password. After authentication, the IdP provides a profile sharing consent form for the user to enables the sharing of his/her profile information.
4. The IdP redirects the user back to the RP together with the requested profile attributes. Before granting access, the RP may prompt the user to complete a registration form in order to gather additional profile information or link to an existing account as illustrated in Figure 2-3.

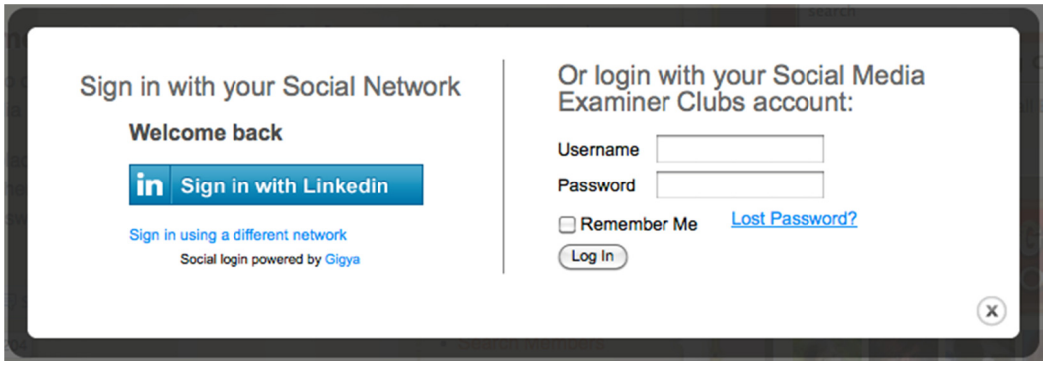


Figure 2-3: Sign up / Login Flow in SSO behind the scene (Adapted from Figure 5 of [23])

2.3.6 Shibboleth: An Example SSO System

This section gives an overview of SSO functionality using an instance of SSO implemented via the Shibboleth System[*]. As mentioned earlier, SSO is a verification system whereby a single user-password combination validates and authorizes a client to access all services within the same domain *without* having to authenticate to each program, application, or system. SSO serves the dual purposes of providing access based upon a single authentication, thereby making the system safe and at the same time the user need not remember multiple user-password combinations.

Shibboleth is flexible, robust, widely accepted, and is favored by Oracle, Apache, and Microsoft for managing SSO [28]. Shibboleth works as follows: When a client tries to access resources protected by the Service Provider (SP), then either prior authentication is necessary or if this is the first visit, then the user will have to be authenticate himself/herself [29]. For this initial authentication, the SP may redirect the client to the Where Are You From (WAYF) procedure [29]. Here the client is offered a list of client credentials that are needed to access the resource(s). If the client selects a credentialed organization from among those appearing in the list offered by the SP, then he/she is directed to a corresponding IdP. Using this IdP, the client follows the procedure for SSO specified by SP for accessing the resource [29].

After successful authentication, the client is associated with certain attributes, such as username, his/her association with the SP, etc. Now the user's browser is reconnected with the web resource that they had attempted to access with the newly authenticated user information provided by the IdP and the attributes of the client. The SP filters these attributes and IdP information to enforce the access control mandated by the SP. If this access control is successful, then the client is allowed to access the resource [30].

[*] https://shibboleth.net/

2.4 Post Authentication Methods

The process of determining whether a client is granted access to a system is done by an authorization decision. This decision is typically made following the client's authenticating himself/herself. The methods used for authenticating users to systems are the main focus of an authentication process, whereas the authentication of one system to another system mainly concerns their agreement and use of a set of security protocols [31].

Authentication of a human can be determined in one of the following ways [32]: something you know (passwords), something you are (biometric attributions), and/or something you have (tokens). Each of these methods of authentication is examined in the following section.

Post authentication refers to authentication, which occurs *after* a primary or prior authentication. Post authentication may be done to strengthen the existing authentication mechanism. Each of the three types of authentication methods might be utilized for post authentication. Using any of them would increase the complexity of the system's architecture, but would also increase the overall security assurance of the system. These post authentication methods can be integrated with SSO and SAML - if additional information is provided to SAML. According to Zeilenga and Melnikov, unless it is specifically allowed, only one SAML authentication exchange takes place in a protocol session. Multiple SAML authentication exchanges would only be allowed if they were specified in this protocol session, e.g. the issuer and the context class reference are needed as shown in the code below. However, if any one of these authentication exchanges is successful, there will not be any further processing or further attempts at authentication [33].

```
<samlp:AuthnRequest
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"ID="_bec424fa5103428909a30ff1e31168327f794
74984" Version="2.0" IssueInstant="2007-12-
10T11:39:34Z" ForceAuthn="false" IsPassive="false"ProtocolBinding="urn:oasis:names:tc:SAML:2
.0:bindings:HTTP-
POST"AssertionConsumerServiceURL="http://moodle.bridge.feide.no/simplesaml/saml2/sp/Assertio
nConsumerService.php">
    <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
        urn:mace:feide.no:services:no.feide.moodle
    </saml:Issuer>
    <samlp:NameIDPolicy
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"Format="urn:oasis:names:tc:SAML:2.0:nameid
-format:persistent"SPNameQualifier="moodle.bridge.feide.no" AllowCreate="true" />
    <samlp:RequestedAuthnContext
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"Comparison="exact">
        <saml:AuthnContextClassRef xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
            urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
        </saml:AuthnContextClassRef>
    </samlp:RequestedAuthnContext>
</samlp:AuthnRequest>
```

SAML authentication can be integrated with post authentication methods. These methods are the well-known authentication methods described in the next section. Post authentication methods are applied *after* a successful SSO authentication. If a second authentication method were to be applied as part of the original SSO authentication, this would simply be a *multifactor* SSO authentication, rather than a *post* authentication.

2.5 Well-known Authentication Methods and their Shortcomings

This section deals with the most well-known authentication methods (password management, biometric attributes, token-based authentication, and real time user behavior attribution) and examines their suitability as post authentication methods.

2.5.1 Passwords and Password Management

Passwords are widely used for authentication. Password based authentication is based upon a combination of a user identifier and a corresponding password. The username and password are generally represented as strings*. Various methods have been created to automate, to some degree, the generation of passwords. According to researchers, arbitrary sequences are more secure than meaningful ones (which would be easier to remember) [34]. Human beings have trouble remembering more than a certain number of arbitrary characters and the difficulty of remembering increases with the length of the sequence. To ease this, some transformation techniques were introduced in which a user generates a password in a way that he/she can recall the password with less hassle [35p. 168].

To further help users, password manager applications were introduced to help users keep track of their user name-password combinations. According to a recent study by Gaw and Felten [36], most such managers are built-in features of the browser itself (for example, based upon auto completion), as opposed to stand-alone programs that work in conjunction with an application program (e.g., Password Multiplier [37]). Stand-alone password managers minimize the difficulty of juggling multiple passwords. Users have problems transporting passwords from one PC to another with theses stand-alone password managers [34], and users may feel insecure about trusting these applications [36]. For utilities that enhance security through automated password creation (such as Passpet [38])), users may be uncomfortable about the fact that the process of creating passwords seems out of their control [34].

Although the length of a user generated password is nearly unlimited in most applications, Klein shows that users choose passwords that are easy to remember and these users utilize only a small subset of all possible passwords [39]. Unfortunately, this means that in practice there is a substantial risk of these passwords being cracked. As a result, passwords do not seem to be suitable for post authentication purposes, as the confidentiality of data cannot be ensured solely using passwords.

2.5.2 Biometric Attributes

Biological attributes cannot be precisely impersonated, reproduced, or easily replaced, as they are unique to each individual. For some time now, biological attributes have been widely used in the form of fingerprints, audio signal processing, and image recognition (such as scanning of facial features, especially eyes) in academic and commercial & organizational applications for security and authentication [40].

The drawbacks of using biological attributes are obvious: For example, an injury or burn to the thumb or fingertips may cause enough distortion that the user might be denied access to the whole system for the duration of the healing or perhaps even forever. Stained or soiled fingers are likely to be rejected during verification by machine scanners. Even as the use of fingerprint based techniques finds wider acceptability and use, the limitations in checking and calibration of scanners can be a cause for concern [41]. Additionally, the ability to forge fingerprints from photographs of an

* However, the password is generally stored as a hash to reduce the risk of the plain text version of the password being exposed.

individual has led many to question the continued value of using fingerprints, unless the scanner is properly supervised. This requirement for supervision may render these techniques unusable in practice.

Verification through audio sensors to authenticate a person using their unique voice, tone, and rendering has also been used in high-security zones for verification and identification for a long time [41]. Another potential application is when seeking access to functions in mobile cell phones, as the phones are already able to transmit and receive audio signals. Modern phones (especially smartphones) support advanced signal conditioning and noise filtering, thus using such phones is an attractive option for users. However, utilizing the audio signals received by the phone is hampered by extraneous factors, such as air pressure and temperature variations. Also, physiological attributes of the user due to factors such as a cold or cough can cause false rejection of a valid individual. Another concern regarding use of audio signals is the potential use of recorded sounds by an impostor in order to gain unauthorized entry into a system. Voice recordings have also been faulted for revealing our ethnicity and gender, and quite perceptibly, the emotional state and age of the user – when in many settings these properties are best kept secret. Once again, a potential solution is supervision, but again this may render these techniques unusable in practice.

Another physiological attribute recognized for uniquely identifying a user is the facial attributes that one possesses [41]: cheekbone, jaw-structure, nose, and eyes (specifically the iris). The measurable quantities are the color, sizes, shapes, and exact distances between relevant features. Some advanced recognition systems also exploit location of moles and wrinkles, while others utilize a three-dimensional model of facial features. However, facial recognition systems do not provide high quality authentication, even with the most advanced technologies prevalent today, as ambient light intensity, side-on postures, spectacles, smiles, make-up, facial expressions, headgear, haircuts, and beards make it difficult for the device to verify and authenticate users [42p. 1294].

In terms of being unique and difficult to impersonate, the iris of the eye does well, as recognition does not depend on color, but rather on the texture of the iris of the user. However, this feature also has drawbacks. Despite the uniqueness of the texture, this can be duplicated and faked with the use of contact lenses – much as voice based authentication can be subverted through recordings. These concerns have restricted the use of iris scanning automation to applications under human surveillance and iris based authentication is only used as an aid and not a conclusive verification on its own [43].

As described above, all of the variants of biometric methods have their shortcomings and they suffer from various risks of impersonation. Thus, their suitability for post authentication is disputed.

2.5.3 Token-based Authentication

Token-based authentication permits users to enter their username and password so as to acquire a token, which enables access to a specific resource for that user without again needing to enter their username and password. The token, which can be hardware or software, offers access to a specific resource for a time period.

2.5.3.1 Time Based One Time Passwords

In this approach, a hardware token computes a result based upon a secret and a clock. The authenticator system also knows these inputs, hence if the result of the remote and local computations match then the user is authenticated [44p. 46]. Due to time limitations on the token's validity and the limited time window used for comparison, such time based tokens cannot be used at a later time, hence the user must re-do the entire procedure.

2.5.3.2 Challenge-Response Authentication

Challenge-Response Authentication (CRA) is similar to time based authentication systems, but with the addition of a secret from the authenticating server [44p. 46]. Similar to Time Based One Time Passwords, this is a method for binding an identity to an entity over a communications medium *without* giving any information to eavesdroppers that might enable them to identify themselves as the originating identity.

The known shortcoming of CRA is that an attacker can modify a valid user's messages if plaintext messages are sent to the server, as the attacker could modify these messages. Moreover, the attacker will simply wait for you to authenticate and then take over your session.

CRA depends on properties of "one-way hashes". A one-way hash is a function that takes an input and returns a "hash value", where finding the input of the function from the hash is "computationally infeasible", i.e., computing the inverse of the hash would take far too many resources to be practical. An additional important property of such hashes is that there should be a very low probability that two different inputs will generate the same hash value.

Security tokens can be used for strong authentication. Although token-based runtime interaction can increase the strength of authentication, they are inconvenient for the user and costly for the service providers, hence they are a costly solution for existing systems [45]. Despite this they are widely used in many systems that seek to provide high security, for example, in on-line banking, digitally signing tax returns, accessing a patient's medical records, etc.

2.6 Post Authentication

Post authentication is a common way to authenticate a user to a system in which earlier authentication is a prerequisite for a subsequent authentication. Post authentication may include customized processing in between one authentication process and another.

There are various approaches used by post authentication mechanisms. Figure 2-4 depicts a simple post authentication process that uses multi-factor authentication and multiple authentication mechanisms.

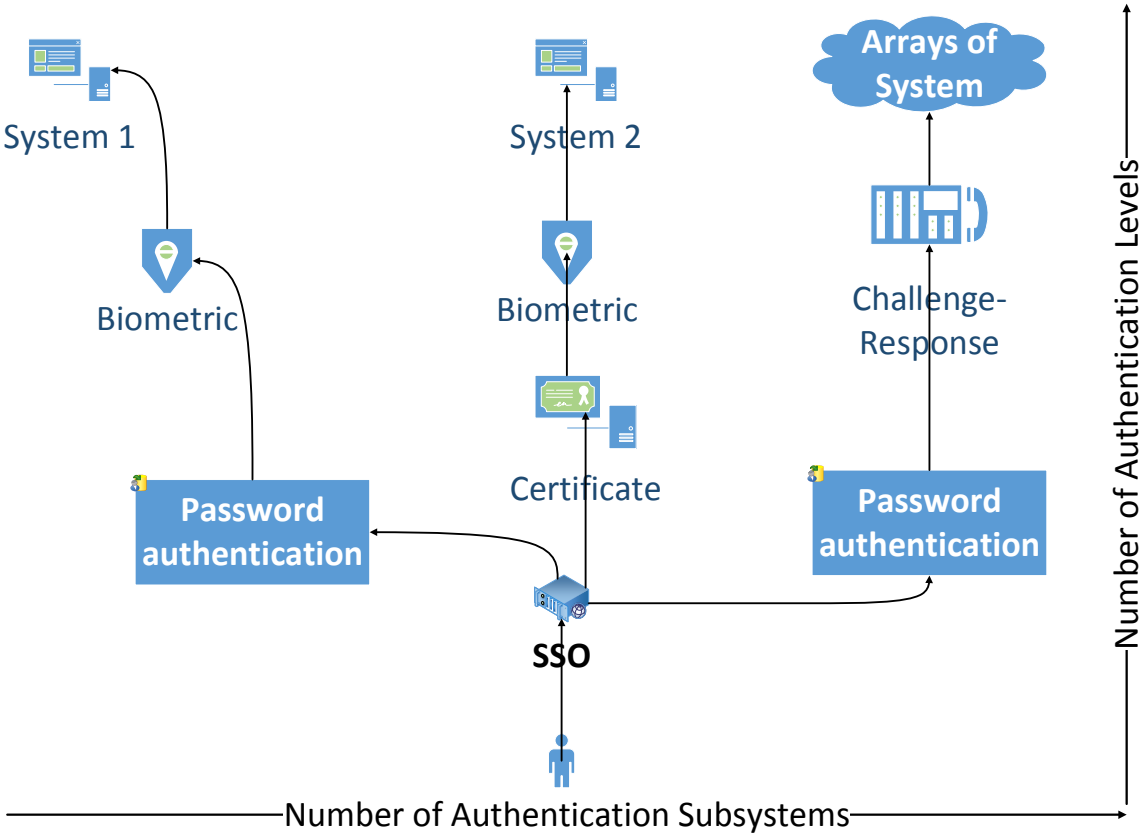


Figure 2-4: **Post Authentication process with multi-factor and multiple authentication mechanisms**

In Figure 2-4 the x-axis is the number of authentication subsystems and the y-axis is the number of authentication levels. In general, the best case for a user to authenticate is $K \times x + A$ and the worst case is $K \times x \times y + A$, where K and A are constants. In Figure 2-4, the best case for a user to be authenticated is $2K + A$ and the worst case for the same user is $3 \times 2K + A$.

Depending on the authentication methods, the post authentication process may add substantial security, while the user's experience might be hampered due to the need for excessive user interaction. For each additional layer of authentication, the system becomes more complex and the addition of the authentication and verification procedure to the existing layers increases the effort required to perform IT security auditing and tracing of the actions of the user [46].

The secondary authentication measures discussed above fall short of providing the confidentiality, integrity, and availability we are increasingly seeking in the digital world. Password protection techniques by themselves are weak defenses. As the number of login procedures increases, this weakness becomes more and more of a liability. Physical attributes are prone to change, despite what our perception tells us. Both password protection and biometric techniques are fallible and prone to being compromised. An additional disadvantage is that a forgotten or compromised password can easily be replaced, while a compromised physiological feature is irreplaceable. What do you do, for instance, if you happen to lose your thumb that was used for biometric evaluation? [43]

A post authentication method integrated with SSO may increase the level of security, but the rationale for using SSO in the first place is also defeated or at least severely weakened.

2.7 Challenge Handshake Authentication Protocol (CHAP)

The Challenge-Handshake Authentication Protocol (CHAP) is a widely used authentication protocol. This protocol is used when there is a need to establish a secure connection without exposing the password of the user. Figure 2-5 illustrates the message exchange of this protocol.

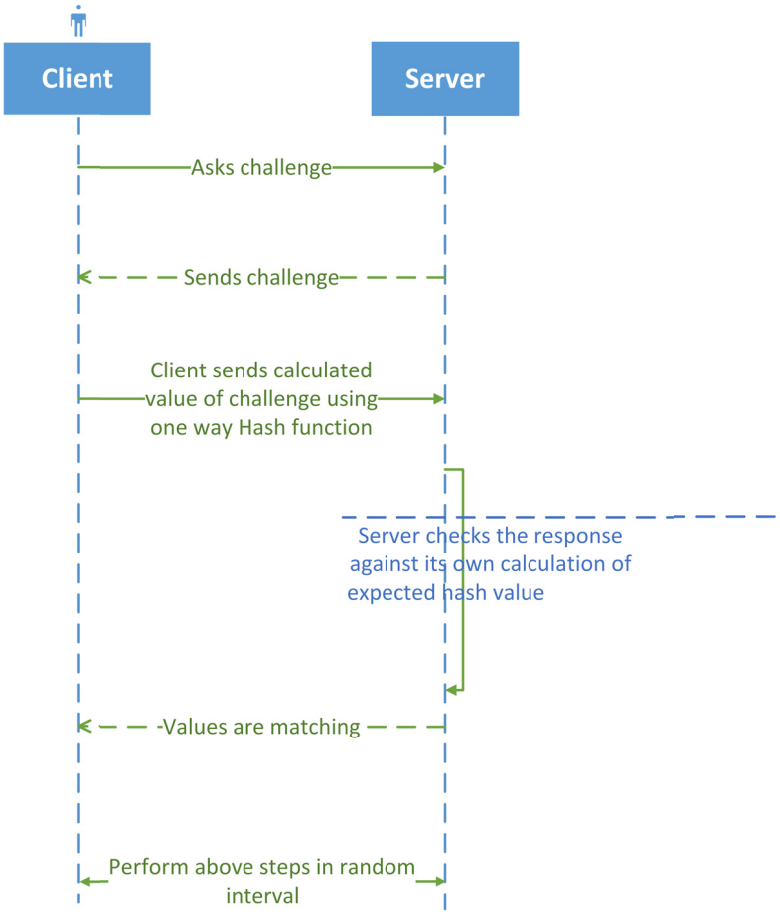


Figure 2-5: CHAP message exchange

CHAP is used for identity verification of the client via the server using a 3-way handshake [47]. As part of this handshake protocol, a hash of the password is transmitted rather than the plaintext password [47]. CHAP was primarily intended for use by hosts and routers that connect to a point-to-point (PPP) network server via switched circuits or dial-up lines, but might be applied to dedicated links as well [47].

CHAP has obvious security advantages over similar methods (such as Password Authentication Protocol (PAP) or Shiva Password Authentication Protocol (SPAP)), as the password is not transmitted over the network as plaintext and therefore cannot be captured [47]. Another advantage is that the protocol allows each message to be fingerprinted in a way that the legitimate sender and receiver can predict, but the attacker cannot [47]. This is accomplished via the challenge value and the message identifier that ties together. This prevents the risk of replay attacks [47].

2.8 Real Time User Behavior Attribution

The previously described authentication methods focus on authentication at a single point in time. However, there is in many settings a need for continuous authentication. This thesis proposes to use real time user behavior attribution to address this need.

Attribution is the assignment of effect to a cause [42, p. 268]. Attribution provides a way to record a meaning for an observed user behavior. The concept of real time user behavior attribution was introduced and patented by Cho and Min [49]. They used a two stage authentication process where the first authentication verifies the user's identity by comparing the authentication information with the template authentication information [49]. If the verification of the identity of the user succeeds in this first authentication stage, then a second authentication stage is used to verify the identity of the user by comparing their behavior patterns with template behavior patterns [49]. The authentication information and behavior patterns are stored in a database [49].

A similar method was proposed by Shi et al. in which real time user behavior is employed as an authentication mechanism, invisible to the user, which they term *implicit authentication* [50]. This authentication is used either as a second factor in authentication or as a primary factor by itself for authentication [50].

There are related methods, also using real time user behavior attribution, but based on biometry and location. One such method is *keystroke dynamics* that uses pattern recognition for password authentication. Shi et al. claim that keystroke dynamics is a viable biometric for providing security [44, p.7]. Another method is *location based access control*. In this method, once a principal's location is verified as an allowed location, the principal is granted access to a particular resource according to the given policy [52].

In order to incorporate real time user behavior attribution into SSO, one needs to consider the False Accept Rate (FAR) and False Reject Rate (FRR). FAR is the percentage of authentication decisions that allow access to a user who should *not* be authenticated [53p. 451]. FRR is the percentage of authentication decisions where a user is incorrectly denied access [53p. 451]. The relationship between these two ratios is shown in Figure 2-6. Minimizing both rates would be ideal [47, p. 452], as the user will not be disturbed by the system and the system will limit access by the user according to attribution via user behavior. However, beyond a certain threshold it is not possible to reduce one rate without increasing the other one.

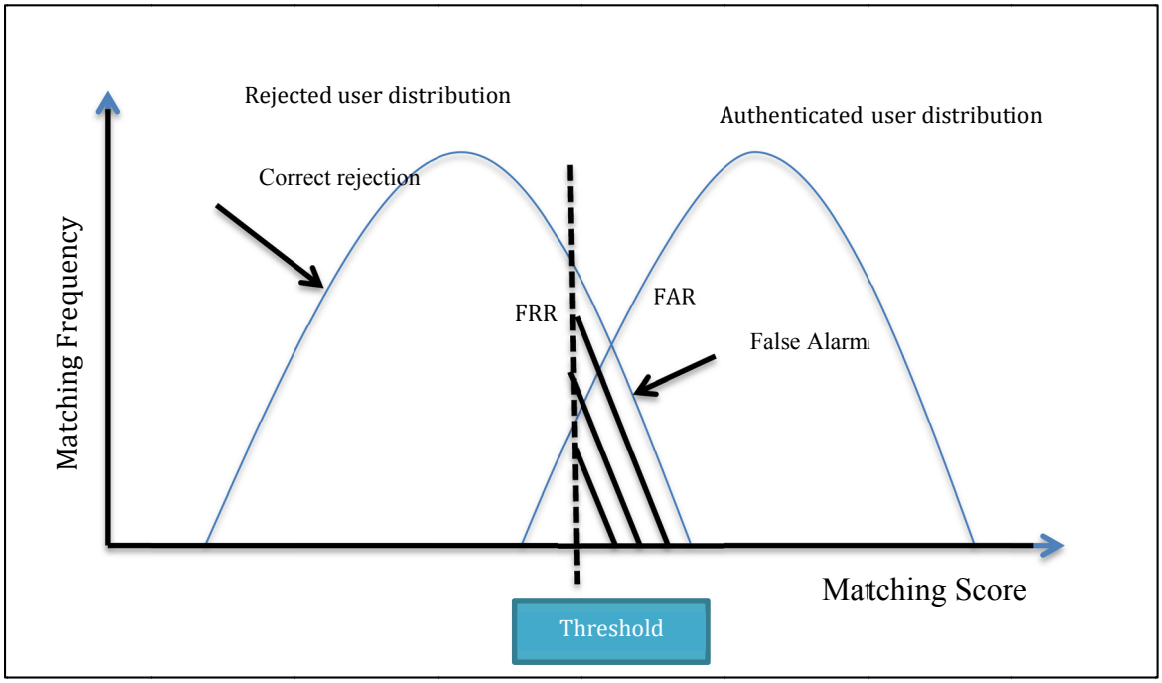


Figure 2-6: **False Accept Rate (FAR) and False Reject Rate (FRR)**

3 Methodology

I selected the design science research methodology for this thesis project as it is the most suitable methodology for an artifact development research project [4]. According to Hevner, et al., the created artifact should be evaluated based upon its "utility, quality, and efficacy". The design of the artifact should assess existing theories, should be verifiable, and the solution should be relevant to the problem [54].

To understand the design problem, we need to understand the problem domain. Herbert A. Simon defines the environment as consisting of people, organizations, and technology [55]. To keep track of these entities, we need a knowledgebase in order to do information science research. Figure 3-1 illustrates the relationship between environment, knowledgebase, and information science research. Using these three components, the problem can be solved.

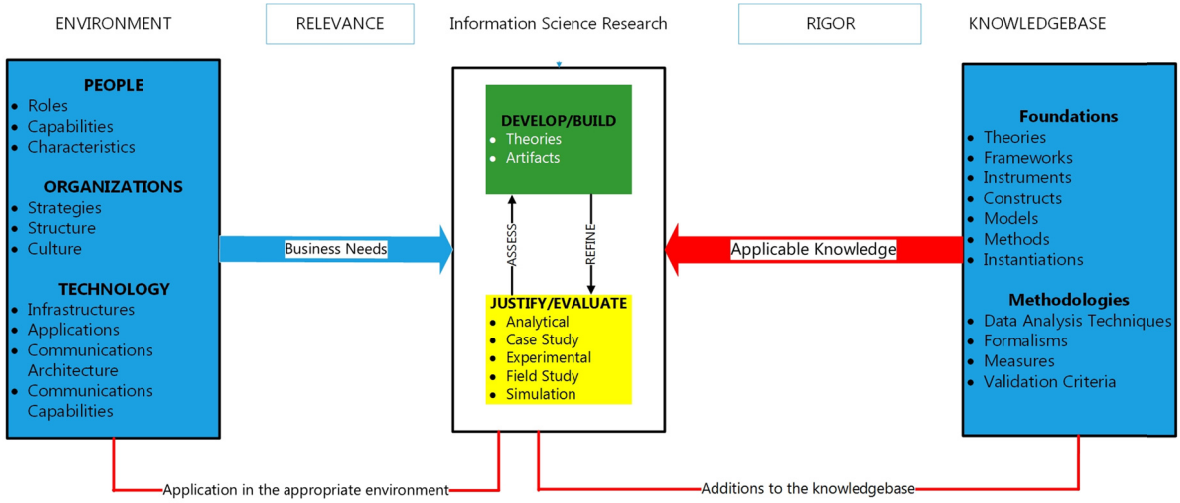


Figure 3-1 Information Systems Research Framework (Adapted from [50, p. 80])

According to Hevner et al. there are two complementary paradigms by which information science research is conducted. Behavioral science research has the goal of truth, leads the research through the development and justification of theories, so as to explain the hypothesis related to the identified problem. On the other hand, design science research, has as a goal utility, hence it evaluates the artifacts designed with respect to their ability to meet the identified problem requirements [54]. In order to formulate the artifact development process, Hevner et al. define guidelines for design-science research. These are listed in Table 3-1.

In this thesis, I claim that incorporating real time user behavior attribution into SSO together with a challenge and response mechanism provides security that prevents the "key to the kingdom" problem. When there is only one logon that enables access to multiple resources, security should be provided by more than one set of user credentials [56]. To achieve this security, I claim that my method is similar to lazy authentication and provides greater ease of use than requiring the user to authenticate repeatedly.

I designed an artifact to realize the above combination by building upon an existing SSO solution by incorporating a real time user behavior attribution extension. This artifact must show that SSO with a secondary authentication mechanism will be slower than my implementation. The designed artifact will be evaluated using statistical analysis. The artifact contributes to existing SSO systems as they should be able to seamlessly incorporate my implementation. The research for a potential SSO solution which adapts a real time user behavior attribution extension requires a deep analysis of existing SSO solutions and existing libraries for its implementation.

The evaluation of the artifact depends on quantitative research and analysis. In the following section, I use power analysis to define the size of the user group for experiments with the artifact. I perform a risk-benefit analysis to show how my artifact can be used by a security manager and to examine where my method is applicable and where it is not.

In order to evaluate FAR & FRR values for the artifact, the incremental time used by the additional authentications for each user in a set of tasks and the cost of collecting, processing, and making decisions based upon the behavioral data are quantitative values, hence I have not selected a qualitative research approach. For these reasons, this thesis project has not utilized qualitative research analysis methods [57].

Table 3-1: Design-Science Research Guidelines (Adapted from [50, p.83])

Protocol	Description
Guideline 1: Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation
Guideline 2: Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
Guideline 3: Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods
Guideline 4: Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Guideline 5: Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
Guideline 6: Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
Guideline 7: Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

3.1 Methods and Research Process

A power analysis is done to determine the size of the user group so that it is possible to observe the magnitude of the effect that I expected to see. Although for the data to be "meaningful" or statistically significant, my implementation needs to consider the entire population of interest; however, this is impractically given the project's time and resource constraints [58].

Power analysis is computed either before a study's final data are collected (*a priori*) or after the study's final data (*post hoc*). Yu underlines that the power analysis should be incorporated before the study, as according to Yu, this analysis consists of two-steps:

> *"(a) Hypothesizing the effect size which is most likely to occur based upon the study's theoretical and empirical context" [58].*

> *"(b) Estimating how probable the study's results are to result in statistical significance if this hypothesis is correct" [58].*

To show statistical significance, it is important to distinguish between True Positive and True Negative results and to minimize the False Positive (Type I Error, α Error) and False Negative results (Type II Error, β Error).

The power analysis also concerns the probability of an impact, difference, or relation. This is defined as $P = 1 - \beta$. It is important to emphasize that increasing the size of the user group (or making the research more sensitive) does *not* necessarily mean that the results will have statistical significance. It may simply mean that the observed differences are not *statistically* significant.

While calculating the desired size of the group, it is important to consider several factors:

- One needs to define the null hypothesis,
- The statistical test must be defined with respect to the variables in the hypothesis,
- We must determine the difference (or variance) indicating statistical significance, and then
- We estimate the desired size of the group.

It is important to underline that power analysis is **not**:

- "the probability of obtaining statistical significance if a specified effect size is obtained" [58],
- "the probability that a test will yield statistical significance and it is independent whether the data is collected properly or not" [58],
- "the probability of obtaining statistical significance if the null hypothesis is false" [58].

If the hypothesized effect size is inappropriate, then the null hypothesis might be false and any power analysis based upon it is inappropriate [58]. If the true effect size is greater than hypothesized effect size, then power will be an underestimated, while if it is smaller, then the computed power will be an overestimate [58].

Power Analysis can be done by using different statistical tests. One type of test is the Chi-Squared test. This test is widely used to compare ratios from different groups. The Chi-Squared test does *not* take in account continuous variability, but rather it aims for Boolean/dichotomous results, such as Yes/No, Male/Female, or Dead/Alive. As the variables are not continuous, there is no average or standard deviation as required in a T-test, instead the ratios are used. In my experiments, I am also using Chi-Squared Test since I am also aiming for Boolean results, and continuous results such as authentication time.

Another type of tests is Student's t-test, a parametric test to compare two groups' averages of continuous variables. In order to use the t-test, the distribution within each of the groups must follow a normal (or Gaussian) distribution. Therefore, in order to use t-test one needs to:

- Define the null hypothesis,
- Define the Effect size, E,
- Define the standard deviation, D,
- Calculate the ratio E/D, and
- Define Type I Error (α Error) and Type II Error (β Error).

In order to define E and D, the power analysis should be considered. Increasing the standardized E will decrease the required sampling size for a given confidence level. Most of the time, the effect size is assumed to be between 0.1 and 0.5. According to Cohen, if there is no previous study or sufficiently detailed information to hypothesize an effect size, it is acceptable to hypothesize the effect size as 0.50 [58].

The confidence interval is an interval that is constructed in a way so that, in the long run, a given proportion of these intervals will include the unknown true parameter value. With a given level of confidence, the true value of the parameter is estimated in the target population [59].

3.2 Research Hypotheses

In this thesis, I have two research hypotheses. My first research hypothesis is that incorporating a real time user behavior attribution extension into SSO will lead to a login process that is faster than SSO with a secondary authentication mechanism incorporated into it. My second research hypothesis is that SSO with the behavior extension does not produce an unacceptable rate of false positive and false negative results.

The null hypothesis would be true if:

- SSO with a behavior extension is **not** faster than SSO with a secondary authentication mechanism[*],
- SSO with the behavior extension produces an **unacceptable** rate of false positive and false negative results.

3.3 Research Hypothesis Discussion

There are a number of questions raised by the first bullet in the research hypothesis:

1. What can be measured?
2. What is the correct statistical test to use?

There are a couple of questions raised by the second bullet in the research hypothesis:

1. What is unacceptable?
2. Why is it important to measure FAR and FRR?

3.3.1 Measurements

Generalizing the question, "What can be measured" leads to the questions: What will be different? Can I do a statistical analysis of what I am measuring? Should I keep the SSO implementation in my experiments or only measure what is different?

My current experimental design for the first hypothesis depends on two experiments (e.g., two different web applications) with control and test experiments. In the control experiment, I utilize SCRAM after SSO, whereas in the test experiment, I utilize my lazy authentication. In this lazy authentication, if the user's behavior does not match the expected behavior, then I invoke SCRAM. I will measure the authentication times in both experiments. The total time to perform authentication can be expressed as:

Initial + Repeated authentication: $SSO_{time} + n * SCRAM$

Initial + Behavior based authentication + SCRAM when user's behavior does **not** match the expected behavior:

$$SSO_{time} + \ additional\ delay\ of\ behavior\ based\ authentication + (n - q) * SCRAM$$

In both cases we perform a SSO, thus in terms of measuring a *difference* in login time, it does not matter how long an SSO authentication takes (as we will subtract this term from both equations). Therefore, it does not seem necessary to measure the time required to perform SSO. For this reason the experiments must measure the difference in time between performing SCRAM n times (where n is the number of gateways to different domains that the system is to protect) and the additional delay of behavior based authentication together with SCRAM performed n-q times,

[*] This means that the incremental time required by the additional authentications for each user requires more total time than my mechanism.

where q is the number of times that behavior based authentication was able to authenticate the user without needing to perform SCRAM.

It is trivial to note that since both expressions for total time to perform authentications are greater than the time for SSO it is clear that requiring more than one sign on will take longer. However, to prevent the "key to the kingdom" problem we have to either do repeated authentications or only perform such authentications when we believe that the user's behavior is not as expected.

In Figure 3-2, the red curve represents the control group's total authentication time (k is the time required for the initial SSO authentication; m is the time required for one additional SCRAM based authentication) and the blue curve represents the experiment group's total authentication time when using behavioral authentication. In this figure we have assumed that the additional time to perform behavior based authentication is "a" units of time and this is repeated periodically with time interval (Δ).

If we assume that behavioral authentication can take place either in the background (when the system is idle) or be performed in parallel with the normal operations of the system, then the actual authentication time is "a" when behavior authentication succeeds and m when behavior authentication fails (since when behavior authentication fails to authenticate the user, the fallback is SCRAM which takes m units of time).

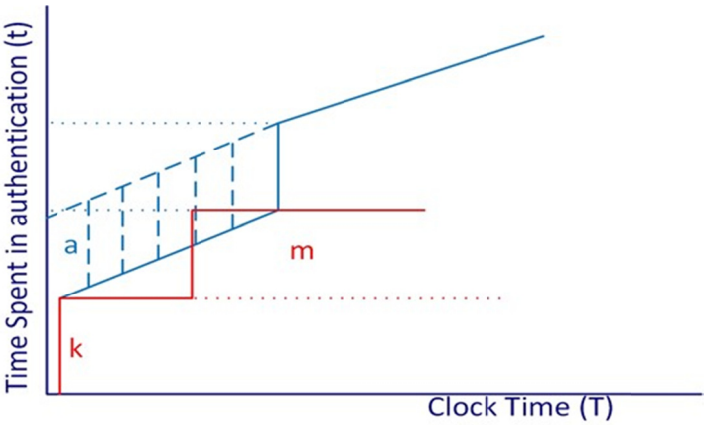


Figure 3-2: Time spent in authentication versus Clock Time

Even if "a" is small, it could grow to cumulatively be larger than m. This requires some analysis to determine how often the behavior analysis should be invoked. See Section 3.4 for details of this. Additionally, if the behavior analysis is based upon completing some task, then there is also a question whether there should be an upper bound on the time (for example, the user should complete the task within 120 seconds, otherwise SCRAM authentication should be invoked). In addition to above argument, if the network is not fast, then the elapsed authentication time for a post authentication method might reach values that are not really desired.

3.3.2 Statistics

I planned to use Student's t-test to assess the time spent in authentication (beyond k) as this is a continuous variable, for this reason I planned to use Student's t-test power analysis. However, when I start analyzing the data, I saw that I needed to use the Mann-Whitney U Test (see Section 3.10.1 Planned Statistical Analysis). This statistics is used to assess the null hypothesis for authentication times:

SSO with a behavior extension is **not** faster than SSO with a secondary authentication mechanism*.

I also studied FAR & FRR and then use a Chi-Squared Test. This is to assess the null hypothesis:

SSO with the behavior extension produces an **unacceptable** rate of false positive and false negative results.

Measuring FAR and FRR is important for an authentication mechanism which depends on the user behavior. The risk- benefit the analysis of this authentication method depends on FAR and FRR measurements and within relevant cost analysis, one can decide whether to invest on.

The unacceptable rates for FAR and FRR are depending on the confidence interval. Confidence intervals consist of a range of values (interval) that act as good estimates of the unknown population parameter.

In this thesis, the acceptable rate for FAR and FRR should be represent with 95% probability that the confidence interval will hold the true value of the parameter. The confidence level is the complement of respective level of significance, for the hypothesis, a 95% confidence interval reflects a significance level of 0.05.

3.3.3 Risk Management

The identification and assessment of risk is defined as risk management. Risk management involves the economical application of resources to minimize, monitor, and control the probability of risk [60]. To quantify the risk, it is important to identify each potential risk, analyze them, estimate their probability of occurrence, and assess the consequences if it does occur [61].

The next step is to manage the risks, i.e., decide on measures and corrective actions and to appoint people who will be responsible and monitor these actions. Risk assessment is the process of discovering and documenting the risks present in an environment. Within a broader context, known as risk management, risk assessment is considered part of the due care an organization applies to the operation of a computerized information system.

3.3.3.1 OCTAVE

There are a range of existing methods to apply within Risk Management field. A comprehensive approach to risk management is the Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) [62]. OCTAVE is a risk methodology that empowers organizations to adjust the security of basic data resources against the expenses of providing insurance and discovery controls. By following the OCTAVE method, an organization can settle on information-protection decisions based on risks to the confidentiality, integrity, and availability (CIA) of critical information resources. To identify the information security needs, a cooperation between the operational or business units and the IT department is necessary.

Using a three-phase approach, the OCTAVE method examines organizational and technology issues to assemble a comprehensive picture of the information security needs of an organization. These phases are defined as follows:

Phase 1: Build asset-based threat profiles
Phase 2: Identify infrastructure vulnerabilities
Phase 3: Develop security strategy and plans

* This means that the incremental time required by the additional authentications for each user requires more total time than my mechanism.

3.3.3.2 NIST SP 800 documentation

National Institute of Standards and Technology (NIST) SP 800 documentation approaches Risk Management by collecting various best practices from the United States of America's federal agencies. NIST documentations for security and risk management are available publicly. These documents have been reviewed by government and industry. The specific documentation for risk management for Information Technology is NIST SP 800.30. This document provides practical guidance for assessing and mitigating risks. This documentation can be used as guideline for risk management process.

3.3.3.3 The ISO/IEC 17799-27005:2013 Model

The first widely used model for security management is ISO/IEC 17799:2005 model. This document contains best practices of control objectives and controls in a number of areas of information security management. However, this model is currently outdated and new standards are defined under ISO/IEC 27005.

ISO/IEC 27000 series is an umbrella specification which specifies an information security management system (ISMS). An ISMS is an arrangement of strategies dealing with information security management or IT related risk.

ISO/IEC 27005 is part of ISO/IEC 27000:2013 series. This standard is designed to assist in the implementation of information security based with a risk management approach [63]. This model aims to enable a business to establish, implement, review and monitor, manage, and maintain an effective ISMS. This standard conforms to ISO/IEC 17799:2005 "code of practice" on information security management.

3.3.3.4 FRAP

The Facilitated Risk Analysis Process (FRAP) involves analyzing one system, application, or segment of business operation at a time and includes both business managers who are familiar with business information and security experts [64].

3.3.3.5 COBIT

Control Objectives for Information and related Technology (COBIT) is a framework for the governance and management of enterprise IT that leverages practices and security control [65].

COBIT provides metrics and maturity models to measure their achievement, and identifyies the associated responsibilities of business and IT process owners [66]. COBIT is known as being suitable for e-commerce solutions [61]. It fact, it is one of the most commonly used frameworks to comply with the U. S. Sarbanes-Oxley Act, also known as "Corporate and Auditing Accountability and Responsibility Act" [67].

3.3.4 Risk–Benefit Analysis

Risk–benefit analysis consists of methods which might be relevant for different areas and addresses the question of whether a risk is "acceptable" [68]. Organizations are expected to perform a cost-benefit analysis to determine the value of the information assets to be protected and the loss in their value if those information assets are compromised due to the exploitation of a specific vulnerability. This decision-making process is called a *cost benefit analysis* or an *economic feasibility study*.

According to D. W. Straub, the cost of development or acquisition of behavior extension includes [61]:

- Training fees (cost to train personnel and end user),
- Cost of implementation (installing, configuring, and testing hardware, software, and services),
- Service costs (vendor fees for maintenance and upgrades), and
- Cost of maintenance (labor expense to verify and continually test, maintain, and update).

Benefit is the value to the organization of using controls to prevent losses associated with a specific vulnerability. The benefit is usually determined by valuing the information asset or assets exposed by the vulnerability and then determining how much of that value is at risk and how much risk there is for the asset [61].

To estimate the potential loss, it is important to identify the financial value of the information asset. While this might not be easy to do, the following financial aspects might be considered [61]:

• Value retained from the cost of creating the information asset,

• Value retained from past maintenance of the information asset,

• Value implied by the cost of replacing the information,

• Value from providing the information,

• Value acquired from the cost of protecting the information,

• Value to owners,

• Value of intellectual property,

• Value to adversaries,

• Loss of productivity while the information assets are unavailable, and

• Loss of revenue while information assets are unavailable.

Upon valuation of the financial value of the asset, the estimation of the potential loss that can occur from the exploitation of a vulnerability can be evaluated as follows [61]:

- What damage could occur and what financial impact would it have?
- What would it cost to recover from the attack, in addition to the financial impact of damage?
- What is the single loss expectancy for each risk?

The calculation of the value associated with the most likely loss from an attack is known as Single Loss Expectancy (SLE). It calculation is as follows [61]:

SLE = asset value (AV) X the exposure factor (EF)

Valuation of an asset is an extremely hard task, whereas the exposure factor which is the percentage loss that would occur from a given vulnerability being exploited is harder.

To determine overall potential loss per risk, the annual rate of occurrence (ARO) of the threat must be evaluated. This threat is defined as Annualized Loss Expectancy (ALE) and is calculated as follows [61]:

ALE = SLE × ARO

Finally cost benefit calculation shall be [61]:

CBA = ALE (prior to control) – ALE (post control) – ACS (annual cost of the safeguard)

This calculation can be done either before a control or safeguard is implemented, to determine if the control is worth implementing, or, after, when they can be calculated have been implemented and have been functioning for a time.

3.4 Proposed Solution and Brief Comparison

The proposed solution incorporates a real time user behavior attribution extension, hence it avoids post authentication and incorporates lazy authentication. The behavior pattern is accessed according to a supplied template.

In well-known classical post authentication methods, authentication is not continuous, but rather is dichotomous. Therefore, if a user fails to authenticate, this user cannot access the next system that utilizes another authentication method, hence the user cannot reach the target system.

Considering the existing SAML request-response schema, real time user behavior attribution provides *continuous* authentication by comparing previously recorded real time user behavior with the user's current behavior. A comparison of these behaviors should result in continued user access or an interruption of the user's interaction with the system.

In Figure 3-3, "k" represents the time required for the successful SSO authentication, "a" represents the work done internally by checking the user's behavior against behavior pattern template. This means that "a" is the time spent assessing the user's pattern behavior. Finally, "m" represents the authentication time for SCRAM authentication which kicks in due to a behavior authentication indicating that there is a miss-match between the expected and the observed behavior.

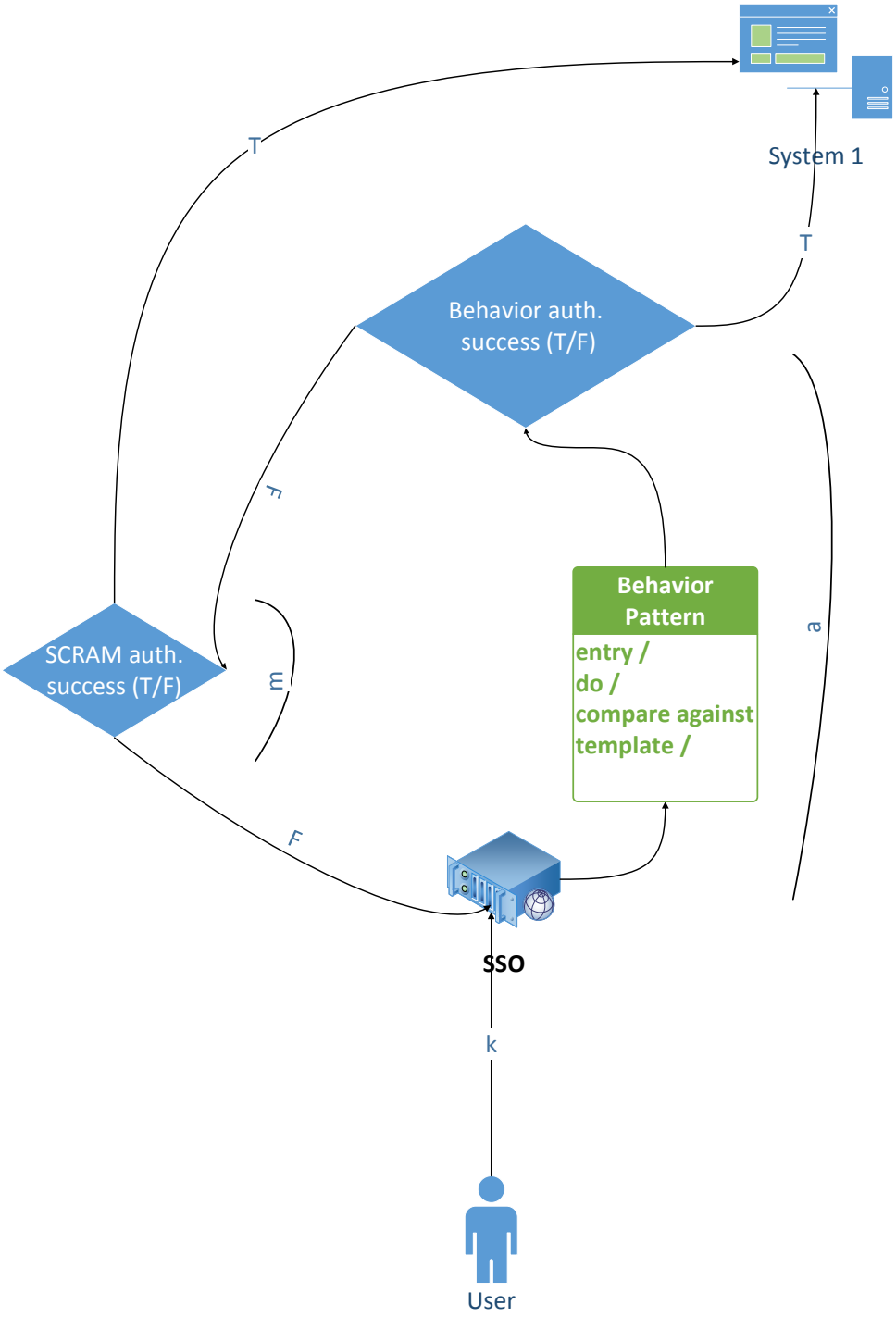


Figure 3-3: Proposed Solution

3.4.1 Fallback to Post Authentication

Due to the potential for misinterpretations of a user's interactions, a fallback mechanism is important. Such a fallback mechanism should react by initiating a re-authentication of the user as rapidly as possible. This section describes a potential authentication method for this fallback mechanism, specifically using the Salted Challenge Response Authentication Mechanism (SCRAM) Secure Hashing Alhorithm-1 (SCRAM SHA-1) protocol.

3.4.2 Salted Challenge Response Authentication Mechanism (SCRAM)

Although CHAP (see Section 2.7) is a good authentication method for providing confidentiality and is resistant against replay attacks, CHAP requires that the secret is available in plaintext form. In addition, CHAP is not very useful for SSO, since every possible secret needs to be maintained at both ends of the link [47]. For this reason I have chosen another method.

SCRAM does not have the problem mentioned above and it is a Simple Authentication and Security Layer (SASL) mechanism. SASL is an application framework to generalize the SAML authentication which is already part of SSO [69]. For these reasons, I chose SCRAM over CHAP. Moreover, SCRAM provides an existing solution that can be integrated with my artifact.

Figure 3-4 illustrates the SCRAM authentication exchange [70]. SCRAM exchanges a number of hashed items between client and server. Just as with CHAP, these items are not susceptible to a playback attack. The exchanged items authenticate the client to the server and authenticate the server (as the holder of the secret) to the client.

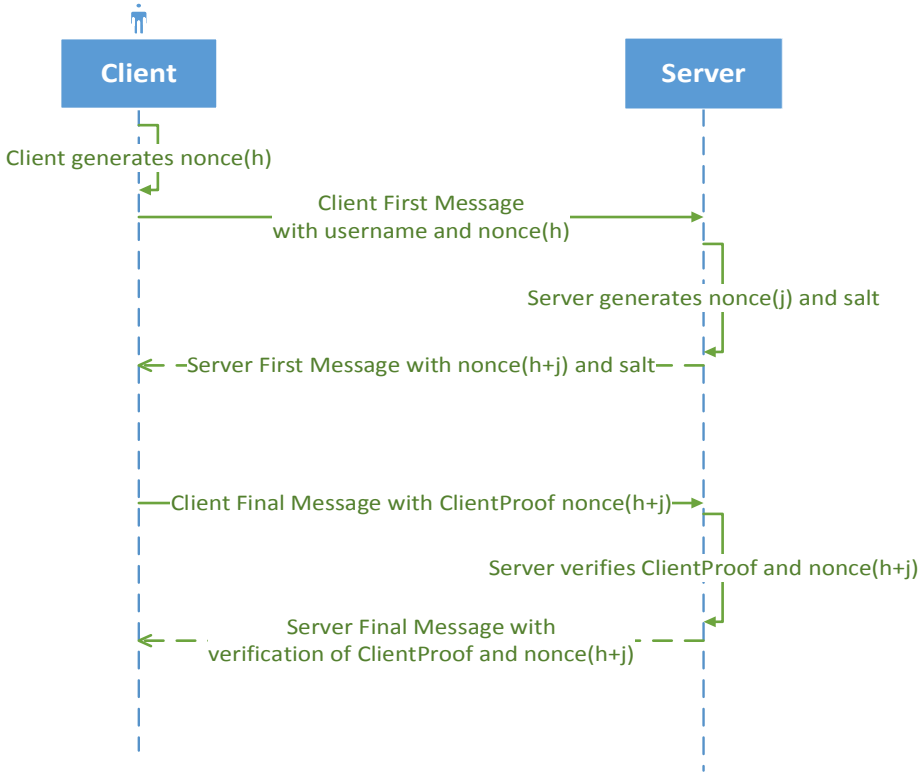


Figure 3-4: SCRAM message exchange

SCRAM is designed to be used with any hash algorithm. However, for compliance with SASL, SCRAM is expected to be used in conjunction with the SHA-1 algorithm [70].

The salt value in Figure 3-4 should be a random value specific either to the server or to the user's account on the server.

SCRAM is intended to be used with Transport Layer Security (TLS) or another data confidentiality mechanism to achieve confidentiality [70]. In order to simplify my implementation I have chosen not to use TLS; hence, my implementation is susceptible to man-in-the-middle attacks. A future improvement of the implementation *should* utilize TLS.

3.5 Experimental designs and planned measurements

In order to assess the null hypothesis, that SSO with a behavior extension is **no** faster than SSO with a secondary authentication mechanism, one needs to perform a meaningful statistical analysis with two different experiment groups of users. Each group performs the same operations using the same user interface. The interface of the application consists of controls that the users can utilize to perform a certain task (See Section 3.6.1 "User Task").

The test volunteers will be divided into two groups: (1) control group and (2) elapsed authentication time experimental group. The elapsed authentication time experimental group will use the SSO mechanism together with the proposed behavioral extension in order to perform a task. However, to reach the user interface (UI) for the task, the user must carry out several steps using the UI and must interact with the system. If there is a deviation from the expected user behavior, then the fallback mechanism which implements SCRAM SHA-1 is triggered.

The elapsed authentication time control group will use the SSO mechanism together with a secondary authentication mechanism. This secondary authentication is the same authentication as the fallback mechanism used for the first group. The UI that the user utilizes is the same and the task to perform is the same.

The assumption that is being made is:

$$\text{Time spent}_{(\text{SSO + Behavioral extension + SCRAM SHA1} * \alpha)} \leq \text{Time spent}_{(\text{SSO + SCRAM SHA1})}$$

Where α represents the binary variable, depending whether SCRAM authentication is in place or not (α is 0 if SCRAM authentication is not in place, 1 if SCRAM authentication is in place).

As stated earlier in Section 3.3.1 since the time for SSO is on both sides of the equation we can remove it to simplify the equations to:

$$\text{Time spent}_{(\text{Behavioral extension + SCRAM SHA1} * \alpha)} \leq \text{Time spent}_{(\text{SCRAM SHA1})}$$

With this reduction in the equation, the experimentation for elapsed time authentication method does not require SSO authentication tests (See Sections 3.5.2.1 "Control Group for Elapsed Authentication Time" and Section 3.5.2.2 "

Experimental Group for Elapsed Authentication Time").

In order to assess the null hypothesis, that SSO with the behavior extension produces an **unacceptable** rate of false positive and false negative results. We use a similar experimental setup as described above. In order to assess false positive and negative results requires two different experiment groups of users. These groups represent the illegitimate and legitimate users. For consistency in experiments, the time until rejection is measured while the users in the group conducting the task defined in Section 3.6.1 "User Task".

3.5.1 Artifact Technologies

The proposed solution has been implemented using Visual Studio 2010 and the .NET 4.0 Framework. The SCRAM SHA-1 implementation for the fallback mechanism was originally implemented by Fizachi [71]. I fine-tuned this implementation and integrated it with my artifact. Further details about the choice of software tools can be found in Section 3.11 on page 72.

The proposed solution uses a template to codify the expected user behavior. This template is static and prepared prior to the experiment. ASP.NET web pages have a lifecycle, which includes postbacks [72]. The implementation records each user action in the browser. Not all of these user actions will necessarily cause a postback. For this reason instead of .NET postbacks, which send the entire page and its contents back to the server, I use an AJAX method which is not as resource intensive as a .NET postback.

To summarize, a client side AJAX function captures the user's action and transfers this action to the server side on the fly. The artifact compares this action against the template (stored in an XML file). If there is a deviation from the expected user behavior, then the fallback mechanism is invoked. This fallback mechanism will make a decision as to whether the user should retain his/her (authenticated) session or be kicked out of the system. The time spent for authentication is measured and logged to a separate file. This timing information is important for statistical analysis reasons (see Section 3.10).

3.5.2 Test environment/test bed

For each experiment, users are informed that they need to reach a web address which hosts the web application. The users do the experiment in the browser of their choice and they are free to use whichever OS they prefer on their client computer. The URLs used for each of the two groups are shown in Table 3-2.

Table 3-2: URLs used by the two test groups

Group	URL
Control	http://77.235.62.4/AftonblatteWebApplication.UI.ControlGroup
Elapsed authentication time experimental	http://77.235.62.4/AftonblatteWebApplication.UI.ExperimentGrou p

For evaluating FAR and FRR, the web application for the legitimate and illegitimate user groups resides at: http://77.235.62.4/AftonblatteWebApplication.UI.ExperimentGroup

The dedicated server that hosts the web applications is shown in Table 3-3.

Table 3-3: Dedicated server specification

Processor	Intel® Xeon® L5520 running at 2.27 GHz (2 processors)
Memory	2.00 GB RAM
Operating system	Microsoft Windows 2008 R2 Standard SP1 64-bit operating system
Web server	Microsoft Internet Information Services (IIS) 7.5.7600.16385

The reason for such low memory on the server is that the prices for dedicated server is not really cheap when it is hosted in Europe and since aftonblatte.se had no notable traffic and since there was no slowness in the machine prior to the experiment, I did not consider increasing the RAM. The allocation of RAM is decided by the OS and on average, w3p.exe process is using 14 000 K on idle state and if the allocation gets doubled when the user interacts with the browser in the behavior extension experiments. On the other hand, SCRAM server which listens incoming request uses 5 000 K on average and when there is a need for SCRAM authentication request, OS allocates around 8 000 K on average.

Logging is separated for each experiment by dedicated server. There is one log file for each of the control and experiment groups for the elapsed authentication time experiments. These log files also contain information about whether the fallback authentication mechanism is triggered or not.

Initially I planned to provide the test bed to the sample groups as a pre-configured virtual machine. To do that I planned to give each participant an ftp URL to a site from which the user could download a virtual machine. However, in there was a problem as the high network load caused by these downloads lead to incomplete or erroneous reception of the VMs. As a result, I gave up on the idea of distributing virtual machines. Instead, I used a dedicated server of aftonblatte.se and the URL provided to the users point to this server.

During the actual experiments, the users must use the latest version of Google Chrome, since (due to limited time) I was unable to test the web application, the integrated SCRAM SHA authentication method, and the behavior extension authentication method with other browsers. It is important to emphasize that if a user using this system utilizes another client computer, the browser settings **must** allow Javascript to handle AJAX calls.

3.5.2.1 Control Group for Elapsed Authentication Time

The control group for elapsed authentication time experiment would have the following protocol (as shown in Figure 3-5). Due to the reason that the elapsed time authentication method does not require SSO authentication explained in Section 3.5 “Experimental designs and planned measurements”, the actual SSO interactions are not included in the experiments. The SSO interactions in Figure 3-5 are included for the sake of completeness.

The experimental design flow sequence with SSO is as follows:

1. The user initiates an SSO login and if the IdP identifies the user, then the user is challenged by the SCRAM post authentication method. Otherwise, the user will not be logged in by the IdP.
2. If the user passes the secondary authentication mechanism, then the user is logged into the target system.
3. The user interacts with the system to perform the task.

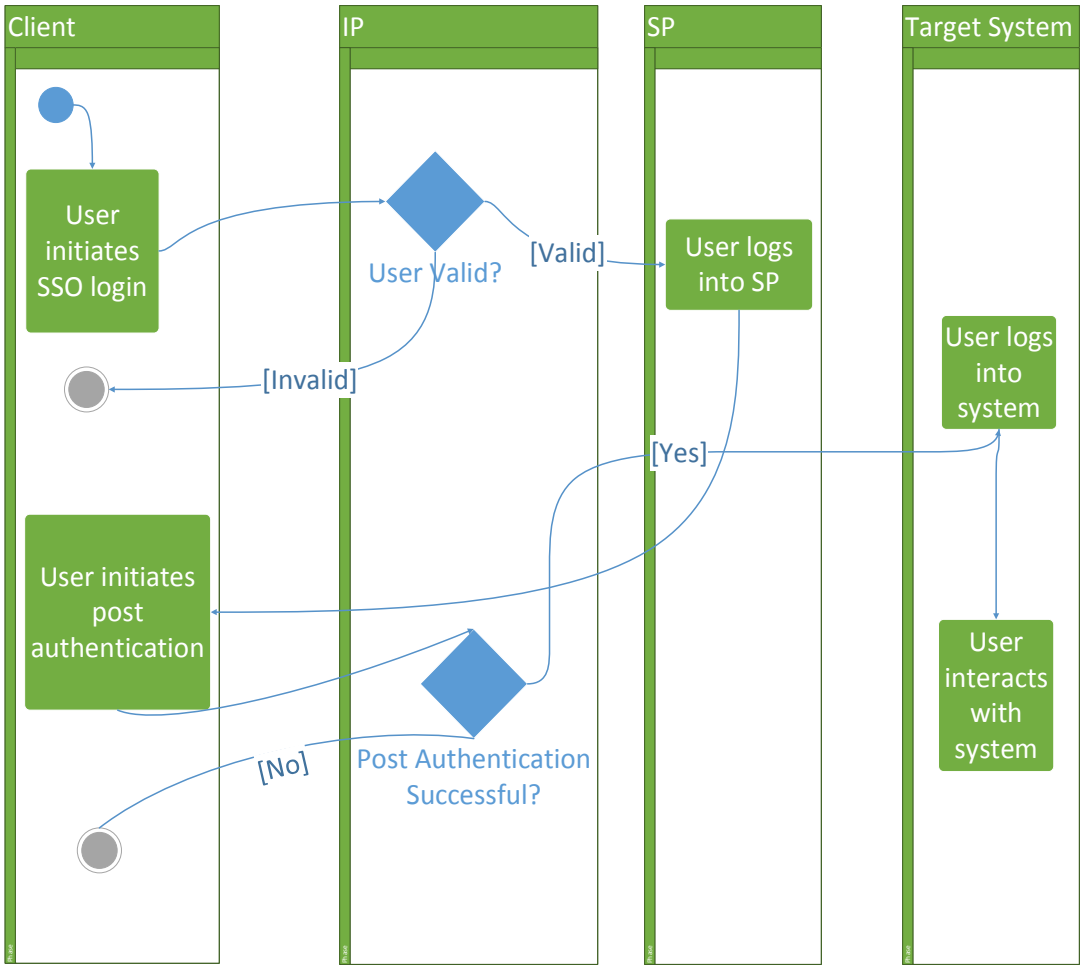


Figure 3-5: Control Group User System Interaction sequence flow

The conducted control group experimental design sequence flow is as follows:

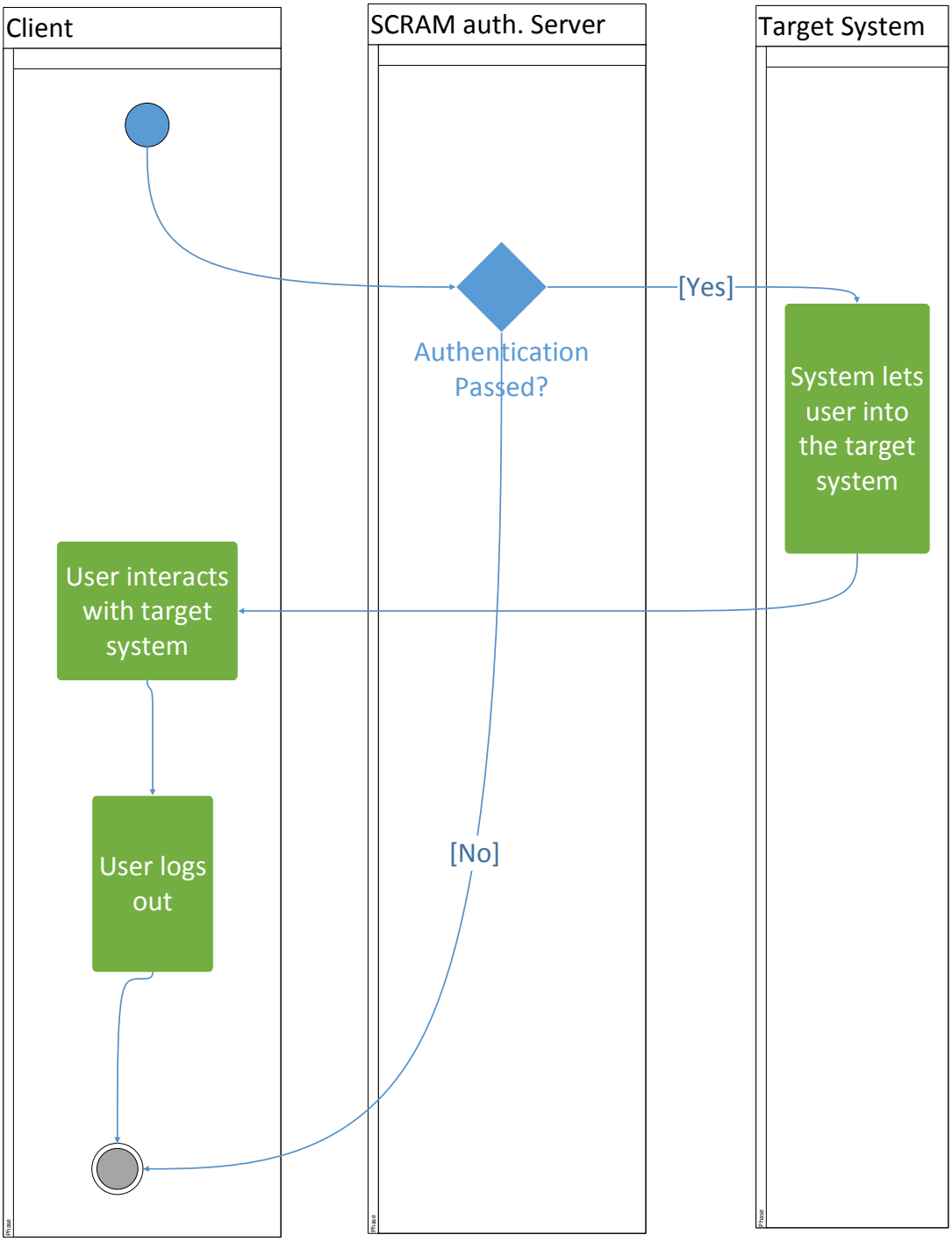


Figure 3-6: Conducted control group experimental design sequence flow

3.5.2.2 *Experimental Group for Elapsed Authentication Time*

The experiment for measuring elapsed authentication times consists of the following steps (shown in Figure 3-7). Because the elapsed time authentication method does not require SSO authentication (as was explained in Section 3.5), the actual SSO interactions are not included in the experiments. The SSO interactions in Figure 3-7 are included for the sake of completeness.

The experimental design flow sequence with SSO is as follows:

1. The user initiates an SSO login and if the IdP identifies the user, then the user is granted the right to access the service on the target system. Otherwise, the user will not be logged in by the IdP.
2. Every user action is considered to be a behavior by the hosting system. These actions are analyzed by the artifact (running in the target system) and compared to the reference template.
3. If the user's interaction is not in keeping with the behavior specified in template, then the user is logged off the SP's system and the system makes a request using the SCRAM authentication method.
4. If the user passes the SCRAM authentication, then he/she is still logged into the SP and can continue to interact with the target system to fulfill their task; otherwise he/she will be logged out of the target system.

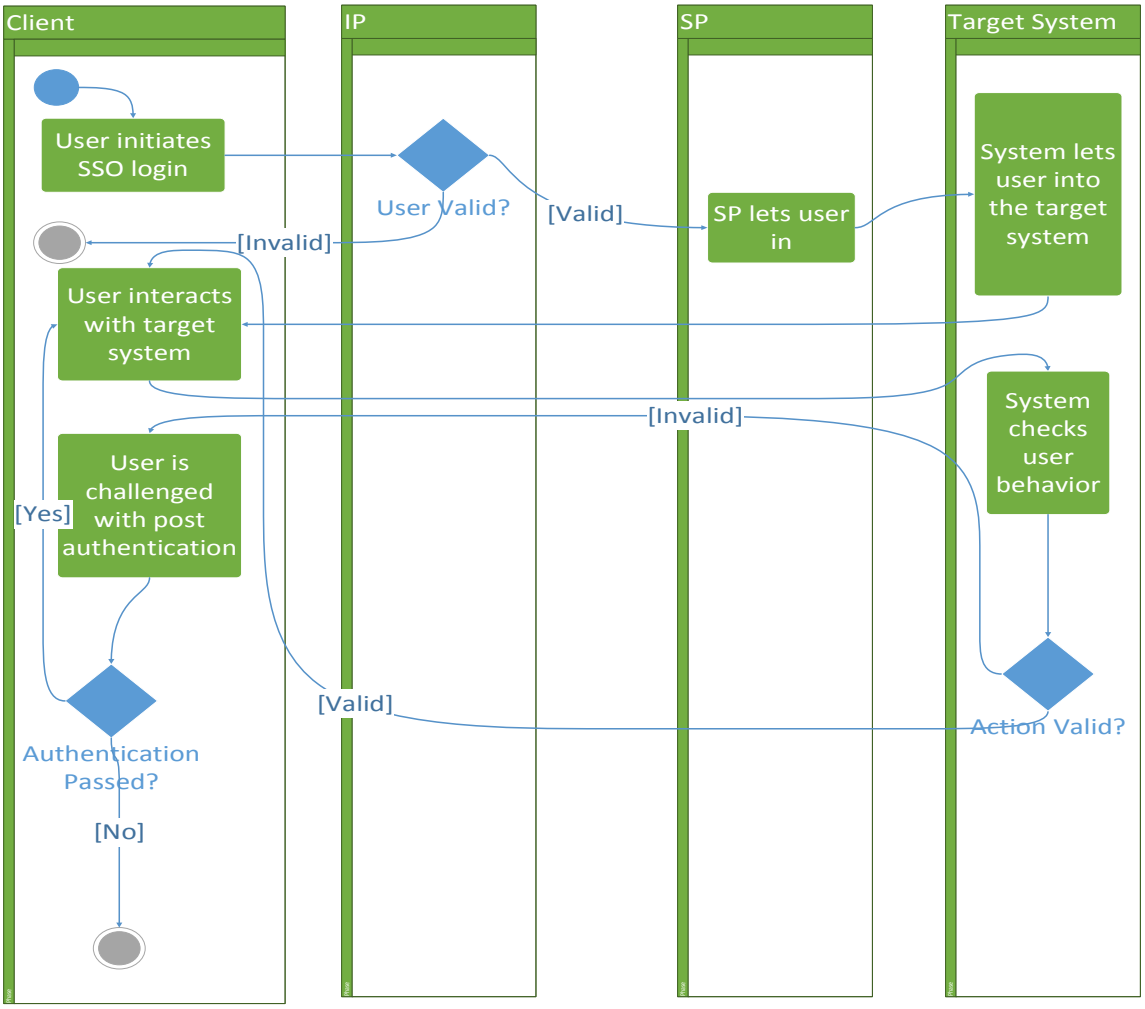


Figure 3-7 **Experiment flow sequence**

The experiment group's experimental design flow sequence is as follows:

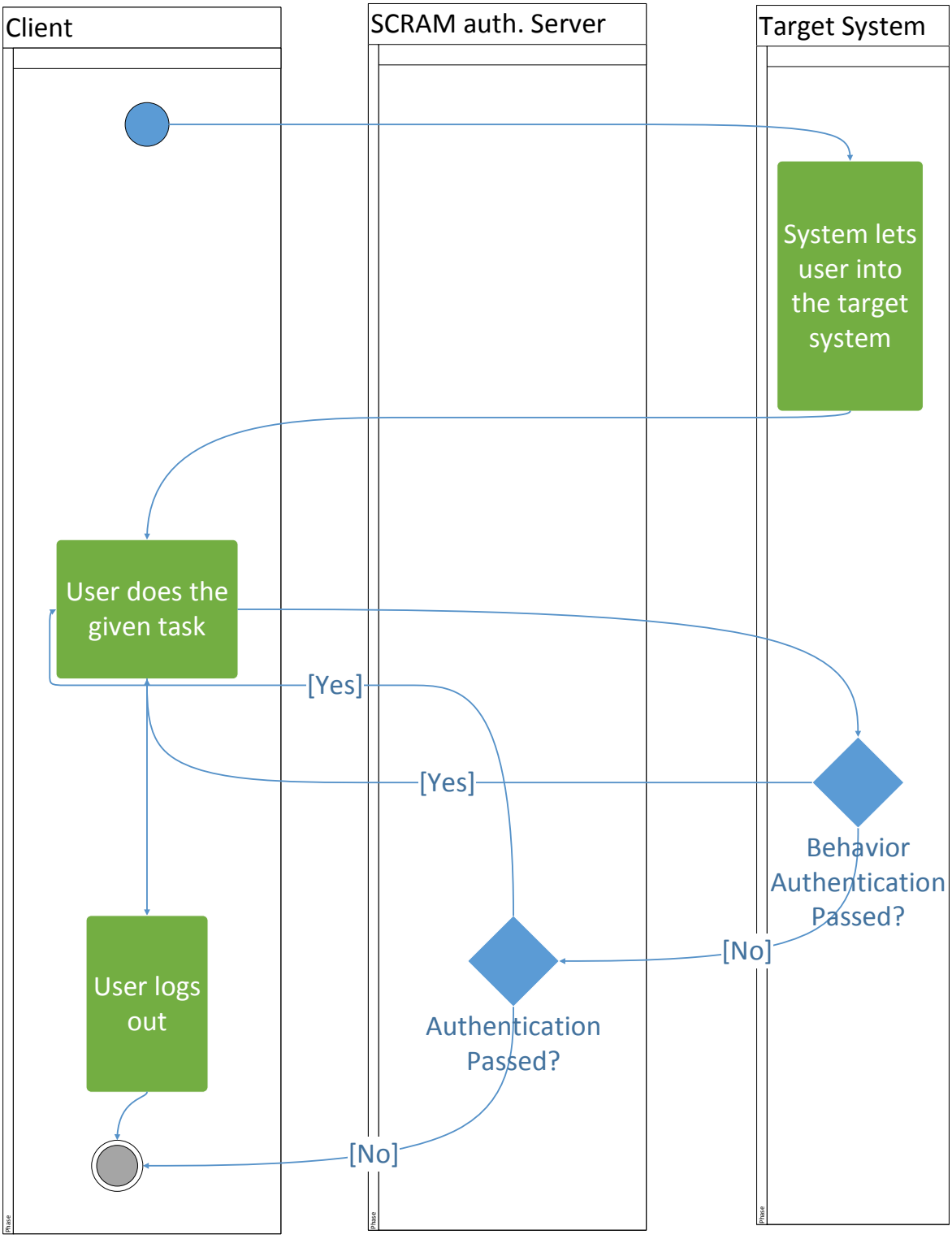


Figure 3-8: **Conducted experiment group experimental design flow sequence**

3.5.3 User Interfaces

In information technologies, usability is the main interest when it comes to human-computer interaction and how users feel when interacting with a system is often not considered [73]. In contrast, Norman emphasizes that there are several aspects which are relevant for interface use, specifically "knowing, doing and feeling" [74]. The reasons to do a task might affect the user's feelings. These reasons can be classified into two when the instrumental reasons are for work or school; whereas non-instrumental reasons are for fun and leisure [75]. In this thesis, the users are approached for a non-instrumental reason in that they volunteered to participate in the experiments for fun.

When implementing new systems, it is important to consider a design that creates a positive psychological effect on the user [76]. Therefore, the UI design requires appealing and organized interfaces that can have positive psychological effects on users participating in the experiment. For this reason, I used an existing website called aftonblatte.se. This website focuses on satire news. The site has contributions both from regular authors and readers of the website. The website is available in both Swedish and English languages and has been in existence since 2012.

The reader layout is same for all control and experiment groups. This layout is illustrated in Figure 3-9 and Figure 3-10. For elapsed authentication time experiments, the addresses to the web pages differ; whereas for legitimate and illegitimate user experiments both the control and experiment groups use the same URL as the elapsed authentication time experiment group.

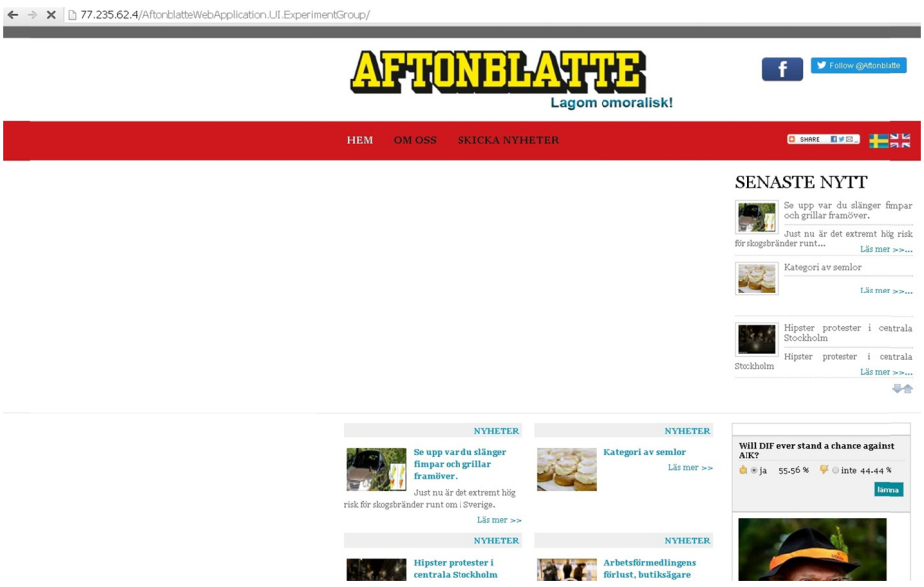


Figure 3-9: Aftonblatte Experiment Group reader layout view

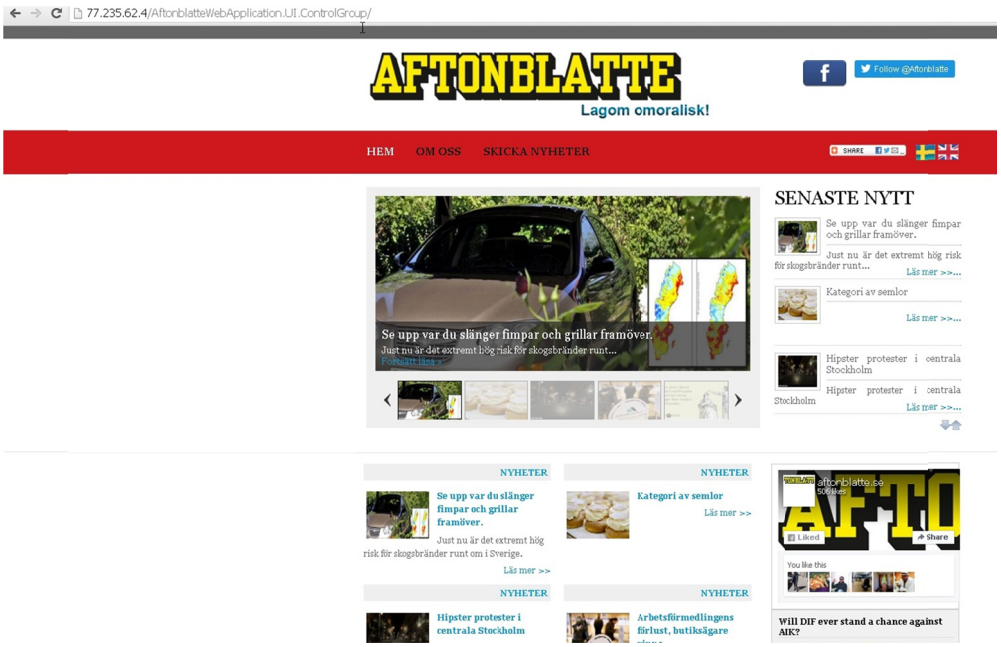


Figure 3-10: Aftonblatte Control Group reader layout view

The admin layout for all experiments is the same. However, for the elapsed authentication time experiments, the addresses in the address bar are different as shown in Figure 3-11 and Figure 3-12. In all of the experiments, the participants have the same page to add and manage the news (see Figure 3-11 and Figure 3-12).

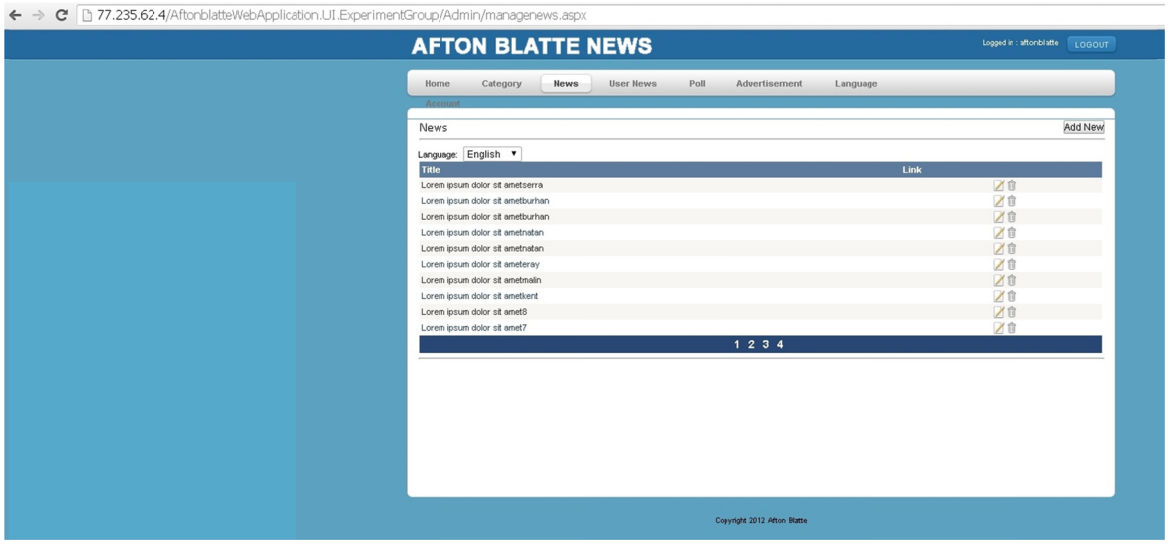


Figure 3-11: Aftonblatte Experiment Group admin layout view

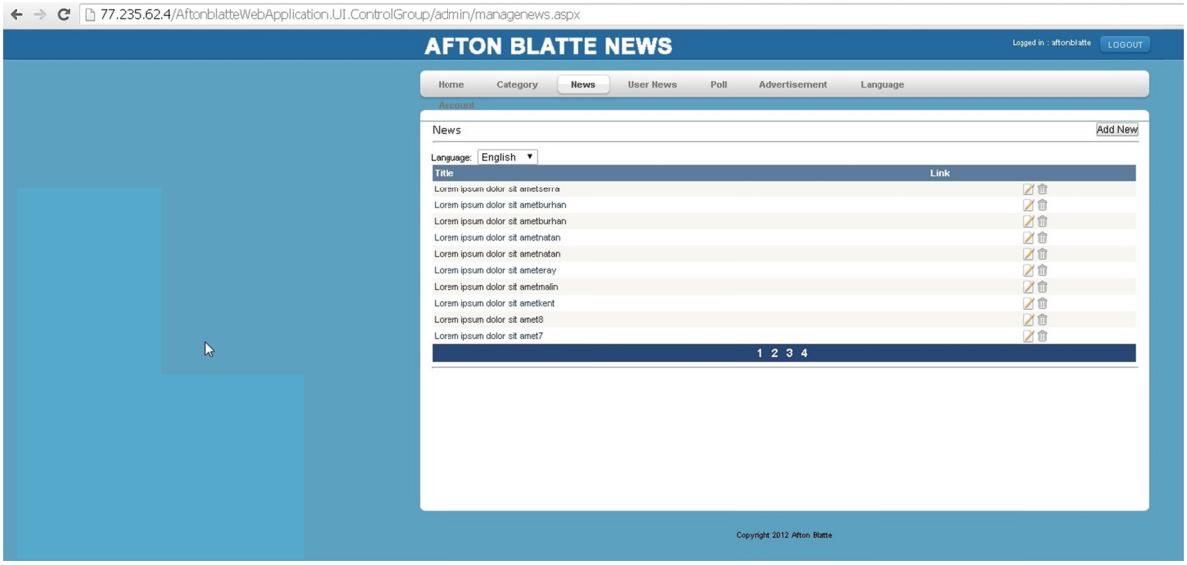


Figure 3-12: Aftonblatte Control Group admin layout view

The UIs used for both the control and the experiment groups follow the design described in Sections 3.5.3.1 and 3.5.3.2. Essentially, the flows of the pages differ depending on the experimental groups.

3.5.3.1 Control Group UI Flow for Elapsed Authentication Time

The client authenticates the user via the SCRAM-SHA authentication method as follows (see Figure 3-13). Note that at this point, the authenticating server must be running, otherwise the SCRAM-SHA authentication would fail.

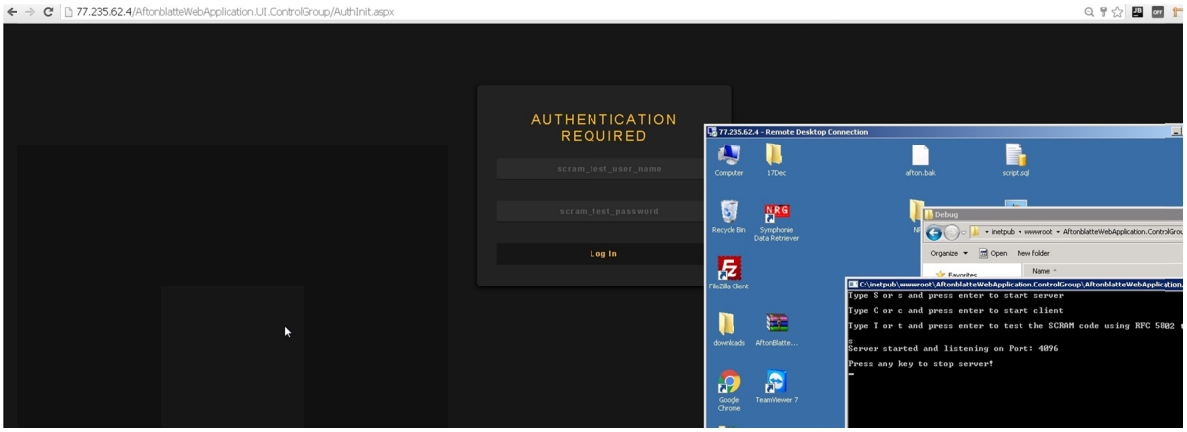


Figure 3-13: SCRAM - SHA server is running and user is expected to login to the web application – Control Group

After supplying the correct username and password, the user is authenticated to the administrator panel as follows:

Success Scenario (see Figure 3-14)

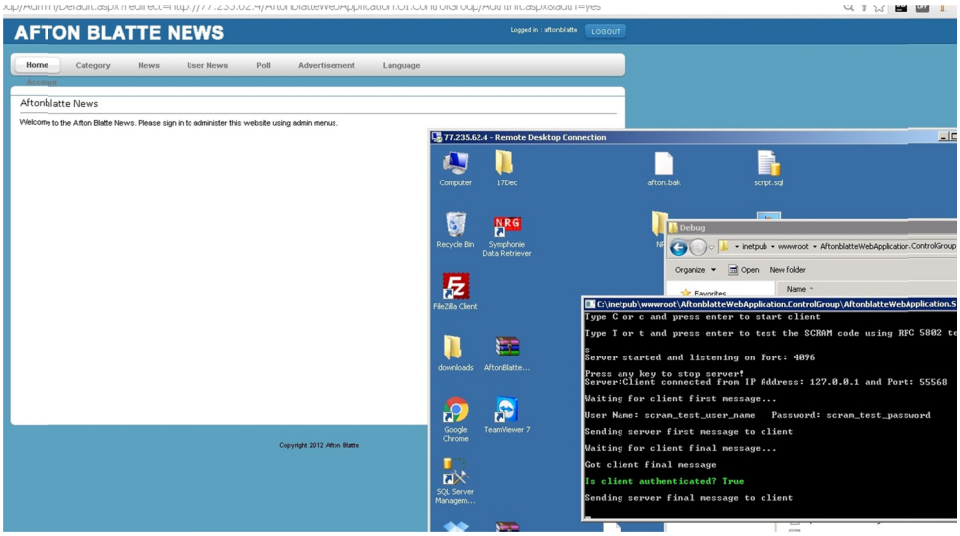


Figure 3-14: Client-SCRAM SHA server authentication success – Control Group

Failure Scenario (see Figure 3-15)

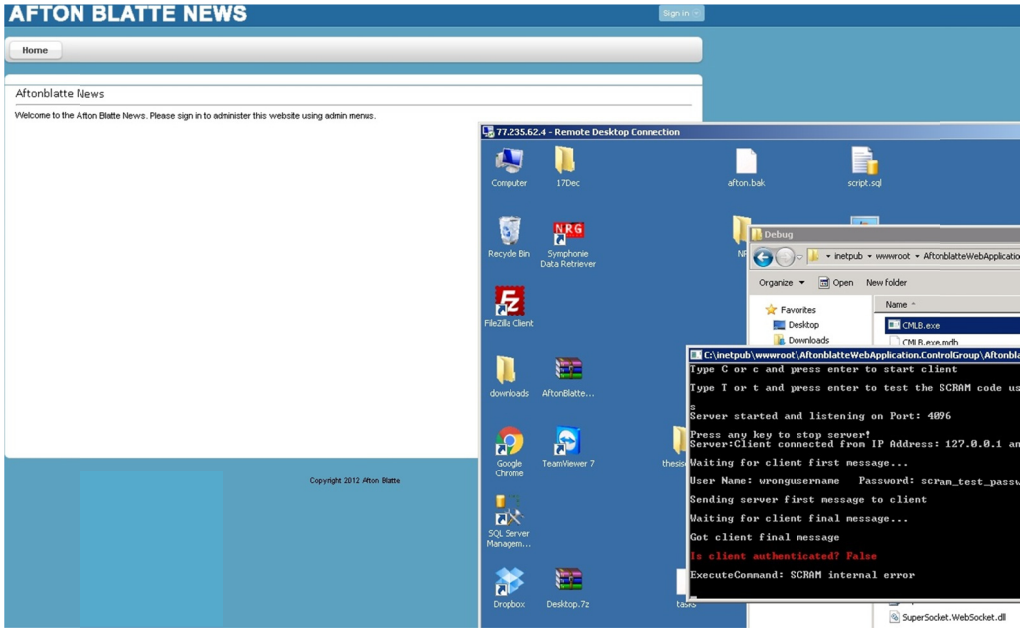


Figure 3-15: Client-SCRAM SHA server authentication failure – Control Group

For this control experiment, the successful authentications are important for statistical analysis and the authentication depends on the elapsed time calculated by the SCRAM authentication and the supplied username and password by the user. The total task time and authentication success and failures are measured, but the total task time is not evaluated since it depends only on the user's behavior and the control experiment for elapsed authentication is independent of user behavior analysis.

3.5.3.2 *Experiment Group UI Flow for Elapsed Authentication Time*

Unlike in the Control Group, in the experiment group UI flow, the user does not see a secondary login until it is triggered by an anomaly being detected in the user's behavior. Instead, the first thing he/she sees is the managenews.aspx in admin panel. The user is expected to add News at this point and then to edit the same news (see Figure 3-16).

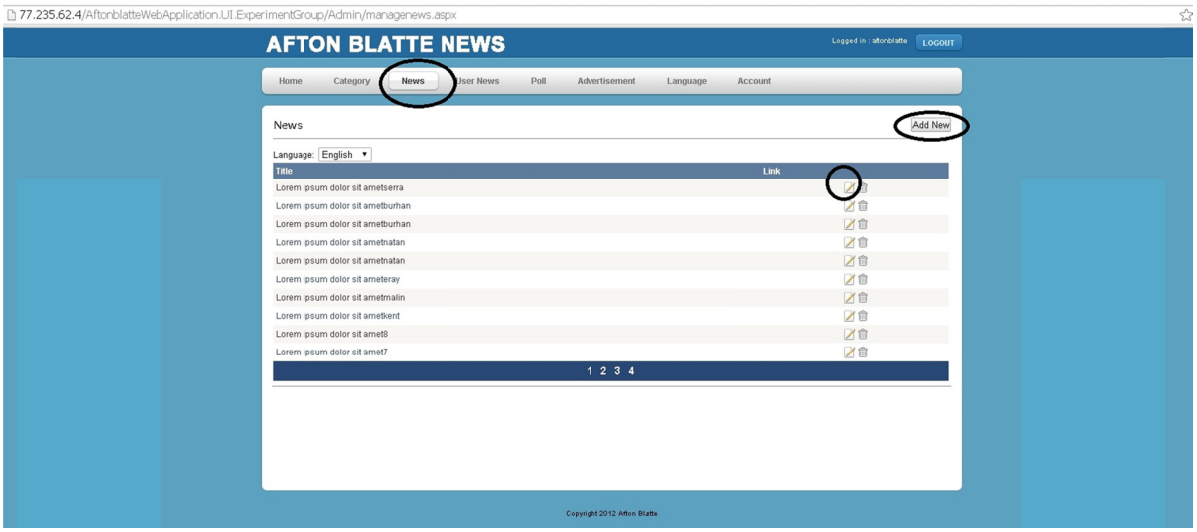


Figure 3-16: User adding and editing news in Experiment Group

In Figure 3-17, red rectangles indicate elements that the behavior template allows the user to interact with *without* triggering secondary authentication, whereas other controls on the website represent controls that users should **not** interact with. If a user engages one those controls, this triggers the system to display a SCRAM-SHA login screen. Note that the user is expected to stay in the managenews.aspx page until he/she saves the news.

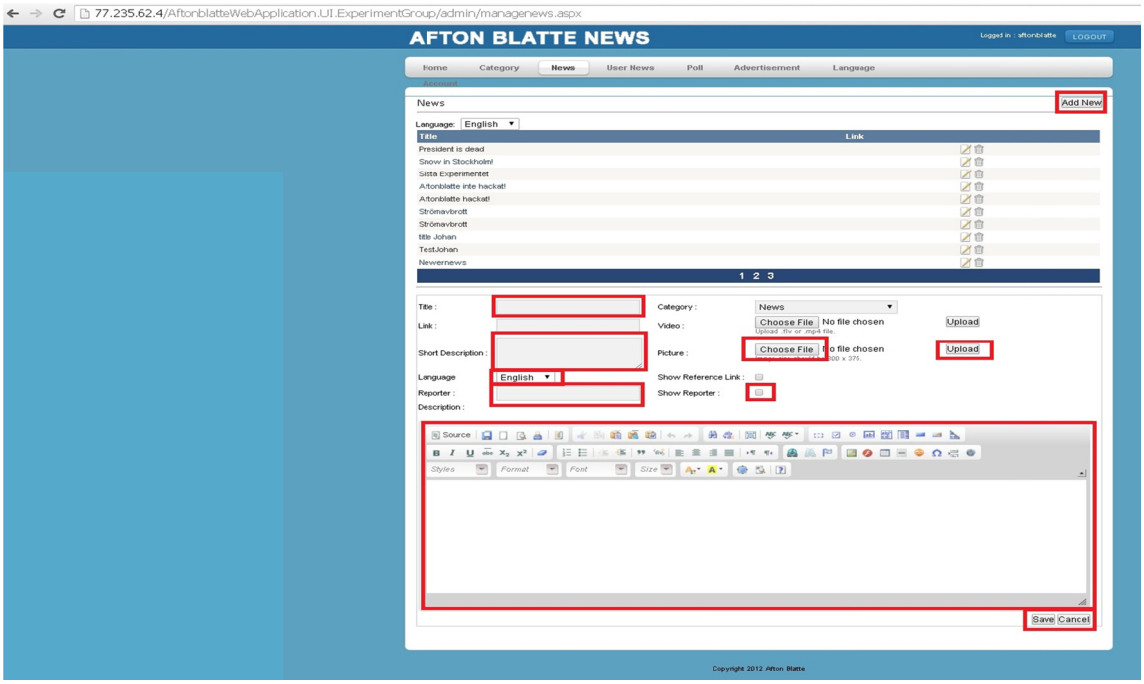


Figure 3-17: Allowed and Prohibited areas in Experiment Group

Each action performed in managenews.aspx page is recorded and checked against the reference (expected) behavior template.

Below is the behavior template file content as used in this experiment. The *name* attribute of the *control* tags represents the controls that user is expected to interact with. The *id* tag is a unique identifier of the control and is only specified when there are multiple controls with the same name on the page. The *btnEdit* controller is an example of such a control since there is more than one edit button. When there are multiple instances with the same name, then a specific such controller can be located only via their id.

```
<?xml version="1.0" encoding="utf-8"?>
<behavior id="managenews.aspx">
 <control name="btnAddNew">
  <id></id>
  <description>
   button to add news
  </description>
 </control>
 <control name="btnEdit">
  <id>ctl02</id>
  <description>
   button to edit news
  </description>
 </control>
 <control name="btnSignout">
  <id></id>
  <description>
   button to sign out
  </description>
 </control>
 <control name="txtTitle">
  <id></id>
  <description>
   Title of the news
  </description>
 </control>
 <control name="txtShortDescription">
  <value>some value</value>
  <id></id>
  <description>
   A short description of the news
  </description>
 </control>
 <control name="drpLanguage">
  <value>English</value>
  <id></id>
  <description>
   Language of the news
  </description>
 </control>
 <control name="txtReporter">
  <value>some value</value>
  <id></id>
  <description>
   Reporter name
  </description>
 </control>
 <control name="fuImage">
  <value>some value</value>
  <id></id>
  <description>
   Image browse
  </description>
 </control>
 <control name="btnUpload">
  <value>some value</value>
  <id></id>
  <description>
   Image Upload
  </description>
```

```
</control>
<control name="chkShowReporter">
  <value>some value</value>
  <id></id>
  <description>
    Show browser
  </description>
</control>
<control name="txtDescription">
  <value>some value</value>
  <id></id>
  <description>
    News Content
  </description>
</control>
<control name="btnSave">
  <value>some value</value>
  <id></id>
  <description>
    button to save news
  </description>
</control>
<control name="btnCancel">
  <value>some value</value>
  <id></id>
  <description>
    button to cancel to add news
  </description>
</control>
      </behavior>
```

If the user behavior is unexpected, in other words, if the user behavior does not fit the behavior user template, then the user is flagged as a potentially illegitimate user. If so, then SCRAM-SHA authentication kicks in as shown in Figure 3-18.

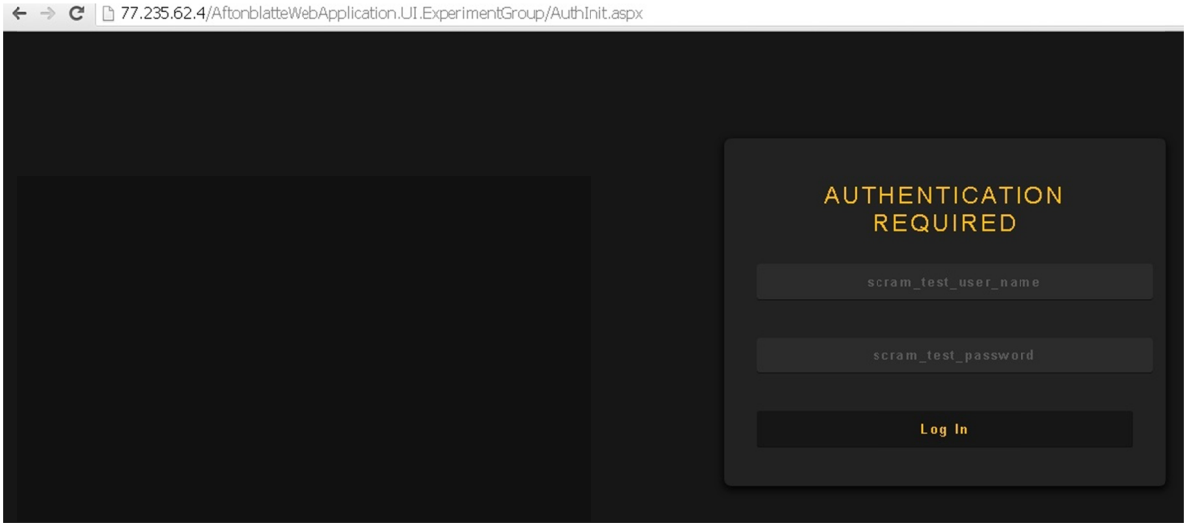


Figure 3-18: SCRAM –SHA authentication kicks in

To retain the session, the user must provide correct credentials. If the user provides correct credentials, then SCRAM SHA evaluates the user as being a legitimate user and the user can continue his/her session as shown in Figure 3-19.

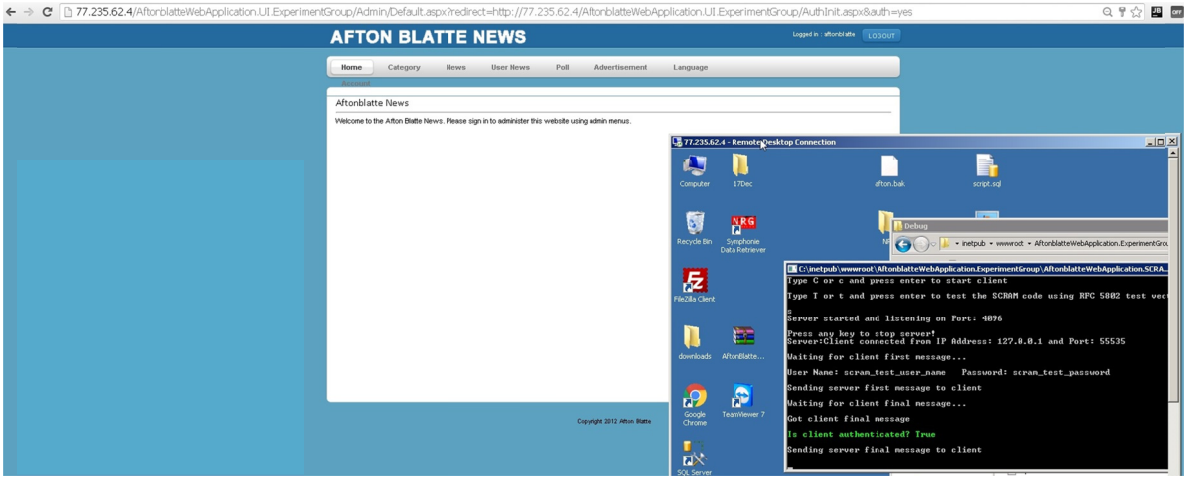


Figure 3-19: User provided correct credentials and server lets user continue

If the user provides the incorrect credentials, then SCRAM SHA evaluates the user as being an illegitimate user. Figure 3-20 shows the case when the user is kicked out of the system and hence needs to login again if they want to continue to access the system.

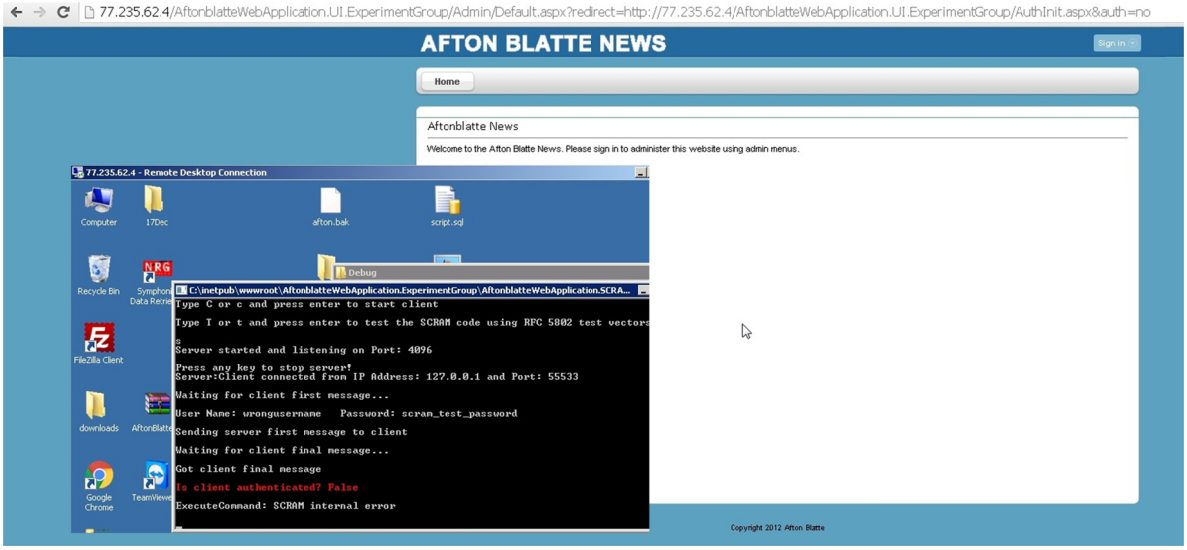


Figure 3-20: User provided wrong credentials and server kicks the user

3.6 Planned Experimental Design

After doing power analysis for elapsed authentication time experiments, the total number of users needed for the control group and the experiment group was calculated as sixty-four users (See Section 3.8.1 “Sample Size for Elapsed Authentication Time Experiments”). On the other hand, the FAR & FRR evaluation experiments need eighty-one users each (See Section 3.8.2.1 “Sample Size for Illegitimate User Experiments” and Section 3.8.2.2 “Sample Size for Legitimate User Experiments”).

Due to these large sample sizes and my time limitation for conducting the experiments, I needed to approach to experiments in a different way as described in the following paragraphs.

I did the experiments with forty-one users in total. The rationale for picking this number is that the experimental groups for elapsed authentication time and legitimate and illegitimate user

experiments can be overlapped and data collection can be done simultaneously for all experiment groups since they use the same the website address for their tests.

The planned experimental design is described in the following ordered list. The reason for this order is that if the same user conducts similar experiments for all groups, then the user must first start as *illegitimate* user. If the user starts with another experiment, then it would be not possible to measure FAR since I give information about the experiment setup in the other experiments.

1. For FAR analysis, the user first does the illegitimate user experiments. This experiment is conducting using the behavior extension with SCRAM as the fallback authentication method. Prior to this experiment, the user is given the task (See Section 3.6.1 "User Task"), but no instruction is given about the website.
2. Next, the same user does the elapsed authentication time experiment for the Control Group. Prior to this experiment, the user is instructed about the given task and then is expected to perform the task (Section 3.6.1 "User Task").
3. Next, the same user does the elapsed authentication time experiment for the Experiment Group. This experiment also covers the legitimate user experiment for FRR analysis, as the experiment setup for elapsed authentication time and legitimate user experiments for FRR analysis are exactly the same applications. In this test, the user already knows about the given task and is expected to complete the task (See Section 3.6.1 "User Task").
4. Experiments with the control groups for FRR and FAR analysis are not done since they represent the perfect group, i.e., they are all legitimate and behave as expected.

In a preliminary analysis of the experiment results, I saw that when the SCRAM authentication server is running continuously, the elapsed authentication time obtained from SCRAM authentication is consistently smaller for successive authentications. I investigated this phenomenon, but was unable to determine a root cause for this. However, running this server continuously would be pointless since the experiments are planned to be done one user at a time. As a result, for each user's experiment I started and stopped the server so as to have reliable results. Due to this phenomenon, I conducted the experiments in such a way that the SCRAM authentication server was running for only one user and the participants in the different groups were not expected to run the experiments simultaneously.

During the experiments, the SCRAM authentication server sometimes generated large values in unevenly spaced time frames. These large values are included in the analysis since they might be important statistically.

During each experiment, I first start the SCRAM server to capture the user's authentication status in all conducted experiments. Then, the user starts to do the given task. Finally, I stop the SCRAM server and collect the log files.

The test environment was primarily running on my development environment. During my tests, I saw that the web application integrated with SCRAM and the behavior extension showed some different results. Since the test groups are dispersed around the globe, I needed to transfer the environment to a dedicated host so that I can have reliable results.

3.6.1 User Task

The task for all users in the experiments is to add a piece of news (a new news item) and add a picture by editing that news item. The details of the task are given as a scenario to control and experiment groups in elapsed authentication time experiments. Both groups are given basic training to be able to handle the experiment without any difficulties. The details of the task are also given to

the legitimate user experiment group. The details are not given to the illegitimate user experiment group. These task details are as follows:

TASK

Preliminary information: If you end up with scram authentication at any point, use the credentials and start over.

You need to press News from above in the admin screen

Add the news.

Fill title, “Lorem ipsum dolor sit amet ***yourname***”

Fill short description, “Lorem ipsum dolor sit amet, consectetur adipiscing elit.”

Fill reporter description, “***yourname***” and be sure that the reporter will be visible.

Write down the following content, you don’t have to write down all of it.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut facilisis consequat est eu venenatis. Nullam sit amet congue magna.

You might need to save the news. You need to edit the news that you created and add an image in it. Don’t forget to save it and logout.

3.7 Legitimate and Illegitimate User Experiment Groups

Essentially, legitimate and illegitimate user experiments can be regarded as a spinoff from the Experimental Group for Elapsed Authentication Time experiments. The flow sequence diagrams of these experiments are shown in Figure 3-6, Figure 3-8, and Figure 3-21.

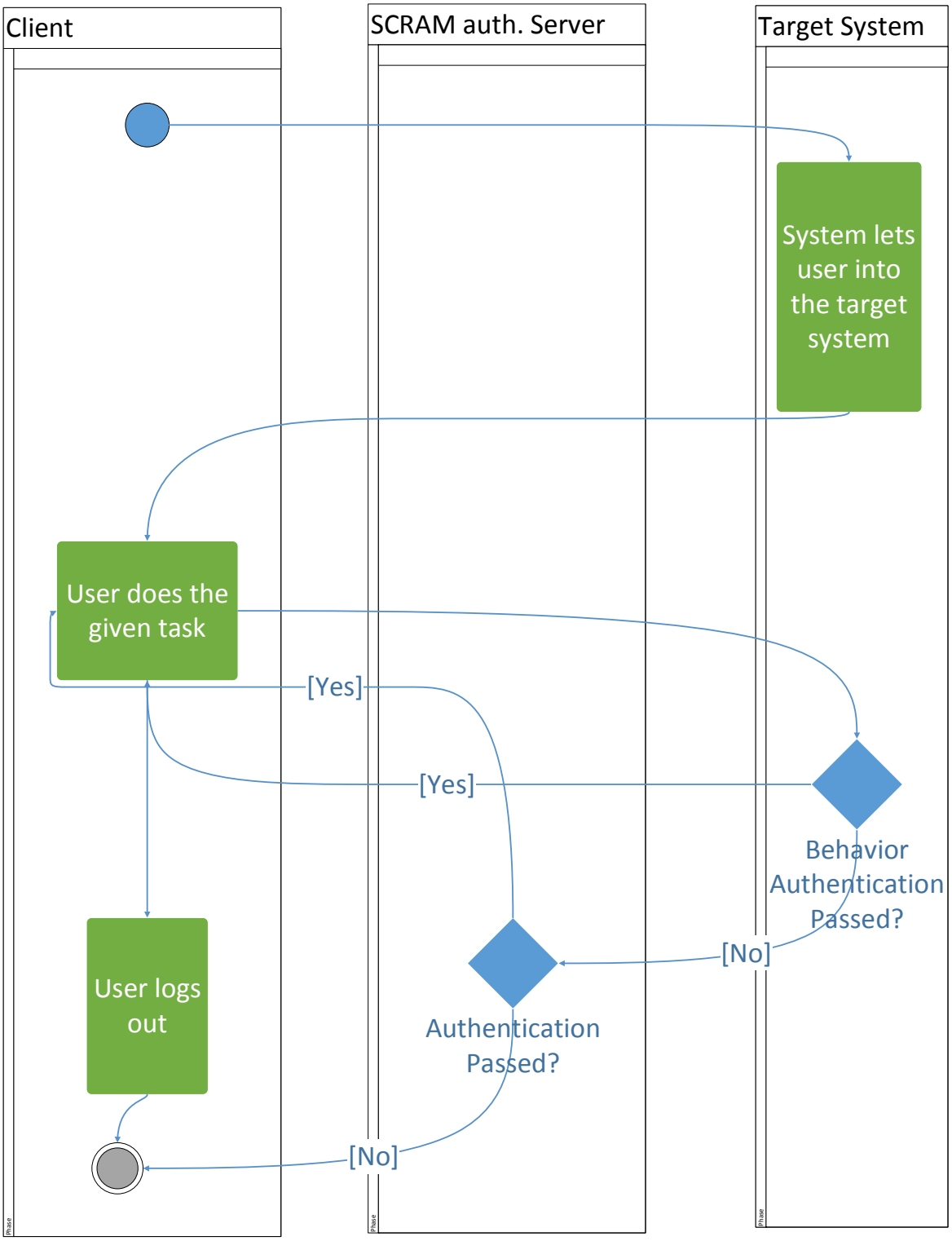


Figure 3-21: **The sequence flow of legitimate and illegitimate group experiments**

In these experiments, there are four groups, two of which represents the Control Groups and they represent the ideal FAR and FRR rates of 0%. This value of 0% indicates that these Control Groups are hypothetical and their results represent a perfect implementation. The other groups are the Legitimate Experiment Group and Illegitimate Experiment Group.

The Legitimate User Experiment Group is a trained group, who are expected to use the application without causing any violation regarding their behavior. This is accomplished by giving them the task explained in the Section 3.6. The users in this group are deemed to be legitimate users and the group is used to measure the FRR of the behavior extension authentication method.

The Illegitimate Experiment Group is an untrained group, who are *not* expected to use the application *without* causing a violation regarding their behavior. This is accomplished by giving them the task, but they are not trained for the application. Instead they are informed roughly how the application works and what a user can do in that admin panel. As they are not informed about the security mechanisms in the web application, the users in this group are deemed to be illegitimate users and the group is used to measure a False Acceptance Rate of the behavior extension authentication method.

3.8 Data Collection

Data collection is done by logging directly from the experimental web application. In all experiments, the logging is done on the server side. In order to cover all of experiment needs regarding logging:

- The server time spent on the authentication method is recorded and logged,
- The user action in the browser is logged, and
- The time taken for fulfilling the task is logged.

No prior technical knowledge is necessary for subjects to participate in the experiment. Out of four experiments conducted, users from three groups are individually trained before the experiments. This training was expected to enable the subjects to grasp their respective scenario faster. The other group, i.e., the illegitimate users, is untrained but has been verbally given a rough description about how the application works.

There are several ethical issues that should be considered when planning any type of data collection involving human participants as data collection can cause participants discomfort, potential harm, and uses some of their time. In this thesis project, the data gathering is done with the consent of the participants in the least intrusive and least costly data collection method possible as discussed in Section 3.10 “Planned Data Analysis”. These subjects were all informed that their data would be used in this thesis project; however, it, the data, would not be individually identifiable. No other ethical issues have been identified in this study.

When it comes to preparing for an experiment, conducting the experiment, and collecting data, it is important to underline usability in the experiment. According to Tom Tullis and Bill Albert there are three aspects of usability: “the extent to which a product can be used by specified users to achieve specified goals with *effectiveness, efficiency*, and *satisfaction* in a specified context of use” [77]. To assess usability in this experiment, there should be observable differences in the authentication time between the control group and the elapsed authentication time experimental group, as this is the core part of the experiments. To provide effectiveness of the behavior extension, FAR and FRR are evaluated in the legitimate and illegitimate user experiments. Finally, to assess efficiency, the authentication time for elapsed authentication time experiments are measured to evaluate how efficient the behavior extension is. Given appropriate effectiveness and efficiency of the proposed method, then the conditions of satisfaction are expected to be fulfilled.

3.8.1 Sample Size for Elapsed Authentication Time Experiments

In order to define the sample size, I planned to use two tailed Student's t-test. As this test has one measurement variable and one nominal variable and the nominal variable has two values, the results of Student's t-test indicate whether the means of the measurement variable differs *between* the two groups. Student's t-test assumes that the observations *within* each group are normally distributed.

In this thesis project, I decided that I would like to be able to detect a difference of 118 milliseconds in the mean post authentication times between the two test groups. I used G*Power software for statistical power analysis (see Section 3.11). In order to learn the approximate mean of the groups, I performed a preliminary experiment in which I measured the time spent for authentication by logging the successful logins via the SCRAM method (i.e., the method to be used by the control group). The mean times were 143.8079 milliseconds for group 1 (control group) and 25 milliseconds for group 2 (test group), and with a standard deviation of 130.7 milliseconds for each group.

Using G*Power, I entered the above means and standard deviation and chose to use $P < 0.05$ probability of detecting a difference this large, corresponding to a 95% confidence (1-beta) with a degrees of freedom represented as potentially independent variables - 2 [58]. After entering these numbers in G*Power, I learned that I need to have a sample size for each group of 33 people. The G*Power input and output are shown in Table 3-4.

Table 3-4: Student's T-test protocol that was planned to be used

t tests – Means: Difference between two independent means (two groups)			
Analysis:	A priori: Compute required sample size		
Input:	Tail(s)	=	Two
	Effect size d	=	0.9086203
	α err prob	=	0.05
	Power (1−β err prob)	=	0.95
	Allocation ratio N2/N1	=	1
Output:	Noncentrality parameter δ	=	3.6908331
	Critical t	=	1.9977297
	Df	=	64
	Sample size group 1	=	33
	Sample size group 2	=	33
	Total sample size	=	66
	Actual power	=	0.9530269

The total sample size versus power plot is shown in Figure 3-22. We can see that a sample size of 33 corresponds to a statistical power of 0.95.

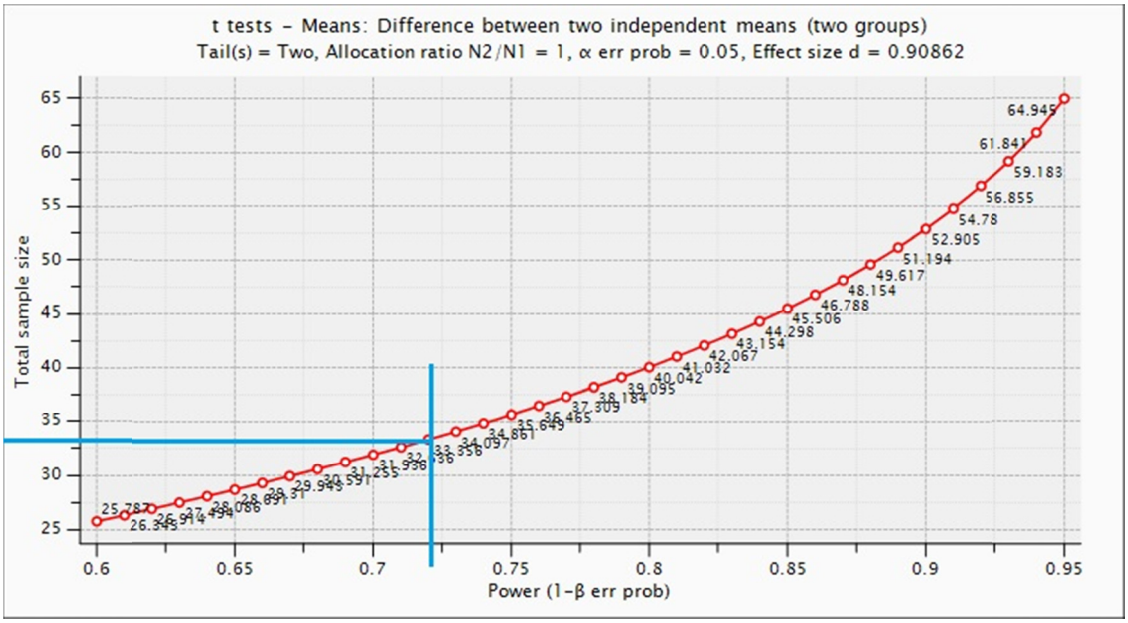


Figure 3-22: **Total sample size versus power plot**

G*Power's central and non-central distributions are shown in Figure 3-23.

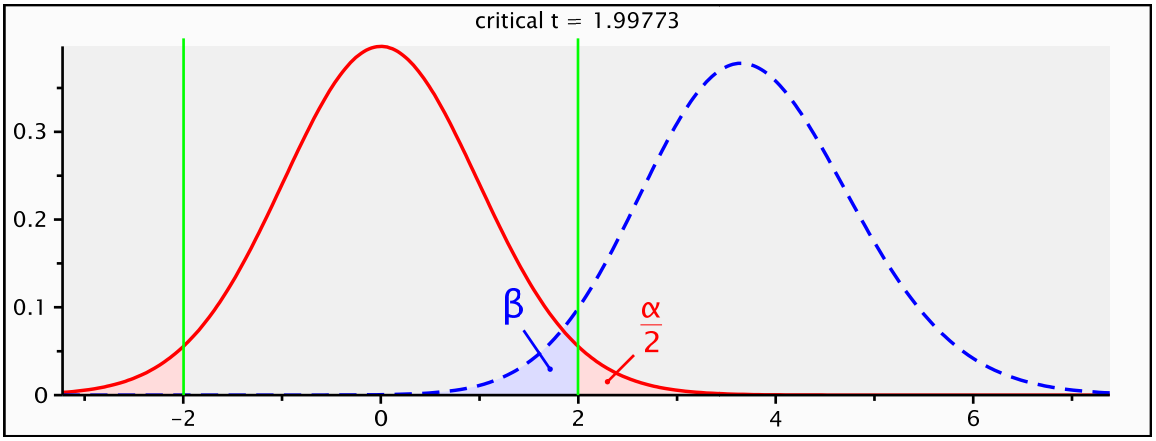


Figure 3-23: **G*Power central and non-central distribution view**

There are several strategies to increase statistical power [58]. Increasing the number of samples would increase the cost of the experiment. To retain the same statistical power, I needed to find another way to maintain statistical power with fewer samples. To increase the statistical power, I needed to increase the size of hypothesized effect size. As SSO times are invariant between groups, to increase the hypothesized effect size, I removed single sign on which was planned as the first authentication method [58]. In addition to this, I also collected additional data for statistical control purposes which should also have an influence on effect size [58].

Since I hypothesized that the standard deviation of both the experimental and control populations was 130.76 ms and I would like to detect a change (in average level between the two *populations*) of 118.80 ms (143.8079 - 25), it appeared that I would need quite a few observations to

be able to reject the hypothesis that the two *populations* have the same mean when the actual difference in the *populations* is about this size given this (relatively large) standard deviation.

However, the variability within the control and the experimental populations are unlikely both to be of this size. The primary reason why this is unlikely is that another preliminary analysis for the control group yielded different results and different means.

Due to these ambiguities, and the expectation that the measurements will not fit a normal distribution, I abandon Student T-test and instead chose to use non-parametric statistical methods, as these methods are designed for analyzing non-normal data using descriptive statistics. However, in order to do this I need to specify my sample size.

Since the distribution is uncertain, I used min asymptotic relative efficiency (ARE) as the distribution [78]. After entering the values in G*Power that were acquired via a preliminary test, I learned that to perform a Wilcoxon-Mann-Whitney test with ARE that I need to have a sample size for each group of 32 people. This estimate is based on 64 independent pieces of information, which determines the degrees of freedom (Df). G*Power specifies the degrees of freedom as 53.29. The G*Power input and output are shown in Table 3-5.

Table 3-5: G*Power protocol that is used

t tests – Means: Wilcoxon–Mann–Whitney test (two groups)			
Options:	ARE method		
Analysis:	A priori: Compute required sample size		
Input:	Tail(s)	=	One
	Parent distribution	=	min ARE
	Effect size d	=	0.9076923
	α err prob	=	0.05
	Power (1–β err prob)	=	0.95
	Allocation ratio N2/N1	=	1
Output:	Noncentrality parameter δ	=	3.3748581
	Critical t	=	1.6739508
	Df	=	53.2960000
	Sample size group 1	=	32
	Sample size group 2	=	32
	Total sample size	=	64
	Actual power	=	0.9541689

The total sample size versus power plot is shown in Figure 3-24. We can see that a sample size of 32 corresponds to a statistical power of 0.77. G*Power's central and non-central distributions are shown in Figure 3-25.

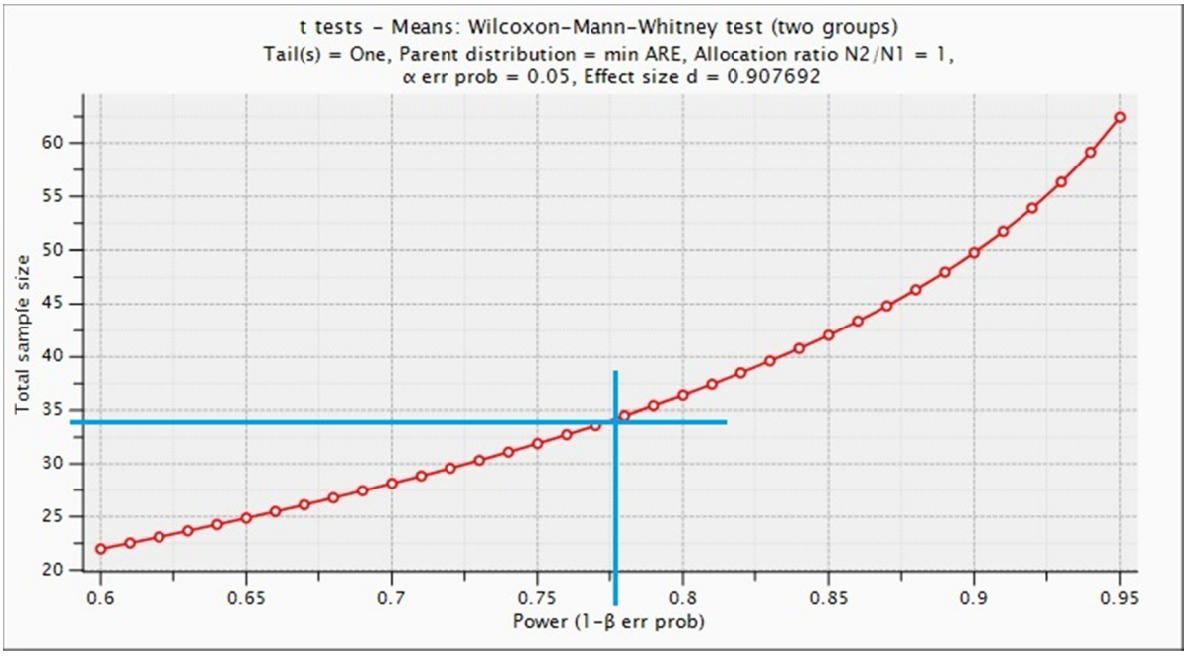


Figure 3-24: **Total sample size versus power plot**

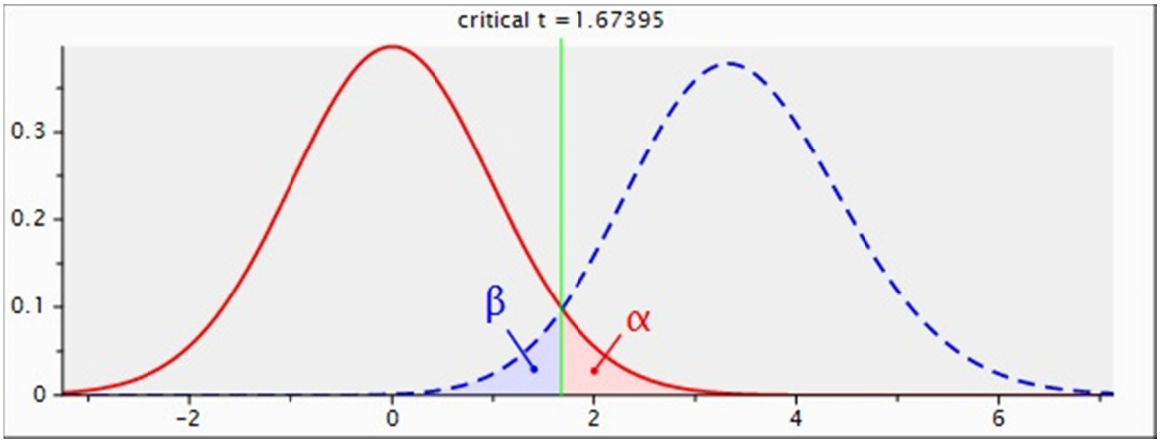


Figure 3-25: **G*Power central and non-central distribution view**

3.8.2 False Acceptance Rate (FAR) and False Rejection Rate (FRR)

The legitimate and illegitimate user experiments rely on dichotomous variables and in order to define the sample size, I assumed the control groups for these experiments would both have a FAR and FRR of 0%. Since G*Power does not let me input 0% for these rates, I use 0.01% for FAR and FRR rates for these control group experiments.

3.8.2.1 Sample Size for Illegitimate User Experiments

Due to lack of prior information, for the significance level (alpha), I choose 5% and for power (1 – beta), I choose 95%.

To falsify the null hypothesis, at most 5% of the illegitimate users shall be rejected by the system, i.e., the effect size is 0.05. In other words, the effect size for FAR and FRR evaluations are 0.05 and if the effect size is 0, then this would confirm the control group's perfectness. After my experiments, I expect to see 5% of the illegitimate users shall be accepted by the system, whereas 95% of the illegitimate users shall be rejected.

The G*Power protocol and total sample size versus power plot for this setup are shown in Table 3-6 and Figure 3-26. G*Power's central and non-central distributions are shown in Figure 3-27.

Table 3-6: G*Power protocol that is used for determining sample size for illegitimate users

χ² tests – Goodness-of-fit tests: Contingency tables			
Analysis:	A priori: Compute required sample size		
Input:	Effect size w	=	0.4020151
	α err prob	=	0.05
	Power (1–β err prob)	=	0.95
	Df	=	1
Output:	Noncentrality parameter λ	=	13.0909074
	Critical χ²	=	3.8414588
	Total sample size	=	81
Actual power = 0.9513586			

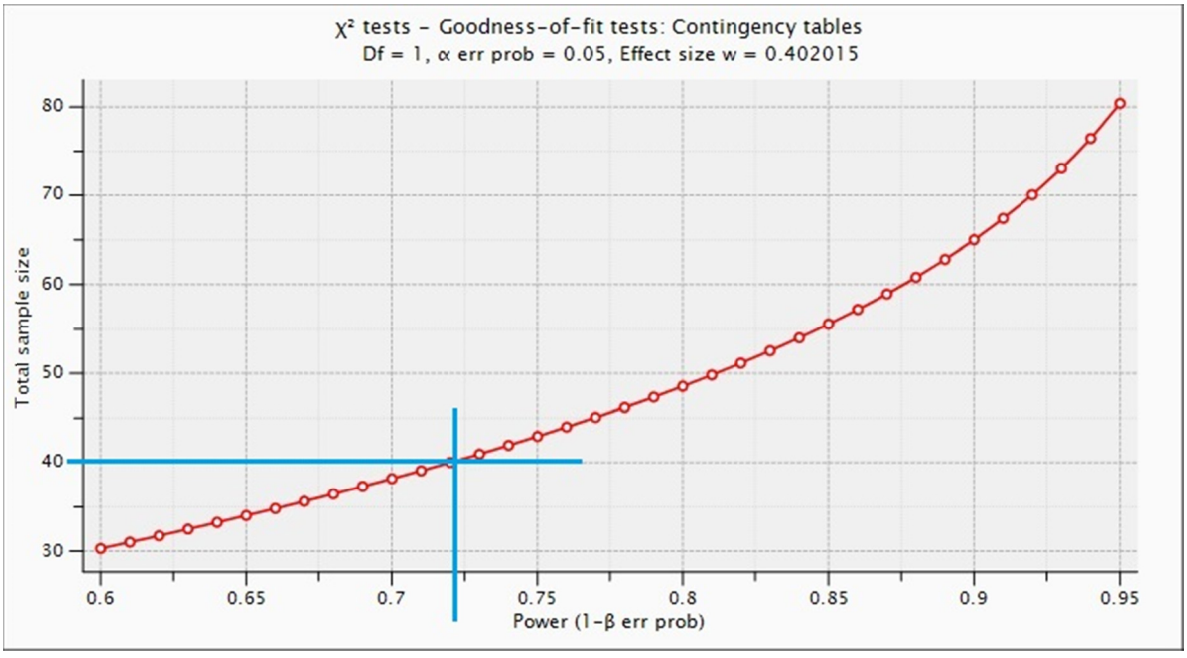


Figure 3-26: Total sample size versus power plot

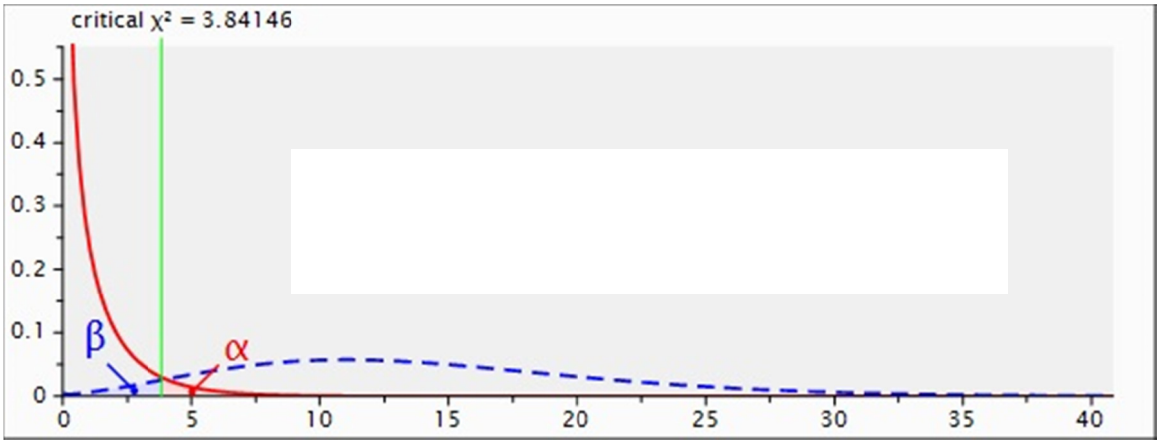


Figure 3-27: **G*Power central and non-central distribution view**

3.8.2.2 Sample Size for Legitimate User Experiments

Due to lack of prior information, for the significance level (alpha), I again choose 5% and for power (1 – beta), I choose 95%.

To falsify the null hypothesis, at most 5% of the legitimate users shall be rejected by the system, i.e. the effect size is 0.05.

The G*Power protocol and total sample size versus power plot for this setup is as in Table 3-7 and Figure 3-28. G*Power's central and non-central distributions are shown in Figure 3-27.

Table 3-7: **G*Power protocol that is used for determining sample size for legitimate users**

χ² tests – Goodness-of-fit tests: Contingency tables			
Analysis:	A priori: Compute required sample size		
Input:	Effect size w	=	0.4020151
	α err prob	=	0.05
	Power (1-β err prob)	=	0.95
	Df	=	1
Output:	Noncentrality parameter λ	=	13.0909074
	Critical χ²	=	3.8414588
	Total sample size	=	81
Actual power = 0.9513586			

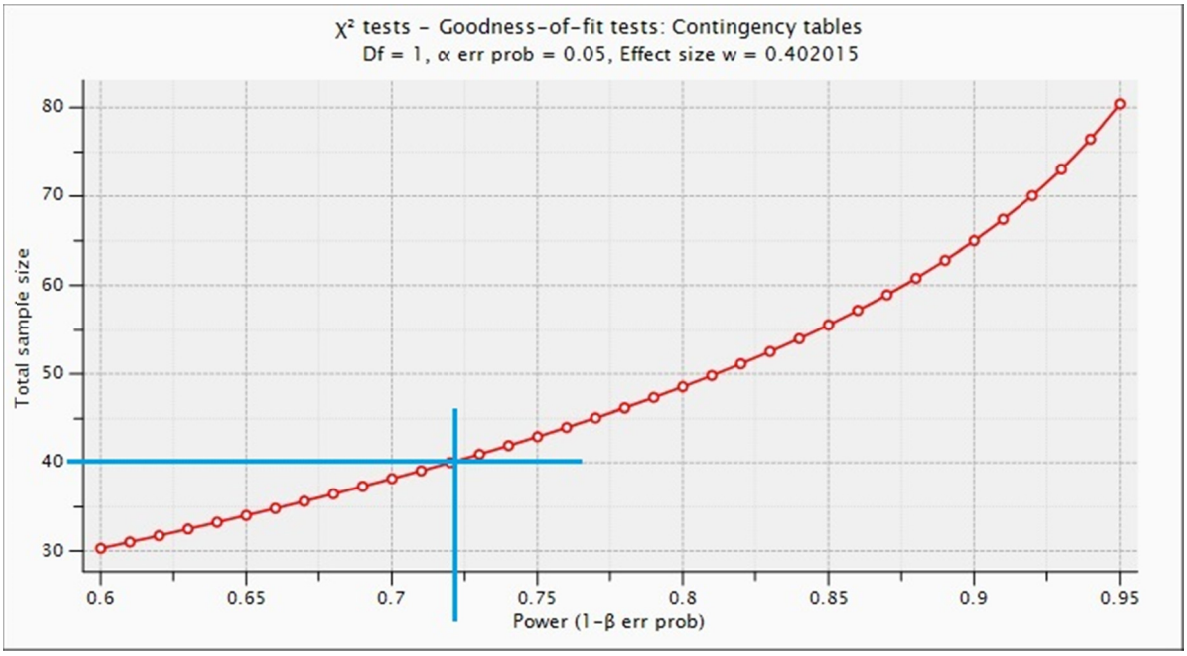


Figure 3-28: Total sample size versus power plot

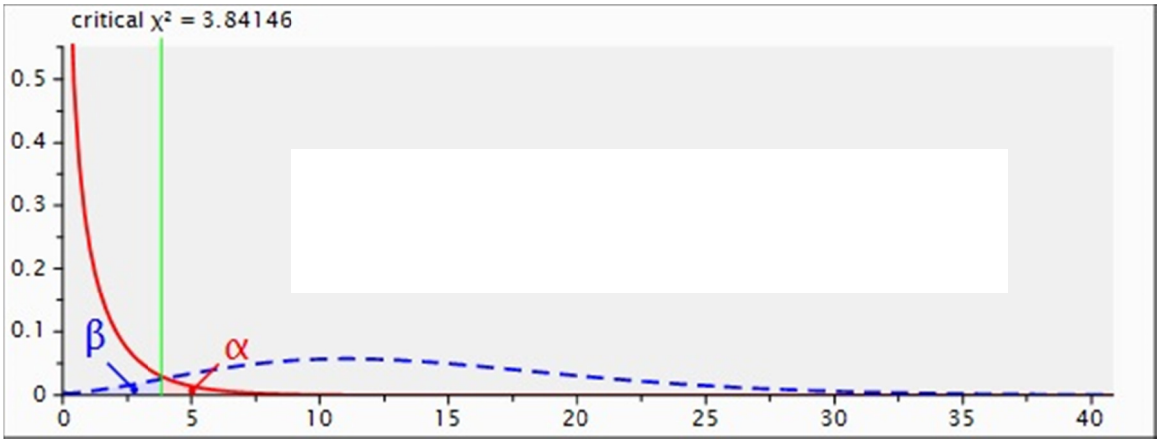


Figure 3-29: G*Power central and non-central distribution view

3.8.3 Target Population

My test subjects/sample groups should be representative of my target populations. In this thesis, determining the target population has not been specified based upon existing studies, as there is a lack of statistical analysis in this field (See 5.2.1 “Lack of previous work”).

Although the test application is based on an existing website, the target population of this study does not consist of the audience of this website, e.g., aftonblatte.se. Instead, the target population would be anyone using SSO solutions and expected to have a basic knowledge about form authentication mechanism, e.g. those using username and password to login a system.

My sample size should be representative of the target population to draw a conclusion using statistical analysis. The confidence interval gives us an interval that with a given level of confidence contains the true value of the parameter being estimated in the target population [59].

It is usually the case that the researcher wants to make predictions or estimate parameter values of a population. The researcher gets the information from a sample (often only a tiny fraction of the population) to make those estimates and/or predictions. A benefit of doing a statistical analysis (as opposed to a mathematical model) is that not only the researcher derive estimates/predictions, but also he/she gets some statement of how close the estimates/predictions are to the actual unknown population value.

Samples in these experiments are Tacton Systems AB employees and my friends in Facebook. The characteristics of the samples are dependent on English comprehension capability, experience on using computers and internet. The volunteers in the experiment are assured that they should feel free to discontinue their participation at any point in the experiment. In the experiments, I observed that there is only 1 volunteer that failed to finish the experiment.

The aim of this thesis is to derive conclusions from the results of the experiments; these conclusions would be applicable to the entire target population. Given the fact that the actual target population is so large, I estimate that a target population within my company and a target population which comprises of my friends might be feasible example users.

3.9 Assessing reliability and validity of the data collected

This section evaluates the reliability and validity of the data collected. In this section, I investigate the concepts of true and error scores, the terms of reliability and validity and their measurements and how they are applicable to the experiments I conduct.

3.9.1 True and Error Scores

True and Error scores are fundamentals of statistics and they are important concepts to consider during experimentation. The experiments done in this thesis project involve estimating both true and error quantities and maximizing the true portion. while minimizing the error portion [79].

True and error scores are mathematically defined as follows:

$X = T + E$ where X is all the observed measurements, T is the True score and E is the Error score [79].

Random error is a type of error where the error is due to chance [79]. Random errors have no pattern and they are assumed to cancel themselves out during repeated measurements [79].

During the experiments, some attributes in the population may be more likely to be included in the sample than others; or errors of observation or measurement may result in a systematic accumulation of inaccuracies in a single direction. Sampling, observation, or measurement errors are known as bias [80]. Bias can enter studies in two primary ways: (1) during the selection & retention of the objects of study or (2) in the way information is collected about the objects [79]. Bias leads to systematic errors, rather than random errors. It is imperative that the sampling procedure is unbiased (or can be corrected for bias). Unbiased means that each member of the population has an equal chance of inclusion. If the sampling procedure is biased then the assumptions that the subsequent statistical analysis rests on are violated and the results are incorrect. If the sample is biased, meaning that in some systematic way it is not representative of the target population, then conclusions drawn from the study's sample may not apply to the target population [79].

3.9.2 Reliability

Reliability indicates how consistent the scores of an experiment are and whether the measurements are repeatable [79]. Unreliability of an experiment is a problem which affects the results and produces conclusions which are not trustworthy [81].

There are several ways of measuring reliability which can be useful in particular contexts [79]. These are multiple-occasions reliability, multiple-forms reliability, and internal consistency reliability [81]. Each of these will be described in the following paragraphs.

Multiple-occasions reliability, also known as the test-retest method, is a method for measuring reliability. In this method, the experiment is only reliable if the same measurement has the same values at a later time. The measurement of repeatability of an experiment may have costs for the experimental groups and also for those who assesses the data. For example, a test subject might not be willing to participate in successive experiments for to several reasons, for instance, the experiment may be long and the test subject might not want to lose time repeating same experiments again and again. Similarly, those who analyze the collected data might have difficulties deriving a conclusion from the results if the amount of collected data requires too much time to analyze.

Another concern is that the time frame for the repetition of the experiment can lead to different results. If the time gap between test and retest is too short, then the test subject can recall his/her conduct and try to improve or worsen his/her conduct by learning between the test and retest time periods. Conversely, if the time gap is too long, the test subject might gain new experiences and recollections that would change their behavior in a systematic way [79].

The second way of measuring reliability is the multiple-forms reliability (also known as parallel-forms reliability). In this method, different versions of the tests are used and administered to the same test subjects. The difference between the results of the various tests are a measure of their reliability [79].

The third way of measuring reliability is internal consistency reliability, which measures how well the components of a test measure the same thing. One way of measuring this is by using split-half methods which require that a test be split into two parts or forms, usually two halves of equal length, which are intended to be parallel [79].

In the elapsed authentication experiments, the same user is participating in both control group and experimental group experiments. Hence, it can be said that multiple-occasions reliability could be applied as the same user is doing the same experiment twice. However, for the legitimate and illegitimate user group experiments, I evaluated the consistency of results from two different tests for the presence or absence of legitimacy in the groups [79]. These latter experiments can be

considered to have multiple-occasion reliability since the same user first acts as an illegitimate user and then acts as a legitimate user, thus having a test-retest approach.

For all of our experiments we cannot apply multiple-forms reliability since it was infeasible to prepare different versions of tests due to the limited duration of this thesis project and the cost of developing different versions of the test was not expected to be justified by the additional data.

Internal consistency reliability was not evaluated as the elapsed authentication experiment is unsuitable to partition in a balanced way. Moreover, these experiments do not have internal consistency reliability since they are measuring different rates, FAR and FRR, and there is no easy way to split the experiments into two forms.

3.9.3 Validity

Validity shows how well a test measures what it is expected to measure. Although there are various controversies about the categorization of validity types, I will consider four types of validity [79]: content validity, face validity, concurrent validity, and predictive validity. Each of these in described further in a paragraph below.

Content validity refers to the correlation with the domain of interest [81]. Since one of the hypotheses of this thesis concerns the user's behavior and secondary authentication methods, the content of the test is relevant to the work. For this reason I am evaluating the post authentication with time and acceptance rate measurements.

Face validity concerns how valid the test seems to the various stakeholders. The stakeholders in my experiments are the test subjects, my examiner, my industrial supervisor, and I. In my experiments, the user groups are aware of what the experiment is intended to measure. However, the face validity of these experiments was not assessed in this thesis. Despite this, by submitting this thesis I am asserting that I find the tests to be valid.

Concurrent validity is a type of validity where the measurement can be used to predict some other behavior or performance that is measured simultaneously [81]. This type of validity is not analyzed in my experimentations.

Finally, predictive validity, another type of validity that measures the ability to infer future events, is not assessed in this experimentation [81].

3.10 Planned Data Analysis

In this section, I evaluate the planned statistical and risk analyses.

3.10.1 Planned Statistical Analysis

In order to avoid possible selection bias, I have retained the outlier values for the Elapsed Authentication Time Control and Experiment Groups.

In the elapsed authentication time experiments, I encountered random high values, which appear to be independent and unrelated to the error of any other score. The root cause of these values might be the SCRAM SHA-1 implementation for the fallback mechanism (see Section 3.5.1). These values might be random errors; however, the measured values are systematically larger. From this, one might infer that the error is not a random error, but rather a systematic error which could be identified and remedied.

Due to the outlier values, I presumed that the distribution is not a normal distribution and I abandoned the use of a Student t-test and chose to do a non-parametric Mann-Whitney U Test

instead. The sample size for the elapsed authentication time experiments was determined by t tests - Means: Wilcoxon-Mann-Whitney test (two groups).

The planned data analysis has the objective of the development of two-sample nonparametric tests which, very generally, seek to determine if two population distributions are identical [80].

Table 3-8: Nonparametric vs. Parametric Tests [80]

Nonparametric Tests	Parametric Equivalents Test
(*a*) Single-Sample Runs Test	(*a_*) No Parametric Equivalent
(*b*) Single-Sample Sign Test	(*b_*) Binomial Test on *p* or *Z* or *t* test of H_0:$\mu = \mu_0$ (provided the population is normal)
(*c*) Single-Sample Wilcoxon Signed Rank	*t* Test of H_0:$\mu = \mu_0$ (provided the test population is normal)

The Mann-Whitney test indicates the null hypothesis as: H0: the population distributions are identical [80]. This hypothesis may be tested against the following alternatives:

	Case I	Case II	Case III
H0:	the population distributions are identical	the population distributions are identical	the population distributions are identical
H1:	population 1 is located to the right of population 2	population 1 is located to the left of population 2	the populations differ in location

The Mann-Whitney test is evaluated as a good nonparametric tests for differences in location (and thus for differences in means or medians) when its asymptotic relative efficiency is compared to that of the parametric t test for differences in means [80]. If the populations are normally distributed, the asymptotic relative efficiency of the Mann-Whitney statistic is at least 95% whereas it is at least 86% in the absence of normality if the population distributions are identical [80].

For continuous and independent distributions, the Mann-Whitney test might perform better than the t test in the absence of the normality. The sample size required for a t test is approximately 95% of that required for the Mann-Whitney test if normality holds whereas Mann-Whitney test requires fewer observations than the t test to achieve comparable α and β risks [80].

Using the standard normal table in Appendix C: Critical values for chi-square distribution, if we find that the result has a p-value greater than 0.05; then, we fail to reject the null hypothesis. In the following figures, the area under the normal curve, expressing the probability that ($0 < x < |a|$), the probability that a value of x lies in the range between 0 and the absolute value of some value a. Suppose a = 0.5. The area ($0 < x < 0.5$) is represented by the shaded area in Figure 3-30 and Figure 3-31. The corresponding area under the normal curve can be seen in the table in Appendix D.

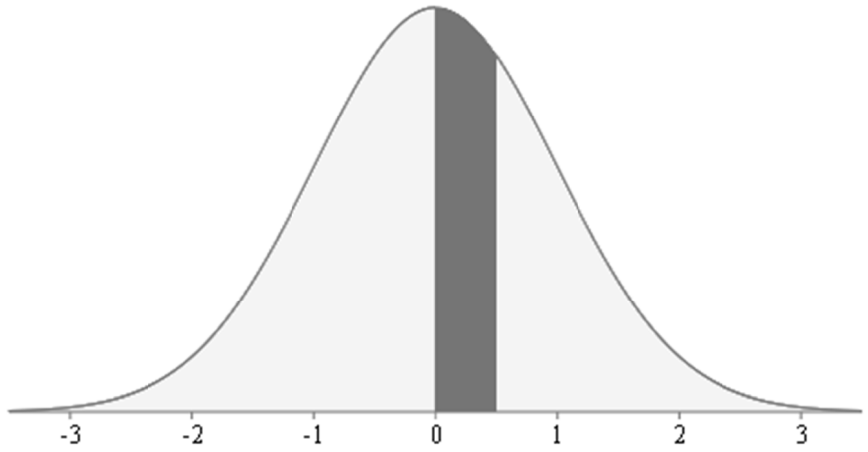


Figure 3-30: Area (0 < x < 0.5) of the standard normal distribution

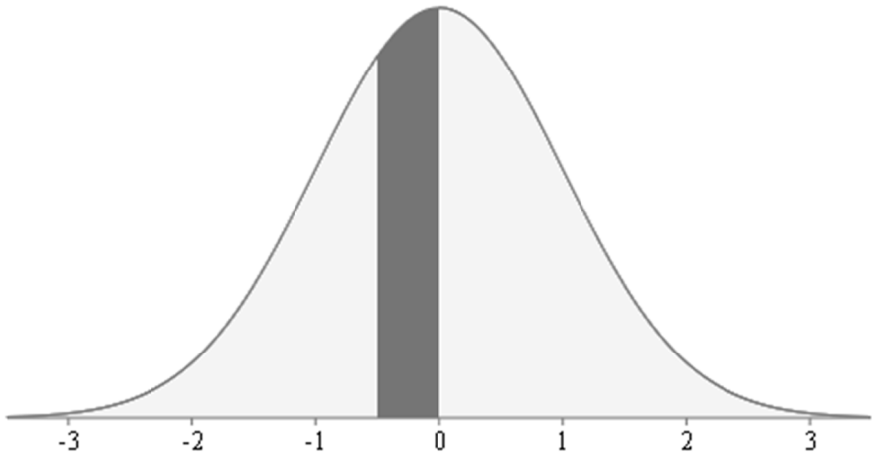


Figure 3-31: Area (−0.5 < x < 0) of the standard normal distribution

The sample size for the illegitimate and legitimate user experiments are determined by *chi-square* (χ^2) goodness-of-fit tests. We use this test because a *chi-square test of goodness of fit* is used to test whether the distribution of a categorical variable follows a specific pattern of proportions or not. This test is calculated using expected values based on hypothesized proportions.

The χ^2 Goodness-of-fit tests are required to have dichotomous data or continuous data that can be grouped into intervals. The χ^2 Goodness-of-fit tends to be less precise for continuous distributions where the grouping into artificial intervals is arbitrary. On the other hand, for the discrete case, the data has natural grouping that is dictated by the specific variable's values.

The test is performed on frequencies. The frequencies of legitimacy are determined for both illegitimate and legitimate user experiments. Within this data, the frequencies are used to calculate False Acceptance and False Rejection rates.

To calculate the goodness of fit, different categories or groups are designated with the subscript i, from 1 to g with the following formula:

$$\chi^2 = \sum_{i=1}^{g} \frac{(O_i - E_i)^2}{E_i}$$

The steps for using this formula are:

1. Calculate the observed and expected values in Table 3-9 and Table 3-10.
2. Square the difference, and divide by the expected value.
3. Do the same for the remaining cells.
4. Add the numbers calculated in steps 1–3.

The degrees of freedom for a chi-square test of goodness of fit is $(g - 1)$. For illegitimate and legitimate user group experiments, the degrees of freedom is 1.

In the illegitimate user group experiments, I plan to categorize the groups in terms of ranks as FAR and Non-FAR and observed FAR frequencies are measured and evaluated with expected FAR which is at most 5%. In Table 3-9, X + Y is the total sample size for the experiment.

Table 3-9: Goodness of Fit setup for FAR analysis

Rank	Observed FAR	Expected FAR
FAR	X	((X+Y) x 5) / 100
Non-FAR	Y	((X+Y) x 0,95) / 100

In the legitimate user group experiments, I plan to categorize the groups in terms of ranks as FRR & Non-FRR and observer FRR frequencies are measured and evaluated with the expected FRR which is at most 5%. In Table 3-10, X + Y is the total sample size for the experiment.

Table 3-10: Goodness of Fit setup for FRR analysis

Rank	Observed FRR	Expected FRR
FRR	X	((X+Y) x 5) / 100
Non-FRR	Y	((X+Y) x 0,95) / 100

For both experiments, if the p-value is less than our alpha level, then we can reject the null hypothesis. The critical value for $\alpha = 0.05$ is 3.841 (see Appendix B). If the value calculated on our data exceeds this critical value, then we should reject the null hypothesis

For the above mentioned ranks, FAR is computed as the ratio of number of false acceptances to total number of attempts and FRR is computed as the ratio of number of false rejections to total number of attempts [82]. This means:

FAR = False Accepts / Number of tests

FRR = False Rejects / Number of tests

Accept Rate = True Accepts / Number of total operations

Accuracy = 1- (No match/True Accepts)

Precision = 1- (False Accepts/True Accepts)

To calculate FAR and FRR, it is important to tabulate the dichotomous data for both experiments and then the rates can be calculate by considering each user's result and evaluating the frequencies of legitimacy.

From start to end, the planned data analysis consists of the following steps:

1. Collecting the logs from the server.
2. Extracting the appropriate measurements from the logs.
3. Tabulating the results of elapsed authentication time experiments.
4. Tabulating the results of legitimate and illegitimate user experiments.
5. Applying Whitney-Mann U statistical test to elapsed authentication experiments.
6. Calculating false acceptance rate for illegitimate experiment group and false rejection rate for legitimate groups.
7. Applying Chi Square Goodness of fit statistical test on Legitimate and Illegitimate Experiment Groups.
8. Check for significance and assess whether to accept or reject null hypothesis for both hypothesis.

3.10.2 Planned Risk Analysis

As risk is a function of vulnerability of a method, it is important to identify the risk during analysis of the experiment.

Generally speaking, risk is the product of likelihood times impact (Risk = Likelihood * Impact) [83].

Because risk is strictly tied to uncertainty, Decision theory should be applied to manage risk as a science, i.e. rationally making choices under uncertainty.

The measure of an IT risk can be determined as a product of threat, vulnerability and asset values [84].

Risk = Threat * Vulnerability * Asset

Organizations have investment and expense plans in different periods, yearly or annually. The plans rely on the fact that the investment shall not exceed the *expected* savings and revenue.

Quantitative risk analysis evaluates the plans and potential loss is calculated by annual loss expectancy (ALE) and estimated annual cost (EAC). It is thus theoretically possible to rank events in order of risk and decisions can be based upon this ranking [85].

On the other hand, qualitative risk analysis methods are used to assess the risk. Unlike quantitative risk analysis, probability data is not required and only estimated potential loss is used.

In Section 3.3.3, various methods were introduced about risk and assessment of risk. One common approach among these methods is formation of a matrix to measure the risk of an asset via threats or vulnerabilities [61].

This thesis uses three separate matrices for risk analysis:

- vulnerability matrix,
- threat matrix and
- control matrix.

The vulnerability matrix contains the associations between the assets and vulnerabilities in the organization. The threat matrix contains the relationships between the vulnerabilities and threats, and the control matrix contains links between the threats and controls.

We can collect the information into a tabular format, which represents the relationship between the row and the column element of the table. The cell contains one of the three values: low, medium or high [86]. For each asset-vulnerability pair, threats are identified and the control strategies for each threat are identified when the controls are in place or when new approaches for the controls exist.

Similar to the tabulation of asset-vulnerability-risk relationship is the ISO27001 asset valuation and risk matrix [87].

Yet another approach, is the treat, vulnerability, and asset (TVA) model defined by Detmar et al. that begins by identifying vulnerabilities for every asset-threat pairs and then control strategies are defined for the identified vulnerability [61].

Assets are identified by the information or data that is crucial to the organization, while a threat is identified by the object, person, or entity that is a constant danger to the asset.

To utilize a vulnerability and risk model, the asset must be identified. Since the proposed behavior extension should be integrated to an existing system, the software, hardware, and the data must be identified. In addition, the authentication system that the users are using is deemed an asset and must be defined. After listing these assets we ask the following questions [61]:

- Which information asset is the most critical to the success of the organization?
- Which information asset generates the most revenue?
- Which information asset generates the most profitability?
- Which information asset is the most expensive to replace?
- Which information asset is the most expensive to protect?
- Which information asset's loss or compromise would be the most embarrassing or cause the greatest liability?

Next step is to identify vulnerabilities. These vulnerabilities representing the systems that give each asset a meaning or that the threats are acting on. For example, the hardware is an asset. The webserver running on this asset might have a vulnerability due to the threat of a Denial of Service attack.

Next step is to identify threats by asking the following questions [61]:

- Which threats present a danger to the organization's information assets in its current environment?
- Which threats represent the most danger to the organization's information assets?
- How much would it cost to recover from a successful attack?
- Which threats would require the greatest expenditure to prevent?

An asset worksheet [61] can be defined as follows. The scale is divided into four categories representing the vulnerability's impact on the asset: 0-No impact, 1-Weak Impact, 3-Moderate Impact, and 9-Strong Impact[61]. An example of such an asset worksheet is shown in Table 3-11.

Table 3-11: Asset worksheet

	Assets & Costs									
Vulnerabilities	Trade Secrets (IP)	Client Secrets	Reputation (Trust)	Lost Sales/Revenue	Cleanup Costs	Information	Hardware	Software	Services	Communication
Web Servers										
Application Servers										
Firewalls										
Client Nodes										
Databases										

Assume that there are m assets where the relative cost of asset a_j is C_j ($j = 1,\ldots, n$) [86]. In addition, let C_{ij} be the impact of vulnerability v_i on asset a_j [86]. Then the relative cumulative impact of a vulnerability V_i on the assets of the organizations is:

$$V_i = \sum_{j=1}^{j=n} v_{ij} \times C_j$$

The resulting vulnerability worksheet [61] is shown in Table 3-12.

Table 3-12: Vulnerability worksheet

	Vulnerabilities								
Threats	Web Servers	Application Servers	Firewalls	Client Nodes	Databases	Software on server	Software on client	Power	Transmission
Denial of Service Attacks									
Spoofing & Masquerading									
Malicious Code (Viruses, etc.)									
Human Errors (Accidental)									
Insider Attacks (Malicious)									
Intrusion									
Spamming									
Physical Damage to Hardware									

Assume that there are p threats that impact the n vulnerabilities and d_{ki} is the potential of damage from threat t_k due to vulnerability v_i [86]. Then the relative cumulative impact of the threat T_k is:

$$T_k = \sum_{i=1}^{i=m} d_{ki} \times V_i$$

The evaluation of the threats for behavior authentication method is as follows [61]:

- Deliberate software attacks

  The internet protocol is vulnerable to denial-of-service attacks.
  The illegitimate user can learn details of the behavior authentication method and can replicate a legitimate user behavior by eavesdropping or shoulder surfing.

- Act of human error or failure

  If administrators make configuration errors in the behavior authentication mechanism, the end users may be disrupted by the behavior extension.

- Technical software failures or errors

  If Javascript is not enabled in the browser, then the behavior authentication mechanism stops working.

- Technical hardware failures or errors

The underlying hardware that runs the web application can fail, thus causing an outage. Additionally, power system failures are always possible.

- Quality of service deviations from service providers

Unless administrators work on potential problems in the configuration of the behavior authentication extension, failure is probable over time.

- Deliberate acts of theft

The end user's routine of in the web application is crucial to protect. Therefore, confidentiality shall be provided by adopting measures such as securing the end user's communications using SSL, otherwise traffic may be captured and the user's behavior stolen - thus compromising the system.

- Deliberate acts of sabotage or vandalism

The internet protocol is vulnerable to denial-of-service attacks. If there is such an attack, then the behavior extension will stop working.
The domain name system may be subject to cache poisoning, thus enabling an illegitimate user to act as a legitimate user.

- Technological obsolescence

Unless reviewed and periodically updated, the behavior extension method might have security problems that are introduced by browser updates and OS updates.

- Forces of nature

All information assets in the organization are subject to forces of nature, unless suitable controls are provided.

- Compromises to intellectual property

The behavior extension method itself has little intrinsic value, but other assets protected by it could be attacked if the underlying system is compromised.

The threats worksheet can be identified [61] as shown in Table 3-13.

Table 3-13: Threat worksheet

	Threats					
Controls	Denial of Service Attacks	Spoofing & Masquerading	Malicious Code (Viruses, etc.)	Human Errors (Accidental)	Insider Attacks (Malicious)	Spamming
Denial of Service Attacks						
Spoofing & Masquerading						
Malicious Code (Viruses, etc.)						
Human Errors (Accidental)						
Insider Attacks (Malicious)						
Intrusion						
Spamming						
Physical Damage to Hardware						

Let us assume that there are q controls that can mitigate the p threats and e_{ok} is the impact of control z_o on threat t_k. Then the relative cumulative impact of the control Z_o is:

$$Z_o = \sum_{l=1}^{l=p} e_{ol} \times T_l$$

After calculating the cumulative risk, it is feasible to decide whether the given method is risky or not.

3.10.2.1 Planned Cost Benefit Analysis

Prior to integrating the behavior extension authentication method, it would be good to investigating the cost / investment in this method.

The investment depends on the number of man-month metric needed to integrate this method into an existing environment. The total time spent on integration affects how much money will be spent. Then, determining the revenue that organization will earn from this method must be assessed. This basic work needs to be done by the organization's management in order to determine if it is worth investing in introducing this method.

In this thesis, the planned cost benefit analysis of behavior authentication extension does *not* include a quantitative analysis, but rather the cost-benefit analysis takes a theoretical perspective when evaluating the system and software complexities, storing user behavior and the cost of comparison a user's behavior with the template, load testing/boundary value cost, and user behavior monitoring costs.

3.11 Software Tools

Visual Studio 2010 [88] has been used to implement the behavior authentication extension.

The software used for the analysis are G*Power 3.1.9.2 and IBM® SPSS® Statistics 22.

G*Power [89]is used for power analysis to determine the sample size prior to experiments. G*Power can be downloaded from http://www.gpower.hhu.de/. The documentation can also be found at the same address [90].

SPSS Statistics is a statistical analysis program [91]. In 2009, the program was acquired by IBM. SPSS can be downloaded for trial use [92]. A wide array of sources can be found for SPSS. I primarily used the online documentation [93]. I used SPSS for the Wilcoxon-Mann-Whitney and χ^2 tests.

3.11.1 Evaluation of framework

The original site web site (aftonblatte.se) is a ASP.NET Website project. The original Website project has been converted to a .NET Web application in order to handle server side integrations, SCRAM integration, AJAX calls, and extensive logging. The framework used in the experiments is ASP.NET Framework 4.0. The reason for choosing this target framework is that the ASP.NET AJAX integration was not possible prior to 3.5 [94].

As was mentioned in Section 3.5.1, the SCRAM SHA-1 implementation that has been used was originally implemented by Fizachi [71]. The reason for choosing this implementation is that it covers the message exchanges in a complete manner and it has both SCRAM server and client implementation within a single project. As a result, I can easily integrate the server part with the existing web application. In the original implementation, some parameters such as username, nonce, password, salt and client-server messages are hardcoded for testing purposes. In the behavior extension method, I did not want the experimenting user to deal with client messages. So, from the web application, the username and password supplied by the user is passed. The nonce and client messages are defaulted to the application. In the end, the SCRAM SHA-1 server decides whether the user is legitimate or not.

3.11.2 Implementation of the Behavior Authentication Method

The behavior authentication algorithm collects behaviors via an AJAX request-response mechanism. This AJAX logic is implemented as a part of JQuery in the frontend of the custom user control News.ascx. The AJAX calls are asynchronously sent to the backend in JSON format via a POST method (as shown in Figure 3-32). The AJAX response can contain additional information to locate the HTML element. This additional information is also used by the behavior authentication algorithm. The class diagram of the behavior authentication method can be found in Appendix A: Class diagram of the implementation (Elapsed Authentication Time - Control Group).

The backend ASP.NET code evaluates the AJAX response against an XML template file. The XML file content is extracted with XPath according to the identifier.

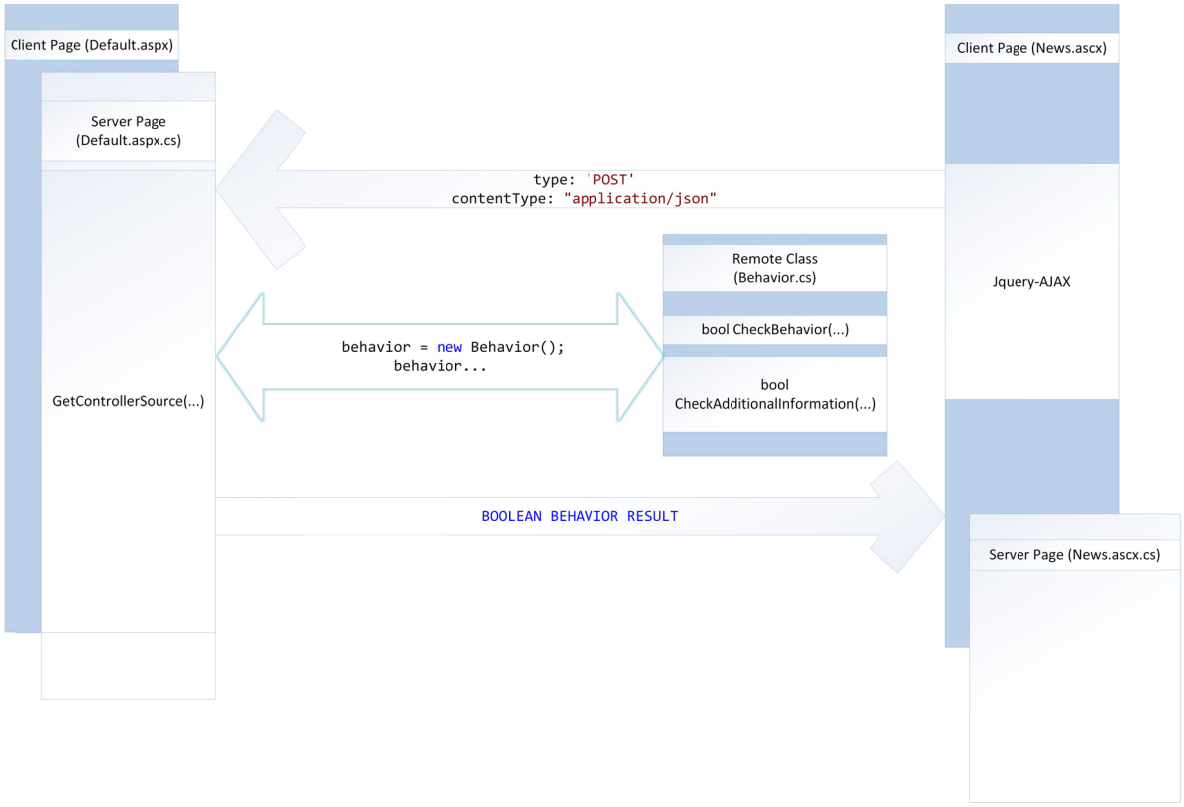


Figure 3-32: Behavior Authentication Data Flow

Figure 3-33 shows the data flow from backend News.ascx.cs to the client page AuthInit.aspx. The behavior extension method is comparing actions of a user on a controller (i.e., the user's behavior) to a template and if the user action does not fit to the template, the user is redirected to AuthInit.aspx. Depending upon the credentials that the user provided, the user is redirected to his/her previous page or the user is logged out of the system. The credentials that the user provided are checked by the SCRAM server (See Figure 3-34).

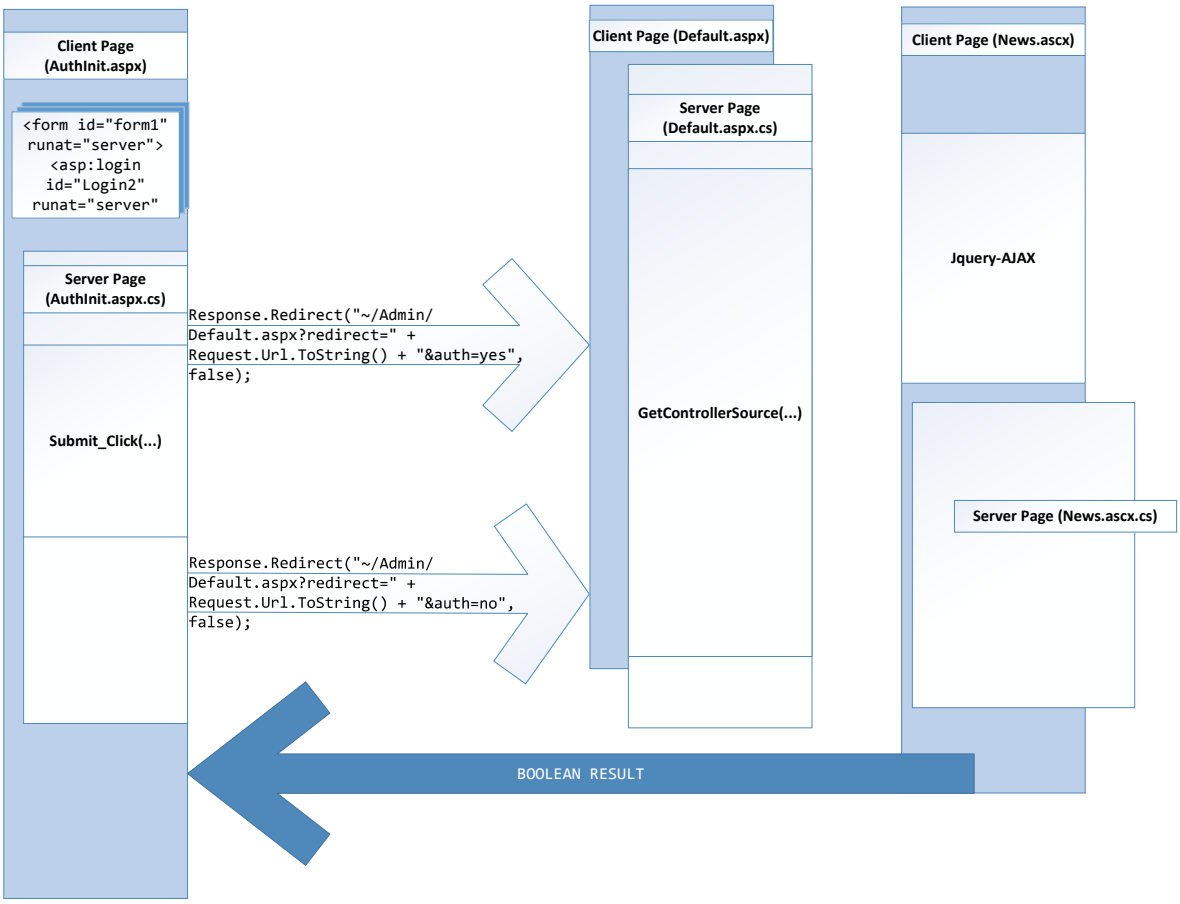


Figure 3-33: Behavior Authentication-SCRAM Authentication

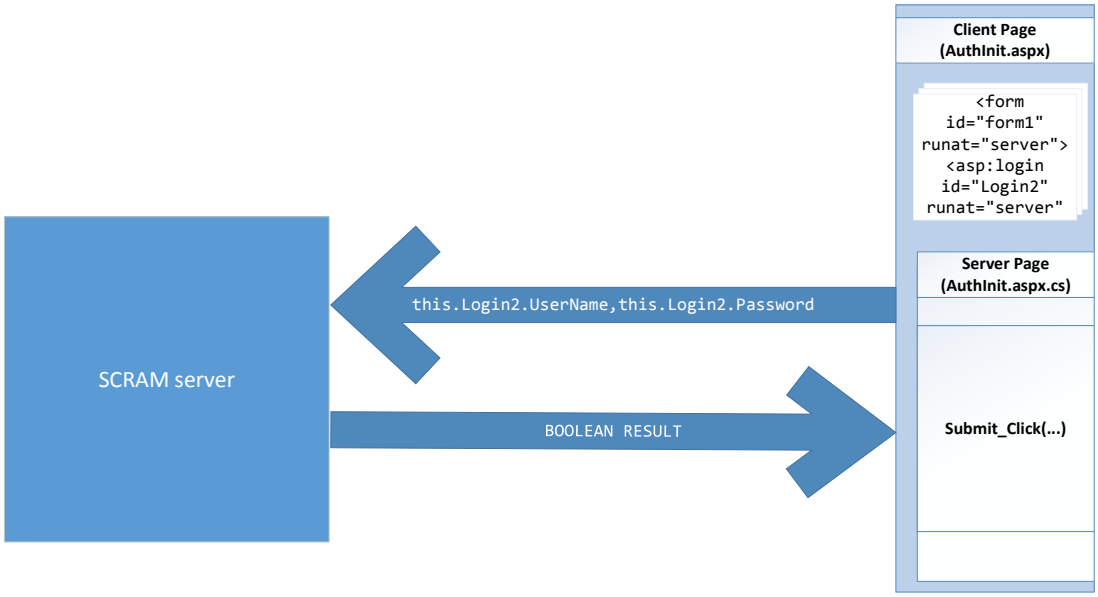


Figure 3-34: SCRAM server-client interaction

The behavior extension is depends on AJAX functionality, since AJAX provides the gateway between client and server (hence the behavior extension is independent of ASP.NET postbacks). For this reason, it is important that the user's web browser is capable of handling AJAX requests. If the client browser does not allow Javascript to run, then the behavior extension will not work. If the expected AJAX response does not arrive, then the current implementation provides a failover method, to handle the extension methods, thus enabling the current web application to continue to run. This is not logged by the application and there is no trigger that prompts user to turn on Javascript.

In the current implementation, every user action is sent back and forth via a AJAX method. This could potential be a problem if many user actions need to be checked by the behavior authentication extension. During the experiments there were no concurrent users (See 5.2.5 "Concurrency limitation").

In Appendix A: Class diagram of the implementation (Elapsed Authentication Time - Control Group) and Appendix B: Class diagram of the implementation (Elapsed Authentication Time - Experiment Group), the class diagrams of both control and experiment data for elapsed authentication time experiments can be seen.

3.11.3 Discarded components

As noted in Section 3.5.2, the originally planned experimental design was to provide the test beds as a VMWare [95] VMs. Identical copies of the VM were planned to be distributed, one to each member of the groups. The operating system used in the VM was Microsoft's Windows 7 Professional 64-bit operating system. Both Microsoft and VMWare solutions are licensed solutions. Microsoft Windows 7 Operating System is available via DreamPark [96]. The VMWare Player is available as a free download for non-commercial uses [95]. Within each VM, Microsoft IIS was set up to host the applications required by the experiment.

However, after conducting several preliminary experiments with different host machines, the VM performance was found to varying depending on the local host OS resources. Another reason to abandon VMWare solution was that providing the VMs to different test users was very hard due to the size of the VMs (the configured VM was 25 GB in size). When using FTP to download the VMs to the client, there were problems with the CRC values of the files not matching between the local client and remote FTP server machines. It is known that the Ethernet and TCP CRCs are not sufficient to detect all frames or TCP segment errors [97].

In the original experimental design, the test bed consisted of an IdP which checks the user's identification and an SP which provides a service to the system. In this scenario, a user initiates interaction with the system via a web browser to a target system (the system that the client wants to perform an operation on). The SSO solution that was planned to be used is based on the Componentspace SAML library [98]. This library provides a reference implementation of SSO that can be fine-tuned and integrated with an existing solution. However, using an SSO framework was discarded since it was unnecessary as was explained in Section 3.5.

In the original experimental design, it was planned that the implementation would record each postback done by the user. This approach was selected because ASP.NET web pages have a lifecycle which includes postbacks [72]. These postback actions are mediated by an ASP.NET web page that sends the entire page and its contents back to the server [72]. However, not every user action causes a postback – as a result the actual experiment design does not depend upon the entire web page being sent back. Instead of implementing a solution that only utilizes postbacks, I implemented a solution in which AJAX calls are invoked for *every* user action. By using this approach, I am able to capture all actions – at the cost of requiring that Javascript is enabled in the user's web browser.

4 Analysis

The experiments conducted for hypothesis testing are evaluated in this chapter. The experiment data can be found in Appendix E. The methods explained in Section 3.10 are used to evaluate the results of the experiments. Given the null hypotheses:

- SSO with a behavior extension is **not** faster than SSO with a secondary authentication mechanism* and
- SSO with the behavior extension produces an **unacceptable** rate of false positive and false negative results.

The metrics for the analysis are continuous authentication time and dichotomous accept and reject rates.

4.1 Elapsed Authentication Time Experiments Data Analysis

To analyze the elapsed authentication time experiment data, I use Mann-Whitney U Test [99]. Although this test requires two dichotomous groups, the outcome is ordinal data. The test requires ordinal input data or one can use continuous values such as elapsed time [100p. 2].

In this analysis the collected values in both control and experiment groups are tabulated with states as 1 and 2 respectively and the total elapsed time for authentication are provided as input values (see Table 4-1).

Table 4-1: Data Analysis – all times in Milliseconds (ms)

States	ms	States	ms	States	ms	States	ms
1	26.26	1	103.05	2	0.32	2	0
1	170.14	1	58.8	2	0.34	2	0.37
1	25.67	1	74.95	2	0.35	2	0.28
1	39.75	1	33.59	2	0.54	2	0.28
1	125.58	1	53.12	2	1.44	2	0.15
1	42.05	1	35.61	2	1.37	2	0.32
1	31.11	1	25.35	2	0.38	2	0.33
1	17.97	1	12.26	2	15.83	2	0.36
1	23.61	1	25.25	2	1.3	2	0.37
1	37.27	1	32.17	2	1.31	2	0.02
1	29.16	1	160.73	2	0.35	2	0.36
1	116.17	1	29.72	2	0.28	2	0.28
1	285.47	1	38.57	2	0.53	2	0.35
1	597.95	1	25.65	2	0.82	2	0.39
1	2310.23	1	28.45	2	0.5	2	0.33
1	584.9	1	286.77	2	1.75	2	0.35
1	213.21	1	39.73	2	0.47	2	0.38
1	116.17	1	27.4	2	0.35	2	0.49
1	118.78	1	26.56	2	0.46	2	0.38
1	121.16	1	58.8	2	0.39	2	0.34
1	673.69			2	0.45		

* This means that the incremental time required by the additional authentications for each user requires more total time than my mechanism.

The descriptive statistics of the data set is shown in Table 4-2.

Table 4-2: Descriptive Statistics

Descriptive Statistics

	N	Mean (ms)	Std. Deviation(ms)	Minimum (ms)	Maximum (ms)
Value	82	84.3722	278.64846	0.00	2310.23
States	82	1.5000	0.50308	1.00	2.00

In order to see the distribution of the continuous data, I needed to bin the data into ranges. An approach to circumvent this is to group the data into intervals. For this I chose to divide it into 10 intervals. The values shown in Table 4-3 are the grouped/bin values for the elapsed authentication time for control group experiment data [101] [102]. The distribution of the elapsed authentication time for the control group is shown in Figure 4-1.

Table 4-3: Binned data

Value (Binned)

Time (ms)		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	10.01 - 20.00	2	4.9	4.9	4.9
	20.01 - 30.00	11	26.8	26.8	31.7
	30.01 - 40.00	8	19.5	19.5	51.2
	40.01 - 50.00	1	2.4	2.4	53.7
	50.01 - 60.00	3	7.3	7.3	61.0
	70.01 - 80.00	1	2.4	2.4	63.4
	100.01 - 110.00	1	2.4	2.4	65.9
	110.01 - 120.00	3	7.3	7.3	73.2
	120.01 - 130.00	2	4.9	4.9	78.0
	160.01 - 170.00	1	2.4	2.4	80.5
	170.01 - 180.00	1	2.4	2.4	82.9
	210.01 - 220.00	1	2.4	2.4	85.4
	280.01 - 290.00	2	4.9	4.9	90.2
	300.01+	4	9.8	9.8	100.0
	Total	41	100.0	100.0	

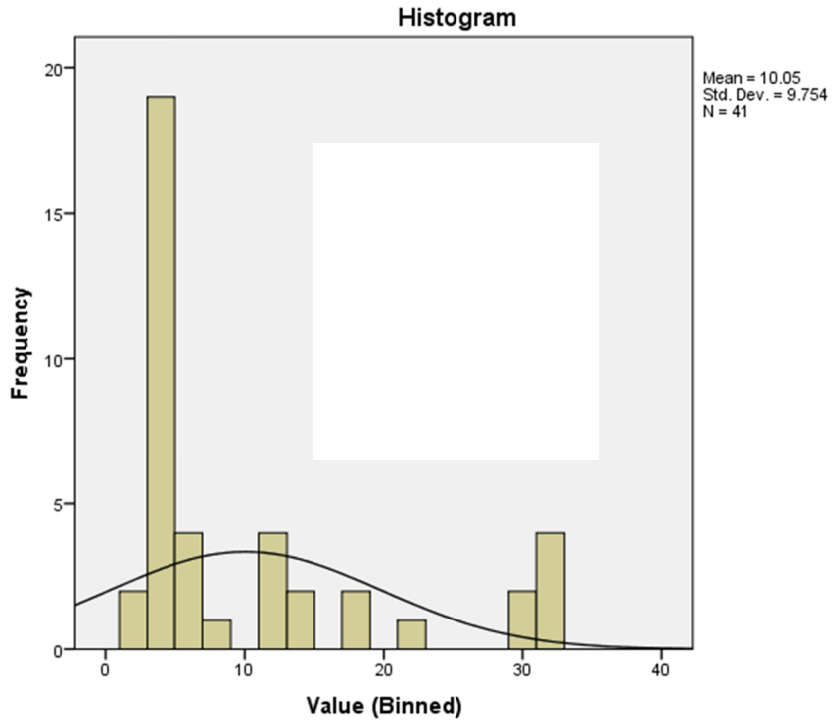


Figure 4-1: Distribution of control group data

Similarly, the values for the experiment group's raw data are binned. I chose an interval value of 0.05 ms. Table 4-4 shows these binned values.

Table 4-4: Binned data

Time (ms)		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	<= .00	1	2.4	2.4	2.4
	.01 - .05	1	2.4	2.4	4.9
	.16 - .20	1	2.4	2.4	7.3
	.26 - .30	4	9.8	9.8	17.1
	.31 - .35	8	19.5	19.5	36.6
	.36 - .40	12	29.3	29.3	65.9
	.41 - .45	1	2.4	2.4	68.3
	.46 - .50	4	9.8	9.8	78.0
	.51 - .55	2	4.9	4.9	82.9
	.81 - .85	1	2.4	2.4	85.4
	1.26 - 1.30	1	2.4	2.4	87.8
	1.31 - 1.35	1	2.4	2.4	90.2
	1.36 - 1.40	1	2.4	2.4	92.7
	1.41 - 1.45	1	2.4	2.4	95.1
	1.76 - 1.80	1	2.4	2.4	97.6
	1.96+	1	2.4	2.4	100.0
	Total	41	100.0	100.0	

The distribution of the elapsed authentication time for control group data is shown in Figure 4-2.

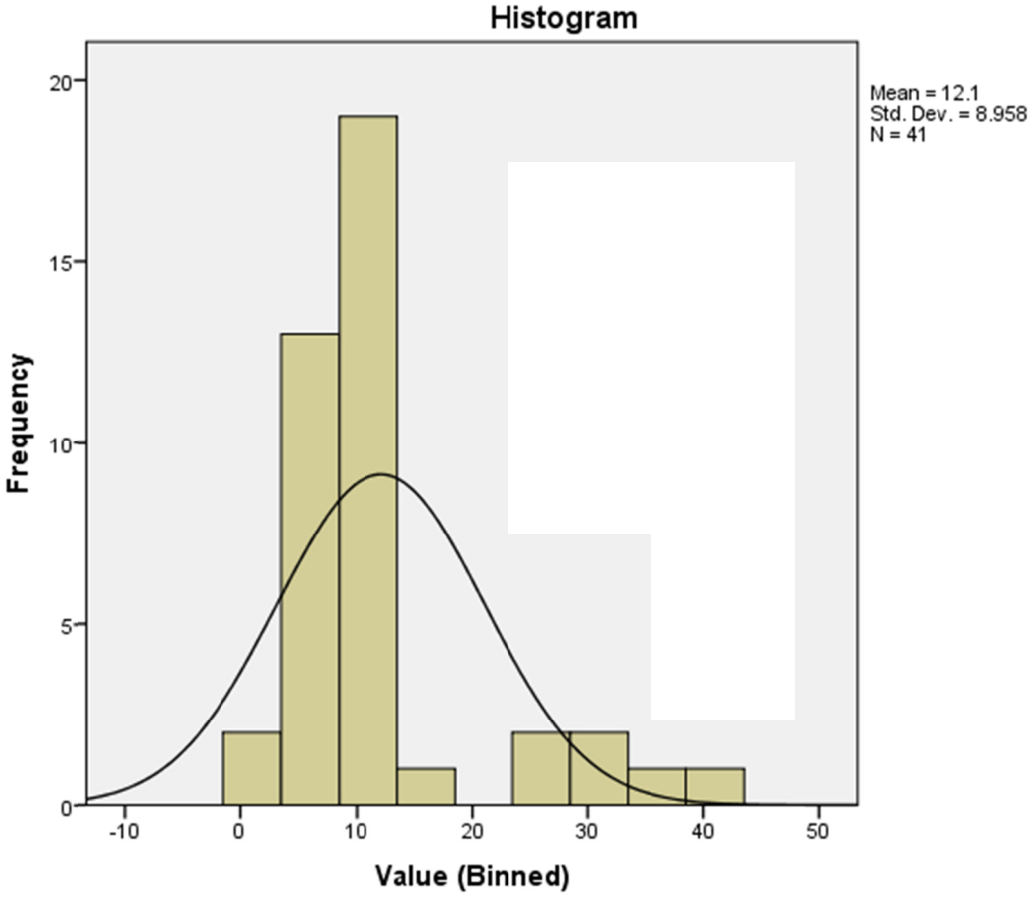


Figure 4-2: Distribution of experiment group data

As it can be seen, the distributions for control group and experiment group elapsed authentication time data are not similar. The experiment group's data is more normally distributed than the control group's data.

Mann-Whitney Test ranks and test statistics are illustrated in Table 4-5 and

Table 4-6.

Table 4-5: Ranks

	States	N	Mean Rank	Sum of Ranks
Value	1.00	41	61.98	2541.00
	2.00	41	21.02	862.00
	Total	82		

Table 4-6: Test Statistics

	Value
Mann-Whitney U	1.000
Wilcoxon W	862.000
Z	-7.786
Asymptotic significance (2-tailed)	.000

a. Grouping Variable: States

The result is statistically significant at p = 0.05, since the asymptotic significance is 0.000. Because the asymptotic significance is less than 0.05, we reject the null hypothesis.

The Mann-Whitney U test is equivalent to the one tailed Student's T-test, except that it does not assume a normal distribution, hence one can still infer how a measurement variable is varying due to the mean differences [103]. However, Mann-Whitney U-test assumes that the two distributions of the data are the same [100p. 1]. Unfortunately, the distributions are different. Hence I cannot draw any conclusions by using this method. If the distributions were similar, I can have further conclusions by comparing the medians of two groups

Figure 4-3 depicts the mean ranks between the groups. If the means of the ranks in the two groups are very different, then the P value will be small [104]. The corresponding test statistics are shown in Table 4-6.

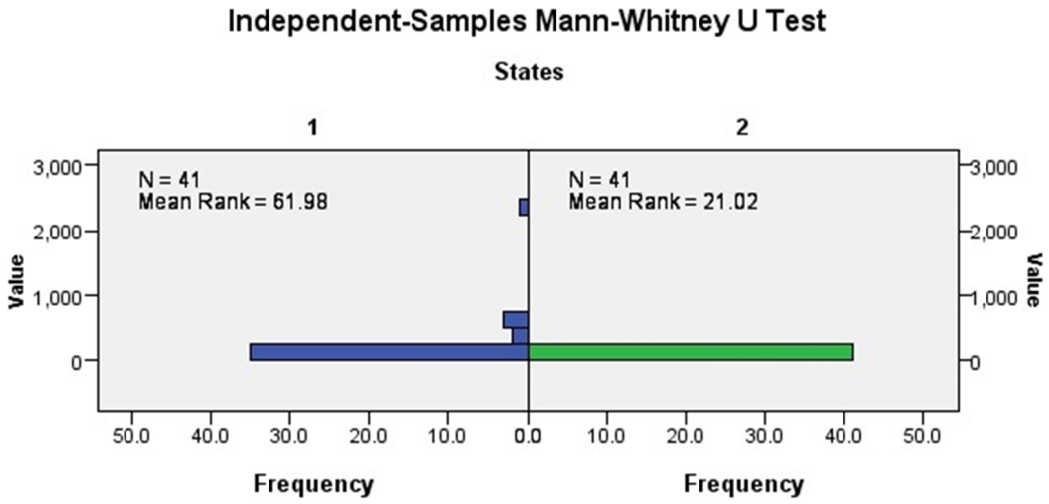


Figure 4-3: Frequencies in Control and Experiment Group

4.2 Legitimate and Illegitimate User Group Experiments Data Analysis

This section evaluates the False Acceptance Rate and False Rejection rate for Legitimate and Illegitimate User Group Experiments. This section also analyzes the goodness of fit for these experiments.

4.2.1 False Acceptance Rate (FAR) and False Rejection Rate (FRR)

The Illegitimate User Group Experiment data is used in order to analyze FAR. Whereas, the Legitimate User Group Experiment data is used in order to analyze FRR. The data collected from Illegitimate User Group Experiments is shown in Table 4-7. In this table, the keyword *Legitimate* means that the user followed the template behavior, whereas the keyword *Illegitimate* means that the user did not follow the template behavior. In addition, the Control Test indicates the ideal condition for illegitimate users (whom should be rejected), whereas the experiment reality represents whether the user matches the behavior template or not.

Table 4-7: Collected Data for FAR analysis

ControlTest	ExperimentReality	ControlTest	ExperimentReality
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Legitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Legitimate	Illegitimate	Legitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate	Illegitimate	Illegitimate
Illegitimate	Illegitimate		

The data collected from Legitimate User Group Experiments is shown in Table 4-8. In this table, as in the previous table, the keyword *Legitimate* means that the user followed the template behavior whereas the keyword *Illegitimate* means that the user did not follow the template behavior. In this table, the Control Test represents the ideal condition for legitimate users (who should be accepted), whereas the experiment reality indicates whether the users match the behavior template or not.

Table 4-8: **Collected Data for FRR analysis**

ControlTest	ExperimentReality	ControlTest	ExperimentReality
Legitimate	Legitimate	Legitimate	Illegitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Illegitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Illegitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Illegitimate	Legitimate	Illegitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate	Legitimate	Legitimate
Legitimate	Legitimate		

Given the above data, there are 3 illegitimate users who were falsely accepted by the algorithm (as shown graphically in Figure 4-4). There are 5 legitimate users who were falsely rejected by the algorithm (as shown graphically in Figure 4-5).

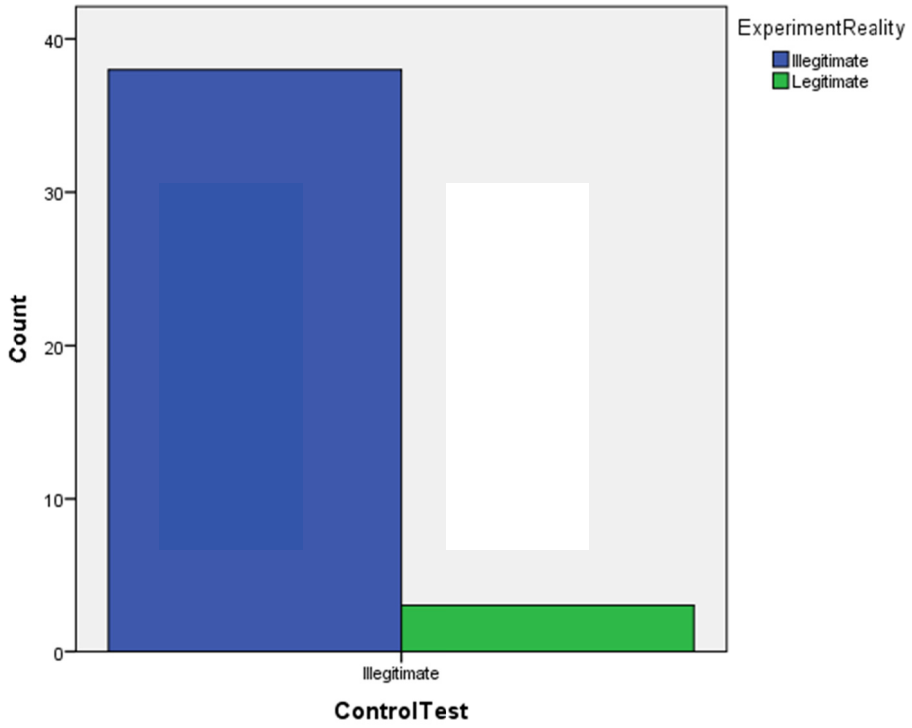


Figure 4-4: **Illegitimate-Legitimate user representation**

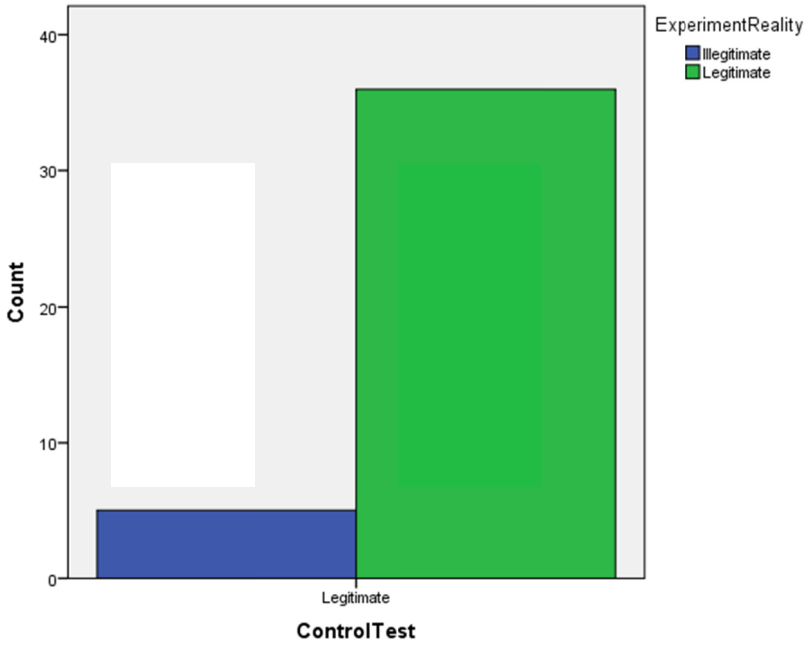


Figure 4-5: Illegitimate-Legitimate user representation

To analyze the Illegitimate Users and Legitimate Users experiment data, I first merged this data into a single data file and use Crosstabs in SPSS. Crosstabs is a method that cross-tabulates two variables, and displays their relationship in tabular form [105]. The two variables here are Control Test which has illegitimate data which represents the perfect control design where all users are deemed to be illegitimate in this experiment. The other variable is the experiment reality which is the experiment results.

Table 4-9: Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
ControlTest * ExperimentReality	82	100.0%	0	0.0%	82	100.0%

Table 4-10: ControlTest * ExperimentReality Crosstabulation

			ExperimentReality		
			Illegitimate	Legitimate	Total
ControlTest	Illegitimate	Count	38	3	41
		% within ExperimentReality	88.4%	7.7%	50.0%
	Legitimate	Count	5	36	41
		% within ExperimentReality	11.6%	92.3%	50.0%
Total		Count	43	39	82
		% within ExperimentReality	100.0%	100.0%	100.0%

When the control test result is illegitimate, then on 38 occasions the reality is also positive. This means that there is an 88.4% true rejection rate and a 7.7% false acceptance rate (also known as a type II error).

When the control test result is legitimate, there is a 92.3% true acceptance rate and an 11.6 % false rejection rate (also known as a type I error).

4.2.2 Goodness of fit for illegitimate user experiment

To analyze the goodness of fit for chi square for illegitimate user experiment, I examine the rates with the following FAR & Non-FAR counts in Table 4-11 [106].

Table 4-11: Goodness of Fit setup for FAR analysis

Rank	Observed FAR	Expected FAR
FAR	3	2.05 (5%)
Non-FAR	38	38.95 (95%)

The descriptive statistics for observed FAR, Observed FAR - Expected FAR comparison, and test statistics are shown in Table 4-12, Table 4-13, and Table 4-14 respectively.

Table 4-12: Descriptive Statistics for Observed FAR

	N	Mean	Std. Deviation	Minimum	Maximum
ObservedFAR	41	35.44	9.228	3	38

Table 4-13: Observed FAR-Expected FAR comparison

	Observed N	Expected N	Residual
FAR	3	2.1	.9
Non-FAR	38	38.9	-.9
Total	41		

Table 4-14: Test Statistics

	ObservedFAR
Chi-Square	0.463[a]
df	1
Asymp. Sig.	0.496

a. 1 cells (50.0%) have expected frequencies less than 5.
The minimum expected cell frequency is 2.1.

The results present above indicate that there is no statistically significant departure from the expected values [107]. This is reflected in the probability of the given experiment having the value 0.496 which is not significant at the given level of significance of 0.05. In other words, since the asymptotic significance is greater than 0.05, the null hypothesis that the *two distributions are the same* is not rejected.

4.2.3 Goodness of fit for legitimate user experiment

To analyze the goodness of fit for chi square, I examine the rates with the following FRR & Non-FRR counts in Table 4-15.

Table 4-15: Goodness of Fit setup for FRR analysis

Rank	Observed FRR	Expected FRR
FRR	5	2.05 (5%)
Non-FRR	36	38.95 (95%)

The descriptive statistics, observed FRR-Expected FRR comparison, and Test Statistics are shown in Table 4-16, Table 4-17, and Table 4-18.

Table 4-16: Descriptive Statistics for Observed FRR

	N	Mean	Std. Deviation	Minimum	Maximum
ObservedFRR	2	20.50	21.920	5	36

Table 4-17: Observed FRR-Expected FRR comparison

	Observed N	Expected N	Residual
FRR	5	2.1	3.0
Non-FRR	36	39.0	-3.0
Total	41		

Table 4-18: Test Statistics

	ObservedFRR
Chi-Square	4.469[a]
df	1
Asymp. Sig.	.035

a. 1 cells (50.0%) have expected frequencies less than 5. The minimum expected cell frequency is 2.1.

Although the FRR rate is 11.6% which is unacceptable in comparison with the expected 5%, the above result indicates that there is statistically significant departure from the expected values and since asymptotic significance is less than 0.05, the null hypothesis that *the two distributions are the same* is rejected [108].

4.3 Risk Analysis

This section analyzes the risk of using the proposed behavior extension in a hypothetical environment. This analysis has various components. The section concludes with an overview of what kinds of control should be conducted when integrating the proposed behavior extension.

For this analysis, I use the matrices which represent assets, vulnerabilities, threats, and controls as per [61].

There are already several approaches for matrix formation. One example is Kamat's Asset Valuation and Risk Analysis matrix which considers Confidentiality, Integrity and Availability (CIA) [87]. However, to create a more complete picture, I needed to include the components of the environment [86].

In this methodology:

1. The relationship between assets & costs and vulnerabilities are ranked with numeric values.
2. The relationship between vulnerabilities and threats are ranked with numeric values.
3. The relationship between threats and controls are ranked with numeric values.

The numeric values (as described in Section 3.10.2) are given with classification of relation between the group elements. Then for each group, the priorities are set. After calculating the ranks, the evaluation of a group member takes place.

In Table 4-19, after calculating the total score by evaluating assets against their vulnerabilities, a ranking is done. This ranking is a cue for the threat matrix. In this table, the highest total score is **application and SSO servers**. This means that the relationship between assets and threats to these components has the highest relationship rank. Thus, an emphasis on those components should be given in the threat matrix analysis.

Table 4-19: Vulnerabilities Matrix (Priority Ranking 1&2: not important; 3: Important not a Key Driver, 4: Important, but impacted by Key Drivers; and 5: Key Driver)

		Priority											
		10	9	8	7	6	5	4	3	2	1		
Vulnerabilities	Assets & Costs	Trade Secrets (IP)	Client Secrets	Reputation (Trust)	Lost Sales/Revenue	Information Integrity	Information confidentiality	Information availability	Software	Hardware	System cleanup	Total Score	RANK
Application Servers	5	9	9	9	3	9	9	9	9	9	9	453	12
SSO Servers	5	9	9	9	3	9	9	9	9	9	9	453	11
Web Servers	5	9	9	9	7	9	9	9	9	9	9	481	13
Physical Security	5	3	3	9	9	3	3	3	1	9	0	258	5
Data Transmission	4	9	9	9	9	9	3	9	9	3	1	445	10
Behavior authentication	4	9	9	9	9	3	3	9	9	9	1	421	9
Password strength	4	9	9	3	1	1	9	1	9	1	0	286	7
Insecure wireless	3	9	9	3	1	1	9	1	1	1	0	262	6
Databases	4	9	9	9	3	3	1	9	9	9	3	371	8
Firewalls	3	1	3	3	1	1	9	1	3	9	0	150	4
Client Nodes	3	1	3	1	3	3	3	1	3	1	0	114	2
SSL	3	1	3	3	1	1	9	1	1	1	0	128	3
Power outage	1	0	0	0	3	1	0	9	1	1	0	68	1

In Table 4-20, the relationship between vulnerabilities and threats is tabulated. The total score against every threat is ranked and highest rank represents the threat that would have the most impact on the system. In this table, the highest rank threat is *insider attacks*, which would be deemed as the least desired threat.

Table 4-20: Threat Matrix (Priority Ranking 1&2: not important; 3: Important not a Key Driver, 4: Important, but impacted by Key Drivers; and 5: Key Driver)

		Priority											
		10	9	8	7	6	5	4	3	2	1		
Threats	Vulnerabilities	Application Servers	SSO Servers	Web Servers	Physical Security	Data Transmission	Behavior authentication	Password strength	Insecure wireless	Databases	Power outage	Total Score	RANK
Intrusion	5	9	9	9	3	9	9	9	9	9	0	444	6
Server software Failures	5	9	9	9	3	9	9	9	9	9	3	447	8
Insider Attacks	5	9	9	9	7	9	9	9	9	9	0	472	9
Spoofing & masquerading	5	3	3	9	9	3	3	3	1	9	0	258	1
Denial of Service	4	9	9	9	9	9	3	9	9	3	1	445	7
Human error (Accidents)	4	9	9	9	9	3	3	9	9	9	1	421	5
Theft of computers (laptops/servers)	4	9	9	3	1	1	9	1	9	1	0	286	3
Malicious Code (Viruses, Worms, etc.)	3	9	9	3	1	1	9	1	1	1	0	262	2
Buffer Overflow attacks	4	9	9	9	3	3	1	9	9	9	0	368	4

In Table 4-21, the relationship between threats and controls are tabulated. The total score of every control is ranked and highest rank represents the control that should be applied to the system to minimize the overall threat impact. *Auditing and monitoring* as a control measure have the highest rank (355) according to this analysis.

Table 4-21: Control Matrix (Priority Ranking 1&2: not important; 3: Important not a Key Driver, 4: Important, but impacted by Key Drivers; and 5: Key Driver)

		Priority										
		9	8	7	6	5	4	3	2	1		
Controls	Threats	Intrusion	Server software failures	Insider Attacks	Spoofing & masquerading	Denial of Service	Human error (Accidents)	Theft of computers (laptops/scanners)	Malicious Code(Viruses, Worms, etc)	Buffer Overflow attacks	Total Score	RANK
Security Policy	5	9	1	1	9	9	3	3	9	9	243	6
Hardening of Environment (physical)	5	9	3	1	3	1	3	1	0	0	150	3
Firewalls	5	9	3	9	9	9	1	3	9	3	301	8
Configuration of Architecture	5	9	9	9	9	9	3	0	3	9	342	9
User disclosure of credentials, passwords, behavior patterns	5	9	1	9	9	3	3	1	1	1	239	5
Employee Training	4	1	1	3	3	3	9	3	3	3	125	2
Auditing & Monitoring (logs, spybot, etc.) -IDS	4	9	9	9	9	9	1	3	9	9	355	10
System Administrative Due diligence	4	9	9	3	3	9	1	0	3	3	250	7
DMZ	3	9	1	1	3	9	1	0	3	1	170	4
Single Sign-on	4	1	0	1	1	0	0	0	1	1	25	1
GPS tracking system (Asset Tracking System)	1	1	0	3	1	3	0	9	1	0	80	2

It is important to emphasize that the components in these tables are hypothetical and the ranking might vary with respect to the organization. However, the methodology applied above provides sufficient information while analyzing the relationship between assets, vulnerabilities to those assets, threats to those vulnerabilities, and finally control measures for those threats.

4.4 Major results

This section describes the major results with respect to the following hypotheses:

- SSO with a behavior extension is faster than SSO with a secondary authentication mechanism.

  For this hypothesis, the experiment results show that SSO with behavior extension is faster than SSO with a secondary authentication mechanism, namely, SCRAM-SHA authentication. The result is statistically significant at $p = 0.05$ since the significance value is 0.00. Thus, we reject the null hypothesis. It is important to emphasize that SSO with the behavior extension is completely dependent upon the task and/or the user activity. If the given task was longer, thus the user would needs to interact with more controllers, and then eventually this hypothesis would be falsified. Similarly, if the user fiddles on the form independent of the assigned task, then the proposed behavior extension method would be slower.

- SSO with the behavior extension produces an acceptable rate of false positive and false negative results.

  For the above hypothesis, the experimental results show that the behavior extension method has unacceptable FAR (7.7%) and FRR (11.6%) values. The expected experiment results were at most 5% for both FAR and FRR rates. Since the asymptotic significance is 0.496 for illegitimate user group experiment; I fail to reject the null hypothesis for FAR rates. In contrast, the asymptotic significance is 0.035 for the legitimate user group experiment; hence I reject the null hypothesis for FRR rates.

As described earlier, the SCRAM SHA-1 implementation was originally implemented by Fizachi [71]. I integrate his solution as a fallback mechanism in the experiment group experiments and as a secondary authentication for control group experiments. However, during the preliminary analysis prior to the actual experimentation, I found out that Fizachi's implementation gave results that vary by more than expected. Some of values for authentication time are unexpectedly larger and perhaps should be considered as outliers.

I observed these outlier values during my preliminary analysis that was performed to determine the necessary sample size (See Section 3.8.1). During my preparation for elapsed authentication time tests, I was planning to use Student's T-test for comparing two continuous variables which are the authentication times for experiments. I needed to give up using Student's T-test due to the fact that I encountered outlier values. These outliers affected the statistical distribution and the distribution for statistical analysis is not a normal-Gaussian distribution. I kept these outlier values during my statistical analysis; hence these values were used in both power analysis and in the elapsed authentication time statistical analysis.

In order to analyze all of the observed values, I needed to use non-parametric analysis while evaluating elapsed authentication time experiments. As a result, I used Mann-Whitney U Test which is suitable for non-parametric continuous data. While this was suitable from a statistical analysis point of view, I should probably have addressed the base problem of the unexpected high variance of the SCRAM SHA-1 times.

5 Conclusions and Future work

This chapter gives my conclusions, limitations that this work has and suggests future works that could be done on this topic and required reflections.

5.1 Conclusions

This section explains the conclusions acquired throughout this research. The concepts of SSO and post authentication methods are revisited and the findings in elapsed authentication time and false accept rate and false reject rate are evaluated. Finally, in this section, the cost analysis of behavior extension is discussed.

5.1.1 SSO

SSO provides ease of use for end users as once the user is authenticated in one system, he/she can be authenticated to other systems without any hassle. This feature of SSO reflects the two sides of the coin due to the fact that SSO introduces a "Key to the Kingdom" problem as an attacker can gain access to other systems once he/she has the access to one system. One approach to mitigate this issue may be facilitating post authentication methods.

5.1.2 Post Authentication Methods

If SSO is the primary authentication method, then post authentication methods are secondary authentication methods. Well-known secondary authentication methods have advantages and disadvantages as described in Section 2.5. However, integrating those methods with existing systems having SSO may be a hard task for a system administrator. Similarly, from a user experience perspective, being required to perform multiple authentications may not be desirable.

5.1.3 Elapsed Authentication Time Evaluation

The elapsed authentication time means the time spent during authentication. This time depends on the authentication method and the infrastructure that the method is running on.

After analyzing the data acquired from the experiments, I concluded that SSO with a behavior extension is *faster* than SSO with the SCRAM authentication mechanism. However, the total authentication time is highly dependent upon the user interaction with the web application and the authentication time values might be larger or smaller depending on the user's task. This is also discussed in Section 3.3.

5.1.4 False Accept Rate and False Reject Rate Evaluation

FAR and FRR are important concepts when it comes to the security measures for a system. In a system that uses an authentication method it should minimize FAR and FRR.

After analyzing the data acquired from the experiments, I concluded that SSO with the specific behavior extension used in this study produces unacceptable rate of false positive and false negative results.

In my literature study, I found mentions of FAR and FRR evaluations in the majority of the authentication methods, with the exception of biometrics authentication methods. Biometrics authentication systems generally consider FAR and FRR rates, but if the threshold of the system is

adjustable, then it is not easy to determine whether the performance of a biometric system is reliable or not. In that respect, since the behavior extension method is comparing actions of a user on a controller (i.e., the user's behavior) to a template, the FAR and FRR rates are adjustable with respect to the changes in template. The difference is that FAR and FRR in biometric methods depends heavily on the biometric attribute that might change in time without realization of the administrators of the system, whereas in behavior authentication extension method, the admin has a complete control on the template file.

Conversely, other well-known authentication mechanisms assume the user is legitimate for the remainder of the session once they are successfully authenticated, hence they do ensure continuous authentication of the session whereas behavior extension fulfills this need.

5.1.5 Cost Analysis of Behavior extension

The costs of collecting, processing, and making decisions based upon the behavioral data are important factors to decide when planning to use the behavior extension method with existing web applications.

The behavior extension authentication method provides a lazy and transparent authentication, hence users do not need to use other (secondary) authentication methods. This means that less time is spent on repetitive request to authenticate.

The number of staff hours spent on repetitive processes * wage per hour * number of employees should literally be **zero**, since there will be no repetitive process *when* all users follow the behavior template. As a result the time spent by the user authenticating themselves following the initial login in procedure would be **zero** since there will be no post authentication requests.

Monitoring the behavior extension authentication method might add additional burdens to a security consultant who must modify the template as needed to prevent high false positives and true negatives. This might be a daunting task since the security consultant will generally be on alert for potential security violations that merit investigation. If the user really is legitimate, then the current implementation's fallback mechanism gives the user another chance to continue his/her task. As a result the security requirements of CIA are still provided. However, because there is a non-zero FAR rate there is a risk that attackers can gain access to one or more systems that they should have been denied access to.

5.1.5.1 System Complexity

In the experiments conducted in this thesis, the system complexity is not a hindrance when integrating the proposed behavior extension method to an existing website running under Windows 2008 and IIS. However, if there are multiple systems which are distributed over the network and if these systems are heavily coupled, then integration might be much harder.

Another aspect of the system's complexity is the maintainability of the behavior extension. If the system interlaces with many different technologies that each require mediator technologies to communicate, this may lead to maintenance problems for the behavior extension.

5.1.5.2 Software Complexity

The most notable difficulty that I encountered was the integration of the behavior extension with existing software.

Since the behavior extension method is a post hoc method that a programmer needs to integrate, this integration was not straightforward. In this thesis project, the behavior extension method was implemented purely in .NET and integrated to an existing ASP.NET solution. The effort

needed to integrate this solution depends heavily on the coupling and cohesion of the existing modules.

From an application developer's point of view, if the all the related code is very interdependent, then integration of the behavior extension method will be extremely hard. To ease this problem, the existing application should be implemented such that all modules are as independent as possible and everything in one module should be close to each other which is known as cohesion. (See Section 5.3).

5.1.5.3 User Behavior Storing and Comparison Costs

In the current implementation, the user behavior is stored in the memory and a single action on the webpage represents a behavior. The actions are not stored and handled cumulatively, but rather are considered one by one. The user behavior is compared against a user behavior template file. For all users, there is a single user behavior template file and it has a size as 1,998 bytes (1.95 KB). The size of the file is quite small and storing this is not a problem, but there might a future improvement in how this file is stored. (See Section 5.3).

The comparison of user behavior has an algorithmic cost for the behavior extension authentication method. However, since the algorithms used in the implementation were not compared to an existing implementation, it is hard to know whether the implemented solution is reliable or not (See Section 5.2).

Yet, the has proven that it can only provide moderately low FAR and FRR rates, unfortunately these are greater than the control experiment's assumption of 0% FAR and 0% FRR.

5.1.5.4 Load Testing / Boundary Value Cost

In this thesis, there was no investigation or load testing of the behavior extension method. The test bed can handle multiple user requests, but the tests were done one at a time rather than concurrently (See Sections 5.2 and 5.3).

5.1.5.5 User Behavior Monitoring Cost

In this thesis project, the user behavior monitoring has been done by logging both task completion time and elapsed authentication time for behavior extension. However, these logs were not evaluated during the experiments, but rather the evaluation took place after the experiments (See Sections 5.2 and 5.3).

5.2 Limitations

This section describes the limitations of this thesis project.

5.2.1 Lack of previous work

During the background analysis, I had difficulties finding resources about any behavior extension method or a similar approach that could be used as a security mechanism. This also affected my power analysis, as to do a power analysis one should use historical results as a starting point.

In this thesis project, the power analysis for elapsed authentication time experiment relies on a preliminary analysis of the control group experiments. The power analyses for legitimate and illegitimate user experiments were done with the *assumption* that the control groups represented a perfect scenario with 0% FAR and FRR rates.

5.2.2 Algorithm suitability

During the implementation of behavior extension authentication method, I used my own approach due to the fact that there was no similar implementation. The algorithms that I used for the behavior extension are essentially of my own design. The research does not generate any conclusion as to other behavior authentication methods.

5.2.3 User behavior comparison mechanism

In the current implementation, the user behavior is transferred from the client side to the server side via an AJAX method. What is compared is not a behavior, in the sense of a pattern of actions, but simply the controller ID that the user acted on. The current implementation does not evaluate the sequence of actions taken by the user to determine user legitimacy (See 5.3.

5.2.4 Behavior template limitation

The behavior template used in the experiments is not unique for a user, but rather is a generic template against which every user's action is checked. To provide uniqueness of behavior for each user, this template should be modified to suite every individual user (See Section 5.3).

5.2.5 Concurrency limitation

In ASP.NET, the controls and components of web forms are designed to compensate for the stateless behavior of the HTTP protocol. To complete requests and responses, a chain of events needs to be completed from the client web browser to the web server and also in the other direction. This chain of events constitutes the life cycle of ASP.NET.

Since HTTP is a stateless protocol, ASP.NET pages need a mechanism to store user information. This can be managed by Session, ViewState, or Application objects. Although the existing web application is using Session management, the behavior extension method is not using any state management system as the experiment was planned to be done by a single user at a time. Thus the concurrency of the current implementation of the extension method is limited.

5.2.6 HTTPS-TLS limitation

All experimentation has been done on an existing web application with HTTP and *without* HTTPS. Although an SSL initial key exchange adds to the latency, all of the communication between the client and server must be encrypted since the user behavior exchanged between the client browser and server and this behavior information is being used in the same was as if the web browser and web server were exchanging credentials.

In the current implementation, the TLS implementation of SCRAM SHA authentication server is based on the *SuperSocket* class by Kerry Jiang [109]. The server's SCRAM SHA-1 logic can be found in the *ScramCommand* class.

While TLS implementation protects against passive eavesdropping, it alone does not prevent man-in-the-middle attacks. To prevent such attacks the endpoints need to mutually assure their identities. This is part of SCRAM authentication mechanism; however, in the experiments, the user identity is assumed to be same in the experiments and all test group users used the same credentials during SCRAM authentication (See section 5.3).

Although the behavior extension itself does not provide SSL mechanism it is important that the existing web application provide a secure channel to prevent revealing plain text information across the network.

5.2.7 Scope limitation for statistical data analysis

In elapsed authentication time experiments, the statistical analysis has been done based upon total authentication times.

In these experiments, the total task time was also measured, but has not been evaluated. The reason for this is that the users in the control experiment could fulfill the given task in various ways which would greatly affect the total time. Hence it would not be possible to perform an unbiased statistical analysis based upon total task time in the control experiment. For example, a user might wait for an indefinite amount of time before completing the given task and this will generate a long task duration for this user. The same issue also occurs in the elapsed time authentication experiment group. Thus, although total task time was logged, it has not been evaluated in data analysis.

5.2.8 Validity-Reliability limitation

As stated in Section 3.10 the total number of experiments performed was forty-one.

During the preliminary experiments, I found that some experiments overlapped. So, I exploited this by utilizing subjects first as illegitimate users, then subsequently as legitimate users. It is important to emphasize that with the experimentation order explained in Section 3.10 a participant acting as a legitimate user cannot be used later as illegitimate user as this user is knowledgeable about the security mechanism since he/she already did the legitimate user experiment.

In these experiments, validity and reliability cannot be judged since they were deemed outside of the scope of this thesis project. Thus, it might be a good to repeat the experiments using a test-retest method to assess reliability (See Section 3.9.2). This would also increase the reliability of the results of this study.

5.3 Future work

In this thesis project, a proposed behavior extension authentication method is evaluated. There is obviously a need for a future work on this topic. Such future work would aim to ease the limitations of this method or present new behavior authentication mechanisms.

The next two obvious efforts might be:

- Securing AJAX calls so that Cross-Site Scripting (XSS) attacks have no effect on the behavior extension authentication method described in this thesis.

  XSS attacks are regarded as a general problem with web applications. In these attacks sort of attacks, malignant scripts are injected into web sites [110]. XSS attacks happen when an attacker tries to manipulate the browser side script so that the script works maliciously on server end [110]. This type of attack can happen unless the input from the user is validated or encoded [110].
  In the current implementation, the behavior extension for authentication does not considering XSS attacks and the AJAX calls can be manipulated by the user. To prevent this

problem the AJAX implementation should follow the rules explained in OWASP AJAX Security Guidelines [111].
Apart from these guidelines, a simple method to provide confidentiality for AJAX calls can be achieved by generating a token on the server side and storing it for later comparison. This token can be send with each request and it can be validated again at the server side. This token can also be a cookie to uniquely identify the user based on the information that the server has by checking the cookie data against a database.

- The current implementation does not specify **per user** behaviors. The behavior extension should be improved so that it can handle multiple behavior templates. Storing this data would be an issue if the number of users is large. For example, this might require use of a structured storage mechanism for maintainability and scalability reasons.

  The user behavior must uniquely identify the user. One way to achieve this identification would be an approach that differentiates users based upon different patterns and ordering/sequencing of the user actions. With this approach, the behavior extension might check a given user's behavior against a certain order of actions. Another approach would be to implementation counter patterns, i.e., disallowed patterns, rather permissible patterns.
  Furthermore, the behavior concept can be iterated by means of evaluating sequences. Such sequencing in behavior extension method would minimize FAR and FRR dramatically.
  The above improvements for the behavior extension authentication method can be augmented by making the method improve the behavior templates based upon continuous evaluation and with self-learning from previous user behaviors. Although there was no discrepancy in the experiment results, the propose behavior extension method might have vulnerabilities and might require more data (and testing) to assure that the implementation is deterministic.
  Machine learning and measurements provide a formal methodology for determining non-deterministic models or assessing a probabilistic density over the variables being referred to [112]. Inside of this generative probabilistic density, one can indicate partial prior knowledge and refine this coarse model utilizing experimental observations. With this approach, the behavior extension method from a rule-based approach to a probabilistic approach. In artificial intelligence, there has been a similar migration from rule-based expert systems to probabilistic generative models. Statistical machine learning techniques are currently being used in many areas, such as robotics, traditional dynamics and control systems, path planning, potential functions, and navigation models.

After fulfilling the above first two improvement, deployment of the behavior extension to another existing web application having SSO integration would be the next step.

In order to validate the measured rates, monitoring of the web application integrated with the behavior extension is important. Continuous monitoring might also be necessary to analyze the performance of the extension.

The entire experimental test bed is running under Windows 2008 together with IIS 7.5. The development environment was Windows 7-64 bit. The web application and the integrations were running without any problem (See also Section 5.2 for limitation of SCRAM). Another investigation effort would be to examine the possibility of integrating the proposed behavior extension method in

non-Microsoft environments. This future work needs to be investigated after it is proven that the behavior extension is a robust authentication method (i.e., does not produce unacceptable FAR and FRR rates in a wider range of test beds). If it is not possible to integrate the behavior extension method natively, then it might be still worth investigating the possibility to implement this method in different programming languages.

Another investigation and improvement would be to make it easier to integrate the proposed method with other code. There might be a room for further improvement of the behavior extension by checking the code against the goals set forth in MSDN to achieve security requirements [113]. The current code is easily traceable and debugging the extension would not be difficult for an experienced programmer. Apart from checking against this Microsoft reference, another improvement might be to evaluate providing helper methods so that another web application can use this behavior extension without any problems.

A further investigation would be to understand why the existing SCRAM authentication method in the web application is producing large outlier values at uneven time intervals and why it also shows very small values when the SCRAM authentication method is continuously evaluating different requests. These investigations might be relevant to the statistical analysis since I expect that the distributions of the control and experiment group in elapsed authentication time values would have the same distribution so that I could draw a conclusion that one method is faster or slower than the other one.

The existing web application does not have SSO integration due to the fact that it would be redundant to integrate SSO with the behavior extension authentication method (See section 3.5). However, for completeness, it would be good to examine this integration in future work.

During the experiments for elapsed authentication time, the total task time statistics were not analyzed due to the fact that the users might not fulfill the assigned task within the expected time and the total task time results might not be statistically relevant when evaluating authentication times. However, a more user behavior focused analysis would examine how much time the user is actually spending to carry out the assigned task. This information could be relevant for future.

In this thesis project, the behavior extension method was not evaluated in terms of being able to predict a future user behavior. However, this might be investigated and a time series analysis might be relevant future work.

In this thesis, I planned to use Student's T-test for data analysis. However, after a preliminary experiment, I saw that there are outlier values that would affect the distribution. If the root cause of outlier values were removed, then the Student T-test could potentially be applied to the data set.

Another statistical analysis that could be done concerns the user's learning process. During the FAR and FRR evaluation of legitimate and illegitimate user groups, I collapsed those experiments so that the same user participated first as an illegitimate user, and then as a legitimate user. A McNemar Test might be applied to the collected data to see whether FAR and FRR rates change significantly [87].

As a future work, a qualitative analysis could be conducted on those users that are using the behavior extension and those administrators who are administering the behavior templates and monitoring the behavior extension. This might be done by coding interviews, performing an online survey, or providing an online service that users can use to provide their opinions about the authentication methods. This could be contextualized to investigate how suitable the behavior extension is for end users.

Given the data, analysis, and results acquired from the experiments, it might be relevant to describe to an organization's security management how the behavior extension authentication

method can augment the existing infrastructure by fulfilling the CIA requirements of an organization.

Finally, it is also relevant to describe to business management how this method will have an impact on business and how costs could be reduce by implementing/integrating this solution. This would be done with a thorough project planning, scheduling and resource planning tasks.

5.4 Required reflections

Some of the ethical issues that should be considered when planning data collection involving human participants were discussed in Section 3.8.

The proposed behavior extension has several potential impacts on the economics of SSO. If the proposed method has suitable FAR and FRR rates, then it will provide efficiency and effectivity in an organization and have an impact on the organization in terms of resource savings.

References

[1] Donald Eastlake 3rd, 'Domain Name System Security Extensions', *Internet Request for Comments*, vol. RFC 2535 (Proposed Standard), Mar. 1999 [Online]. Available: http://www.rfc-editor.org/rfc/rfc2535.txt

[2] S. Naqvi and M. Riguidel, 'Security architecture for heterogeneous distributed computing systems', presented at the 38th Annual International Carnahan Conference on Security Technology, 2004, pp. 34–41 [Online]. DOI: 10.1109/CCST.2004.1405366

[3] Jongil Jeong, Dongkyoo Shin, Dongil Shin, and Kiyoung Moon, 'Java-Based Single Sign-On Library Supporting SAML (Security Assertion Markup Language) for Distributed Web Services', in *Advanced Web Technologies and Applications*, vol. 3007, J. X. Yu, X. Lin, H. Lu, and Y. Zhang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 891–894 [Online]. Available: http://link.springer.com/10.1007/978-3-540-24655-8_99. [Accessed: 16-Jul-2015]

[4] Ken Peffers, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee, 'A Design Science Research Methodology for Information Systems Research', *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 2007. DOI: 10.2753/MIS0742-1222240302

[5] Johannes A. Buchmann, Evangelos Karatsiolis, and Alexander Wiesmaier, *Introduction to public key infrastructures*. New York: Springer, 2013, ISBN: 978-3-642-40656-0.

[6] Carlisle Adams, Steve Lloyd, and Carlisle Adams, *Understanding PKI: concepts, standards, and deployment considerations*. Boston: Addison-Wesley, 2003, ISBN: 978-0-672-32391-1.

[7] Suranjan Choudhury, Kartik Bhatnagar, and Wasim Haque, *Public key infrastructure: implementation and design*. New York, NY: M&T Books, 2002, ISBN: 978-0-7645-4879-6.

[8] Alper E. Yegin and Fujio Watanabe, 'Authentication, Authorization, and Accounting', in *Next Generation Mobile Systems 3G and Beyond*, M. Etoh, Ed. Chichester, UK: John Wiley & Sons, Ltd, 2005, pp. 315–343 [Online]. Available: http://doi.wiley.com/10.1002/0470091533.ch11. [Accessed: 16-Jul-2015]

[9] M. A. Sasse, S. Brostoff, and D. Weirich, 'Transforming the "Weakest Link" — a Human/Computer Interaction Approach to Usable and Effective Security', *BT Technology Journal*, vol. 19, no. 3, pp. 122–131, Jul. 2001. DOI: 10.1023/A:1011902718709

[10] C. Finseth, 'An Access Control Protocol, Sometimes Called TACACS', *Internet Request for Comments*, vol. RFC 1492 (Informational), Jul. 1993 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1492.txt

[11] C. Rigney, S. Willens, A. Rubens, and W. Simpson, 'Remote Authentication Dial In User Service (RADIUS)', *Internet Request for Comments*, vol. RFC 2865 (Draft Standard), Jun. 2000 [Online]. Available: http://www.rfc-editor.org/rfc/rfc2865.txt

[12] Joshua Hill, 'An Analysis of the RADIUS Authentication Protocol', 24-Nov-2001. [Online]. Available: http://www.untruth.org/~josh/security/radius/radius-auth.html. [Accessed: 16-Jul-2015]

[13] Eric Rescorla, Gregory Lebovitz, and Internet Architecture Board, 'A Survey of Authentication Mechanisms', IETF Network Working Group, Internet-Draft draft-iab-auth-mech-07.txt, Feb. 2010 [Online]. Available: https://tools.ietf.org/group/iab/draft-iab-auth-mech/draft-iab-auth-mech-07-from-06.diff.txt. [Accessed: 16-Jul-2015]

[14] Yash Kedia, Amit Agrawal, and K. Chandrasekaran, 'Securing Single Sign-On Mechanism', *IJACET*, vol. 2, no. 2, pp. 17–21, Mar. 2015.

[15] Kevin White, 'Which Identities Are We Using to Sign in Around the Web? [INFOGRAPHIC]', *Gigya Inc.* [Online]. Available: http://www.gigya.com/blog/which-identities-are-we-using-to-sign-in-around-the-web-infographic/. [Accessed: 09-Jun-2015]

[16] Shakir James, 'Web Single Sign-On Systems', Dec-2007. [Online]. Available: http://www.cs.wustl.edu/~jain/cse571-07/ftp/websso/#sec1.1. [Accessed: 09-Jun-2015]

[17] Jan de Clercq and Guido Grillenmeier, *Microsoft windows security fundamentals*. Burlington, MA: Elsevier Digital Press, 2007, ISBN: 1-55558-340-7.

[18] InfraSec 2002, George Davida, Yair Frankel, and Owen Rees, *Infrastructure security international conference, InfraSec 2002, Bristol, UK, October 1-3, 2002: proceedings*. Berlin; New York: Springer, 2002, ISBN: 978-3-540-45831-9 [Online]. Available: http://rave.ohiolink.edu/ebooks/ebc/354045831X. [Accessed: 16-Jul-2015]

[19] Frederick Hirsch, Rob Philpott, and Eve Maler, 'Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0', OASIS, OASIS Standard saml-sec-consider-2.0-os, Mar. 2005 [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf

[20] Tibor Jager and Juraj Somorovsky, 'How to break XML encryption', in *Proceedings of the 18th ACM conference on Computer and communications security CCS '11*, New York, NY, USA, 2011, pp. 413–422 [Online]. DOI: 10.1145/2046707.2046756

[21] OASIS, 'OASIS Security Services (SAML) TC', Technical Committee website [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security. [Accessed: 09-Jun-2015]

[22] Nick Ragouzis, John Hughes, Rob Philpott, Eve Maler, Paul Madsen, and Tom Scavo, 'sstc-saml-tech-overview-2.0.pdf'. [Online]. Available: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.pdf. [Accessed: 16-Jul-2015]

[23] San-Tsai Sun, Eric Pospisil, Ildar Muslukhov, Nuray Dindar, Kirstie Hawkey, and Konstantin Beznosov, 'Investigating Users' Perspectives of Web Single Sign-On: Conceptual Gaps and Acceptance Model', *ACM Transactions on Internet Technology*, vol. 13, no. 1, pp. 1–35, Nov. 2013. DOI: 10.1145/2532639

[24] R. Housley, W. Polk, W. Ford, and D. Solo, 'Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile', *Internet Request for Comments*, vol. RFC 3280 (Proposed Standard), Apr. 2002 [Online]. Available: http://www.rfc-editor.org/rfc/rfc3280.txt

[25] B. Campbell, C. Mortimore, M. Jones, and Y. Goland, 'Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants', *Internet Request for Comments*, vol. RFC 7521 (Proposed Standard), May 2015 [Online]. Available: http://www.rfc-editor.org/rfc/rfc7521.txt

[26] Microsoft, '[MS-SAMLPR]: Element <StatusCode>', *Microsoft Developer Network*. [Online]. Available: https://msdn.microsoft.com/en-us/library/hh269642.aspx. [Accessed: 16-Jul-2015]

[27] Gartner, Inc., 'Web Access Management (WAM) - Gartner IT Glossary'. [Online]. Available: http://www.gartner.com/it-glossary/wam-web-access-management. [Accessed: 09-Jun-2015]

[28] 'Shibboleth: Federated Single Sign-On Authentication Service | Unicon'. [Online]. Available: https://www.unicon.net/opensource/shibboleth. [Accessed: 23-Aug-2015]

[29] Shibboleth Consortium, 'How Shibboleth Works'. [Online]. Available: https://shibboleth.net/about/basic.html. [Accessed: 09-Jun-2015]

[30] Marco De Marco, Dov Te'eni, Valentina Albano, and Stefano Za, Eds., *Information systems: crossroads for organization, management, accounting and engineering: ItAIS: The Italian Association for Information Systems*, Softcover repr. of the hardcover 1st. ed. 2012. Berlin: Physica-Verl, 2014, ISBN: 978-3-7908-2789-7.

[31] Mark Stamp, *Information security: principles and practice*, 2nd ed. Hoboken, NJ: Wiley, 2011, ISBN: 978-0-470-62639-9.

[32] Ross Anderson, *Security engineering: a guide to building dependable distributed systems*, 2nd ed. Indianapolis, IN: Wiley Pub, 2008, ISBN: 978-0-470-06852-6.

[33] F. Zhang, R. Jing, and R. Gandhi, 'RSVP-TE Extensions for Associated Bidirectional Label Switched Paths (LSPs)', *Internet Request for Comments*, vol. RFC 7551 (Proposed Standard), May 2015 [Online]. Available: http://www.rfc-editor.org/rfc/rfc7551.txt

[34] Sonia Chiasson, P. C. van Oorschot, and Robert Biddle, 'A Usability Study and Critique of Two Password Managers', in *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, Berkeley, CA, USA, 2006 [Online]. Available: http://dl.acm.org/citation.cfm?id=1267336.1267337

[35] M. Bishop, 'Password management', in *Digest of Papers Compcon Spring '91.*, 1991, pp. 167–169 [Online]. DOI: 10.1109/CMPCON.1991.128801

[36] Shirley Gaw and Edward W. Felten, 'Password management strategies for online accounts', in *Proceedings of the second symposium on Usable privacy and security SOUPS '06*, New York, NY, USA, 2006, pp. 44–55 [Online]. DOI: 10.1145/1143120.1143127

[37] J. Alex Halderman, Brent Waters, and Edward W. Felten, 'A convenient method for securely managing passwords', in *Proceedings of the 14th international conference on World Wide Web WWW '05*, New York, NY, USA, 2005, pp. 471–479 [Online]. DOI: 10.1145/1060745.1060815

[38] Ka-Ping Yee and Kragen Sitaker, 'Passpet: convenient password management and phishing protection', in *Proceedings of the second symposium on Usable privacy and security SOUPS '06*, New York, NY, USA, 2006, p. 32 [Online]. DOI: 10.1145/1143120.1143126

[39] Matt Bishop and Daniel V. Klein, 'Improving system security via proactive password checking', *Computers & Security*, vol. 14, no. 3, pp. 233–249, Jan. 1995. DOI: 10.1016/0167-4048(95)00003-Q

[40] J. Ortega-Garcia, J. Bigun, D. Reynolds, and J. Gonzalez-Rodriguez, 'Authentication gets personal with biometrics', *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 50–62, Mar. 2004. DOI: 10.1109/MSP.2004.1276113

[41] Umut Uludag, 'Secure Biometric Systems', Doctoral dissertation, Michigan State University, Computer Science & Engineerin, East Lansing, Michigan, USA, 2006 [Online]. Available: http://biometrics.cse.msu.edu/Publications/Thesis/Reserved/UmutUludag_SecureBiometrics_PhD06.pdf. [Accessed: 16-Jul-2015]

[42] Yan Sui, Xukai Zou, Eliza Y. Du, and Feng Li, 'Secure and privacy-preserving biometrics based active authentication', presented at the IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2012, Seoul, Korea, 2012, pp. 1291–1296 [Online]. DOI: 10.1109/ICSMC.2012.6377911

[43] Geoff Duncan, 'Why haven't biometrics replaced passwords yet? | Digital Trends', 09-Mar-2013. [Online]. Available: http://www.digitaltrends.com/computing/can-biometrics-secure-our-digital-lives/. [Accessed: 09-Jun-2015]

[44] Simson Garfinkel and Heather Richter Lipford, 'Usable Security: History, Themes, and Challenges', *Synthesis Lectures on Information Security, Privacy, and Trust*, vol. 5, no. 2, pp. 1–124, Sep. 2014. DOI: 10.2200/S00594ED1V01Y201408SPT011

[45] Parekh Tanvi, Gawshinde Sonal, and Sharma Mayank Kumar, 'Token Based Authentication Using Mobile Phone', 2011, pp. 85–88 [Online]. DOI: 10.1109/CSNT.2011.24

[46] Stephen Lawton, 'Single Sign-On (SSO) Solutions: Advantages and Challenges', 06-Jan-2015. [Online]. Available: http://www.tomsitpro.com/articles/single-sign-on-solutions,2-853.html. [Accessed: 09-Jun-2015]

[47] W. Simpson, 'PPP Challenge Handshake Authentication Protocol (CHAP)', *Internet Request for Comments*, vol. RFC 1994 (Draft Standard), Aug. 1996 [Online]. Available: http://www.rfc-editor.org/rfc/rfc1994.txt

[48] Myriam Abramson and David Aha, 'User Authentication from Web Browsing Behavior - 6081'. [Online]. Available: https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS13/paper/viewFile/5865/6081. [Accessed: 23-Aug-2015]

[49] Sungzoon Cho and Jang Min, 'System and method for performing user authentication based on user behavior patterns', [Online]. Available: http://www.google.com/patents/US20070236330. [Accessed: 23-Aug-2015]

[50] Elaine Shi, Yuan Niu, Markus Jakobsson, and Richard Chow, 'Implicit Authentication through Learning User Behavior'. [Online]. Available: https://www.cs.umd.edu/~elaine/docs/isc.pdf. [Accessed: 23-Aug-2015]

[51] Kenneth Revett, 'A bioinformatics based approach to user authentication via keystroke dynamics', *International Journal of Control, Automation and Systems*, vol. 7, no. 1, pp. 7–15, Feb. 2009. DOI: 10.1007/s12555-009-0102-2

[52] Naveen Sastry, Umesh Shankar, and David Wagner, 'Secure verification of location claims', in *Proceedings of the 2nd ACM workshop on Wireless security iSe '03*, San Diego, CA, USA, 2003, pp. 1–10 [Online]. DOI: 10.1145/941311.941313

[53] Tao Feng, Ziyi Liu, Kyeong-An Kwon, Weidong Shi, Bogdan Carbunar, Yifei Jiang, and Nhung Nguyen, 'Continuous mobile authentication using touchscreen gestures', presented at the IEEE Conference on Technologies for Homeland Security (HST), 2012, Waltham, MA, USA, 2012, pp. 451–456 [Online]. DOI: 10.1109/THS.2012.6459891

[54] Alan Hevner, Salvatore March, and Jinsoo Park, 'Design Science in Information Systems Research', *MIS Quartely*, vol. 28, no. 1, pp. 75–105, Mar. 2004.

[55] Herbert A Simon, *The sciences of the artificial*. Cambridge, Mass.: MIT Press, 1996, ISBN: 0-585-36010-3 [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=49230. [Accessed: 12-Sep-2015]

[56] Derek E. Brink, 'The Many Faces of Single Sign-On', Aberdeen Group, Boston, MA, USA, White paper 010908a, Jun. 2008 [Online]. Available: https://www.meritalk.com/uploads_legacy/whitepapers/ManyFacesofSingleSignOn.pdf. [Accessed: 12-Sep-2015]

[57] Uwe Flick, *An introduction to qualitative research*, 4. ed., repr. Los Angeles, Calif.: SAGE, 2011, ISBN: 978-1-84787-324-8.

[58] R. Barker Bausell and Yu-Fang Li, *Power Analysis for Experimental Research: A Practical Guide for the Biological, Medical and Social Sciences*. Cambridge: Cambridge University Press, 2002, ISBN: 978-0-511-54193-3 [Online]. Available: http://ebooks.cambridge.org/ref/id/CBO9780511541933. [Accessed: 12-Sep-2015]

[59] Frederick J. Dorey, 'In Brief: Statistics in Brief: Confidence Intervals: What is the Real Result in the Target Population?', *Clinical Orthopaedics and Related Research®*, vol. 468, no. 11, pp. 3137–3138, Nov. 2010. DOI: 10.1007/s11999-010-1407-4

[60] Douglas W. Hubbard, *The failure of risk management: why it's broken and how to fix it*. Hoboken, N.J: Wiley, 2009, ISBN: 978-0-470-38795-5.

[61] Detmar W. Straub, Ed., *Information security: policy, processes and practices*. Armonk, NY: Sharpe, 2008, ISBN: 978-0-7656-1718-7.
[62] Alberts, Dorofee, Stevens, and Woody, *Alberts, C., Dorofee, A., Stevens, J. and Woody, C. (2003) Introduction to the Octave Approach*. 2003.
[63] 'ISO/IEC 27005:2011 - Information technology -- Security techniques -- Information security risk management'. [Online]. Available: http://www.iso.org/iso/catalogue_detail?csnumber=56742. [Accessed: 28-Jan-2016]
[64] Thomas R. Peltier, 'Facilitated Risk Analysis Process (FRAP)', Auerbach Publications, CRC Press LLC, Auerback Information Management Service 85-01-21, Nov. 2000 [Online]. Available: http://www.ittoday.info/AIMS/DSM/85-01-21.pdf. [Accessed: 28-Jan-2016]
[65] ISACA, 'COBIT 5: A Business Framework for the Governance and Management of Enterprise IT'. [Online]. Available: http://www.isaca.org/Cobit/pages/default.aspx. [Accessed: 28-Jan-2016]
[66] Sokratis K. Katsikas, International Federation for Information Processing, and International Conference on Information Security, Eds., *Information systems security: facing the information society of the 21st century*, 1. ed. London: Chapman & Hall, 1996, ISBN: 0-412-78120-4.
[67] The Institute of Internal Auditors, 'Common Internal Control Frameworks', 21-Nov-2008. [Online]. Available: http://www.theiia.org/intAuditor/media/images/Burch_dec'08_artok_cx.pdf. [Accessed: 28-Jan-2016]
[68] Bernie J. O'Brien, 'Book Review Risk–Benefit Analysis By Richard Wilson and Edmund A.C. Crouch. 370 pp., illustrated. Cambridge, Mass., Harvard University Press, 2001. $25. 0-674-00529-5', *New England Journal of Medicine*, vol. 346, no. 14, pp. 1099–1100, Apr. 2002. DOI: 10.1056/NEJM200204043461421
[69] Klaas Wierenga, Eliot Lear, and Simon Josefsson, 'A Simple Authentication and Security Layer (SASL) and GSS-API Mechanism for the Security Assertion Markup Language (SAML)'. [Online]. Available: https://tools.ietf.org/html/rfc6595. [Accessed: 16-Sep-2015]
[70] C. Newman, A. Menon-Sen, A. Melnikov, and N. Williams, 'Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms', *Internet Request for Comments*, vol. RFC 5802 (Proposed Standard), Jul. 2010 [Online]. Available: http://www.rfc-editor.org/rfc/rfc5802.txt
[71] 'Salted Challenge Response Authentication Mechanism (SCRAM) SHA-1 - CodeProject'. [Online]. Available: http://www.codeproject.com/Articles/698219/Salted-Challenge-Response-Authentication-Mechanism. [Accessed: 12-Sep-2015]
[72] Microsoft, 'ASP.NET Page Life Cycle Overview'. [Online]. Available: https://msdn.microsoft.com/en-us/library/ms178472.aspx. [Accessed: 16-Sep-2015]
[73] J. M. Christian Bastien and Dominique L. Scapin, 'Preliminary findings on the effectiveness of ergonomic criteria for the evaluation of human-computer interfaces', presented at the Conference Companion on Human Factors in Computing Systems CHI '93 INTERACT '93 and CHI '93, Amsterdam, The Netherlands, 1993, pp. 187–188 [Online]. DOI: 10.1145/259964.260198
[74] Donald A. Norman, *Emotional design: why we love (or hate) everyday things*. New York: Basic Books, 2005, ISBN: 978-0-465-05136-6.
[75] Beth E. Kolko and Robert Racadio, 'The Value of Non-Instrumental Computer Use: A Study of Skills Acquisition and Performance in Brazil', *Information Technologies & International Development*, vol. 10, no. 3, pp. pp. 47–65, Sep. 2014.

[76] Michael Humphreys and William Revelle, 'Personality, Motivation, and Performance', *Psychological Review*, vol. 91, no. 2, pp. 153–184, Apr. 1984.

[77] Tom Tullis and Bill Albert, *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Amsterdam ; Boston: Elsevier/Morgan Kaufmann, 2008, ISBN: 978-0-12-373558-4.

[78] hhu, 'GPowerManual.pdf'. [Online]. Available: http://www.gpower.hhu.de/fileadmin/redaktion/Fakultaeten/Mathematisch-Naturwissenschaftliche_Fakultaet/Psychologie/AAP/gpower/GPowerManual.pdf. [Accessed: 20-Feb-2016]

[79] Sarah Boslaugh and Paul A. Watters, *Statistics in a nutshell*. Farnham: O'Reilly, 2008, ISBN: 978-0-596-51049-7.

[80] Cosma Rohilla Shalizi, *Advanced Data Analysis from an Elementary Point of View*, Draft textbook for 36-402, Advanced Data Analysis. Pittsburgh, PA, USA: Carnegie Mellon University., 2016 [Online]. Available: http://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/ADAfaEPoV.pdf. [Accessed: 28-Jan-2016]

[81] Daniel Muijs, *Doing quantitative research in education with SPSS*. Los Angeles, Calif.; London: Sage, 2014, ISBN: 978-1-4462-8798-9.

[82] Atul Bansal, Ravinder Agarwal, and R. K. Sharma, 'FAR and FRR based analysis of iris recognition system', 2012, pp. 1–6 [Online]. DOI: 10.1109/ISPCC.2012.6224358

[83] ECM Europe, 'Risk Assessment in Ship Operations', Rome Harbour, Italy, 30-Mar-2010 [Online]. Available: http://www.ecmeurope.net/wp-content/uploads/2009/10/ECM-Europe-Risk-Management-Mar-2010.pdf. [Accessed: 28-Jan-2016]

[84] John R. Vacca, Ed., *Computer and information security handbook*. Amsterdam ; Boston : Burlington, MA: Elsevier ; Morgan Kaufmann, 2009, ISBN: 978-0-12-374354-1.

[85] Dawid Czagan, 'Quantitative Risk Analysis', *InfoSec Resources*, 04-Nov-2013. [Online]. Available: http://resources.infosecinstitute.com/quantitative-risk-analysis/. [Accessed: 28-Jan-2016]

[86] Sanjay Goel and Vicki Chen, 'Information Security Risk Analysis – A Matrix-Based Approach', in *Proceedings of the Information Resource Management Association (IRMA) International Conference*, San Diego, CA, USA, 2005, p. 9 [Online]. Available: http://www.albany.edu/~goel/publications/goelchen2005.pdf. [Accessed: 28-Jan-2016]

[87] Mohan Kamat, 'Microsoft Word - Matrices for Asset Valuation and Risk Analysis.doc - ISO27k_Matrices_for_Asset_Valuation_and_Risk_Analysis.pdf', ISO 27001 security, ISO 27001 Implementer's Forum ISMS/GL/004, Jul. 2009 [Online]. Available: http://www.iso27001security.com/ISO27k_Matrices_for_Asset_Valuation_and_Risk_Analysis.pdf. [Accessed: 28-Jan-2016]

[88] Microsoft, 'Visual Studio (ISO image)', *VS2010*. 2010 [Online]. Available: http://download.microsoft.com/download/2/4/7/24733615-AA11-42E9-8883-E28CDCA88ED5/X16-42552VS2010UltimTrial1.iso

[89] Franz Faul, Edgar Erdfelder, Albert-Georg Lang, and Axel Buchner, 'G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences', *Behavior Research Methods*, vol. 39, no. 2, pp. 175–191, May 2007. DOI: 10.3758/BF03193146

[90] Axel Buchner, Edgar Erdfelder, Franz Faul, and Albert-Georg Lang, 'G*Power: Statistical Power Analyses for Windows and Mac', *Heinrich-Heine-Universität Düsseldorf, Allgemeine Psychologie und Arbeitspsychologie*. [Online]. Available: http://www.gpower.hhu.de/. [Accessed: 11-Oct-2015]

[91] 'IBM SPSS software'. [Online]. Available: http://www-01.ibm.com/software/analytics/spss/. [Accessed: 28-Dec-2015]
[92] 'IBM trial download : SPSS Statistics Desktop 22.0'. [Online]. Available: https://www-01.ibm.com/marketing/iwm/iwmdocs/tnd/data/web/en_US/trialprograms/W110742E06714B29.html. [Accessed: 20-Feb-2016]
[93] 'IBM SPSS Statistics 22 Documentation - United States'. [Online]. Available: http://www-01.ibm.com/support/docview.wss?uid=swg27038407. [Accessed: 28-Dec-2015]
[94] 'Microsoft Support Lifecycle'. [Online]. Available: https://support.microsoft.com/en-us/lifecycle/search/default.aspx?sort=PN&alpha=.NET%20Framework&Filter=FilterNO. [Accessed: 28-Jan-2016]
[95] 'VMWare downloads'. [Online]. Available: https://my.vmware.com/web/vmware/downloads. [Accessed: 12-Sep-2015]
[96] 'Microsoft DreamSpark - Software Catalog'. [Online]. Available: https://www.dreamspark.com/Student/Software-Catalog.aspx. [Accessed: 20-Feb-2016]
[97] Jonathan Stone, Michael Greenwald, Craig Partridge, and James Hughes, 'Performance of Checksums and CRC's over Real Data', *IEEE/ACM Trans. Netw.*, vol. 6, no. 5, pp. 529–543, Oct. 1998. DOI: 10.1109/90.731187
[98] 'SAML SSO for ASP.NET, C#, VB.NET - ComponentSpace'. [Online]. Available: http://www.componentspace.com/. [Accessed: 23-Aug-2015]
[99] 'How to Use SPSS: Mann-Whitney U Test - YouTube'. [Online]. Available: https://www.youtube.com/watch?v=3r5xR16x5Dk. [Accessed: 20-Feb-2016]
[100] how2stats, *Mann-Whitney U - SPSS (part 2)*. 2011 [Online]. Available: https://www.youtube.com/watch?v=sEOnGMmQJoM. [Accessed: 20-Feb-2016]
[101] stikpet, *SPSS Grouped frequency table*. 2014 [Online]. Available: https://www.youtube.com/watch?v=4XQxHYJHERs. [Accessed: 20-Feb-2016]
[102] LearnByVideo, *SPSS Skill #27 : Creating Interval Data using Visual Binning*. 2012 [Online]. Available: https://www.youtube.com/watch?v=jj48sD9CW-o. [Accessed: 20-Feb-2016]
[103] Todd Grande, *Mann-Whitney U Test in SPSS*. 2015 [Online]. Available: https://www.youtube.com/watch?v=rMom1mAbPiU. [Accessed: 20-Feb-2016]
[104] 'GraphPad Statistics Guide'. [Online]. Available: http://www.graphpad.com/guides/prism/6/statistics/index.htm?how_the_mann-whitney_test_works.htm. [Accessed: 20-Feb-2016]
[105] Todd Grande, *Sensitivity, Specificity, False Positives, and False Negatives in SPSS*. 2015 [Online]. Available: https://www.youtube.com/watch?v=XdPAjO718Fg. [Accessed: 20-Feb-2016]
[106] Jason Popan, *Chi-square Goodness-of-fit Test in SPSS*. 2012 [Online]. Available: https://www.youtube.com/watch?v=nITYifnU_6o. [Accessed: 20-Feb-2016]
[107] BrunelASK, *Chi-square test in SPSS + interpretation*. 2013 [Online]. Available: https://www.youtube.com/watch?v=wfIfEWMJY3s. [Accessed: 21-Feb-2016]
[108] 'Chi Square Statistics'. [Online]. Available: http://math.hws.edu/javamath/ryan/ChiSquare.html. [Accessed: 21-Feb-2016]
[109] 'SuperSocket, an extensible socket server framework', *CodePlex*. [Online]. Available: https://supersocket.codeplex.com/Wikipage?ProjectName=supersocket. [Accessed: 28-Jan-2016]
[110] 'Cross-site Scripting (XSS) - OWASP'. [Online]. Available: https://www.owasp.org/index.php/XSS. [Accessed: 21-Feb-2016]

[111] 'OWASP AJAX Security Guidelines - OWASP'. [Online]. Available: https://www.owasp.org/index.php/OWASP_AJAX_Security_Guidelines. [Accessed: 28-Jan-2016]

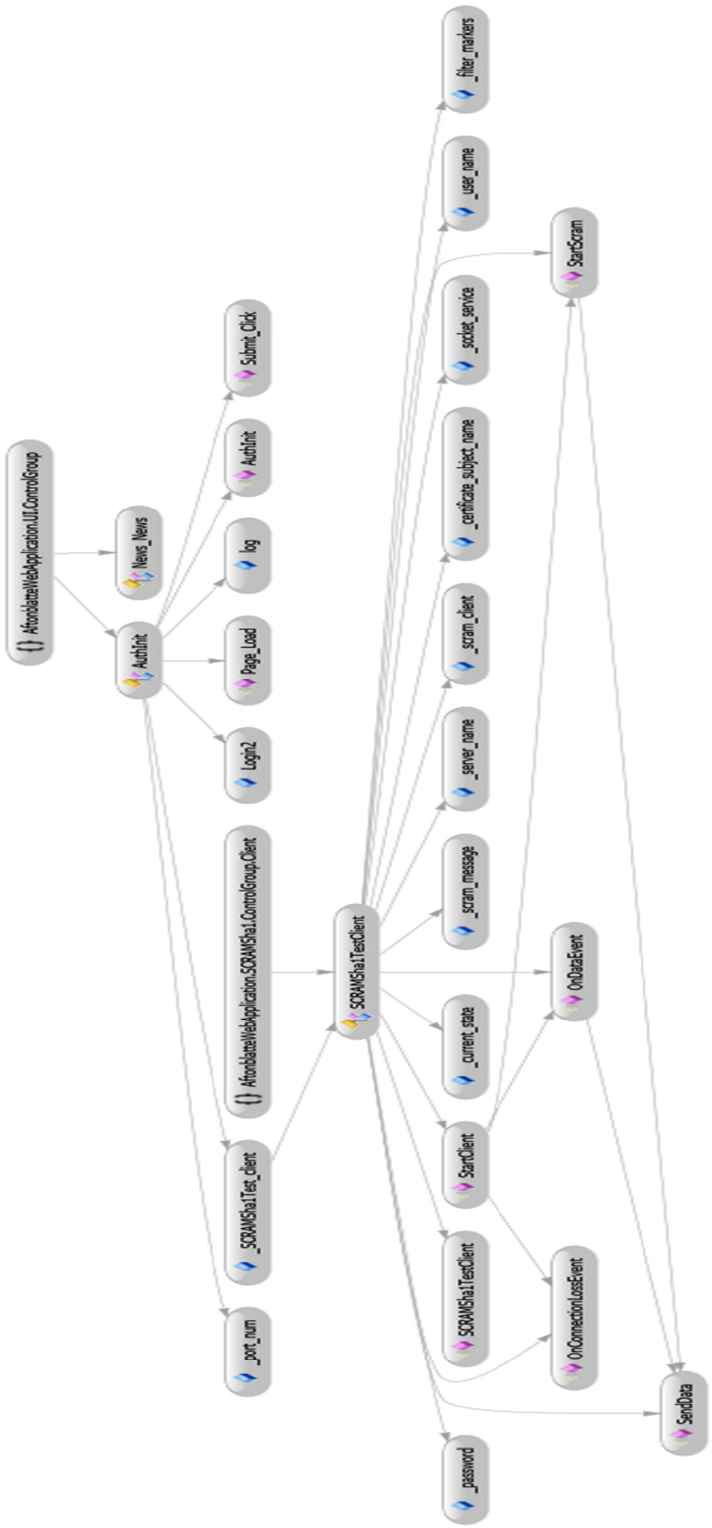
[112] Tony Jebara, *Machine Learning*. Boston, MA: Springer US, 2004, ISBN: 978-1-4613-4756-9 [Online]. Available: http://link.springer.com/10.1007/978-1-4419-9011-2. [Accessed: 28-Jan-2016]

[113] Microsoft, 'Code analysis rule set reference: Visual Studio 2015', *Microsoft Developer Network*. [Online]. Available: https://msdn.microsoft.com/en-us/library/dd264925.aspx. [Accessed: 28-Jan-2016]

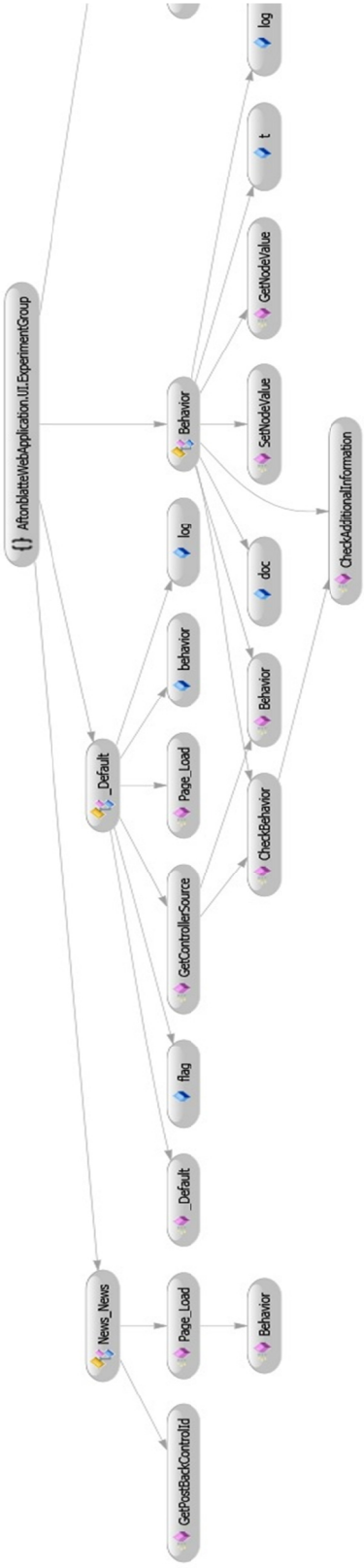
[114] 'Critical_Values_of_the_Chi-Squared_Distribution.doc - Critical_Values_of_the_Chi-Squared_Distribution.pdf'. [Online]. Available: https://courses.physics.illinois.edu/phys598aem/Software/Critical_Values_of_the_Chi-Squared_Distribution.pdf. [Accessed: 01-Mar-2016]

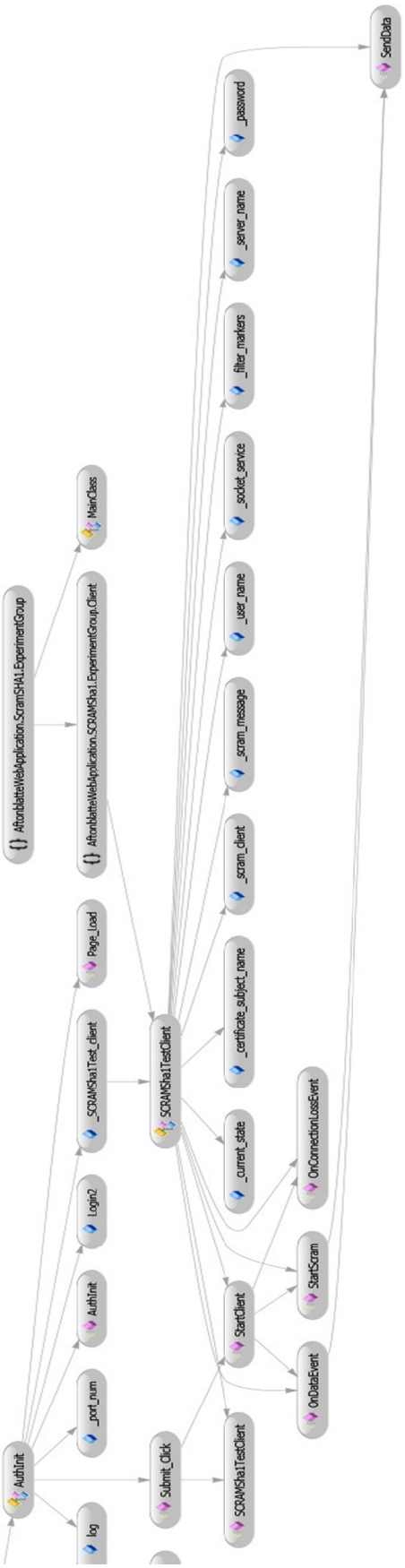
[115] '1.3.6.7.1. Cumulative Distribution Function of the Standard Normal Distribution'. [Online]. Available: http://www.itl.nist.gov/div898/handbook/eda/section3/eda3671.htma. [Accessed: 01-Mar-2016]

Appendix A: Class diagram of the implementation (Elapsed Authentication Time - Control Group)



Appendix B: Class diagram of the implementation (Elapsed Authentication Time - Experiment Group)




SendData
_password
_server_name
_filter_markers
_socket_service
_user_name
_scram_message
_scram_client
_certificate_subject_name
_current_state
OnConnectionLossEvent
StartScram
OnDataEvent
StartClient
SCRAMSha1TestClient
Submit_Click
AuthInit
log
_port_num
AuthInit
Login2
_SCRAMSha1Test_client
Page_Load
SCRAMSha1TestClient
AfterAbbateWebApplication.SCRAMSha1.ExperimentGroup_Client
MainClass
AfterAbbateWebApplication.ScramSHA1.ExperimentGroup

Appendix C: Critical values for chi-square distribution

Upper critical values of chi-square distribution with v degrees of freedom (based upon [114])

v	0.1	0.05	0.025	0.01	0.001
1	2.706	3.841	5.024	6.635	10.828
2	4.605	5.991	7.378	9.210	13.816
3	6.251	7.815	9.348	11.345	16.266
4	7.779	9.488	11.143	13.277	18.467
5	9.236	11.070	12.833	15.086	20.515
6	10.645	12.592	14.449	16.812	22.458
7	12.017	14.067	16.013	18.475	24.322
8	13.362	15.507	17.535	20.090	26.125
9	14.684	16.919	19.023	21.666	27.877
10	15.987	18.307	20.483	23.209	29.588
11	17.275	19.675	21.920	24.725	31.264
12	18.549	21.026	23.337	26.217	32.910
13	19.812	22.362	24.736	27.688	34.528
14	21.064	23.685	26.119	29.141	36.123
15	22.307	24.996	27.488	30.578	37.697
16	23.542	26.296	28.845	32.000	39.252
17	24.769	27.587	30.191	33.409	40.790
18	25.989	28.869	31.526	34.805	42.312
19	27.204	30.144	32.852	36.191	43.820
20	28.412	31.410	34.170	37.566	45.315
21	29.615	32.671	35.479	38.932	46.797
22	30.813	33.924	36.781	40.289	48.268
23	32.007	35.172	38.076	41.638	49.728
24	33.196	36.415	39.364	42.980	51.179
25	34.382	37.652	40.646	44.314	52.620
26	35.563	38.885	41.923	45.642	54.052
27	36.741	40.113	43.195	46.963	55.476
28	37.916	41.337	44.461	48.278	56.892
29	39.087	42.557	45.722	49.588	58.301
30	40.256	43.773	46.979	50.892	59.703
31	41.422	44.985	48.232	52.191	61.098
32	42.585	46.194	49.480	53.486	62.487
33	43.745	47.400	50.725	54.776	63.870
34	44.903	48.602	51.966	56.061	65.247
35	46.059	49.802	53.203	57.342	66.619
36	47.212	50.998	54.437	58.619	67.985
37	48.363	52.192	55.668	59.893	69.347
38	49.513	53.384	56.896	61.162	70.703
39	50.660	54.572	58.120	62.428	72.055
40	51.805	55.758	59.342	63.691	73.402

Appendix D: Area under the Normal Curve from 0 to X

This table is based upon [115].

X	0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.000	0.004	0.008	0.012	0.016	0.01994	0.02392	0.0279	0.03188	0.03586
0.1	0.040	0.044	0.048	0.052	0.056	0.05962	0.06356	0.06749	0.07142	0.07535
0.2	0.079	0.083	0.087	0.091	0.095	0.09871	0.10257	0.10642	0.11026	0.11409
0.3	0.118	0.122	0.126	0.129	0.133	0.13683	0.14058	0.14431	0.14803	0.15173
0.4	0.155	0.159	0.163	0.166	0.170	0.17364	0.17724	0.18082	0.18439	0.18793
0.5	0.191	0.195	0.198	0.202	0.205	0.20884	0.21226	0.21566	0.21904	0.2224
0.6	0.226	0.229	0.232	0.236	0.239	0.24215	0.24537	0.24857	0.25175	0.2549
0.7	0.258	0.261	0.264	0.267	0.270	0.27337	0.27637	0.27935	0.2823	0.28524
0.8	0.288	0.291	0.294	0.297	0.300	0.30234	0.30511	0.30785	0.31057	0.31327
0.9	0.316	0.319	0.321	0.324	0.326	0.32894	0.33147	0.33398	0.33646	0.33891
1.0	0.341	0.344	0.346	0.348	0.351	0.35314	0.35543	0.35769	0.35993	0.36214
1.1	0.364	0.367	0.369	0.371	0.373	0.37493	0.37698	0.379	0.381	0.38298
1.2	0.385	0.387	0.389	0.391	0.393	0.39435	0.39617	0.39796	0.39973	0.40147
1.3	0.403	0.405	0.407	0.408	0.410	0.41149	0.41308	0.41466	0.41621	0.41774
1.4	0.419	0.421	0.422	0.424	0.425	0.42647	0.42785	0.42922	0.43056	0.43189
1.5	0.433	0.434	0.436	0.437	0.438	0.43943	0.44062	0.44179	0.44295	0.44408
1.6	0.445	0.446	0.447	0.448	0.450	0.45053	0.45154	0.45254	0.45352	0.45449
1.7	0.455	0.456	0.457	0.458	0.459	0.45994	0.4608	0.46164	0.46246	0.46327
1.8	0.464	0.465	0.466	0.466	0.467	0.46784	0.46856	0.46926	0.46995	0.47062
1.9	0.471	0.472	0.473	0.473	0.474	0.47441	0.475	0.47558	0.47615	0.4767
2.0	0.477	0.478	0.478	0.479	0.479	0.47982	0.4803	0.48077	0.48124	0.48169
2.1	0.482	0.483	0.483	0.483	0.484	0.48422	0.48461	0.485	0.48537	0.48574
2.2	0.486	0.486	0.487	0.487	0.487	0.48778	0.48809	0.4884	0.4887	0.48899
2.3	0.489	0.490	0.490	0.490	0.490	0.49061	0.49086	0.49111	0.49134	0.49158
2.4	0.492	0.492	0.492	0.492	0.493	0.49286	0.49305	0.49324	0.49343	0.49361
2.5	0.494	0.494	0.494	0.494	0.494	0.49461	0.49477	0.49492	0.49506	0.4952
2.6	0.495	0.495	0.496	0.496	0.496	0.49598	0.49609	0.49621	0.49632	0.49643
2.7	0.497	0.497	0.497	0.497	0.497	0.49702	0.49711	0.4972	0.49728	0.49736
2.8	0.497	0.498	0.498	0.498	0.498	0.49781	0.49788	0.49795	0.49801	0.49807
2.9	0.498	0.498	0.498	0.498	0.498	0.49841	0.49846	0.49851	0.49856	0.49861
3.0	0.499	0.499	0.499	0.499	0.499	0.49886	0.49889	0.49893	0.49896	0.499
3.1	0.499	0.499	0.499	0.499	0.499	0.49918	0.49921	0.49924	0.49926	0.49929
3.2	0.499	0.499	0.499	0.499	0.499	0.49942	0.49944	0.49946	0.49948	0.4995
3.3	0.500	0.500	0.500	0.500	0.500	0.4996	0.49961	0.49962	0.49964	0.49965
3.4	0.500	0.500	0.500	0.500	0.500	0.49972	0.49973	0.49974	0.49975	0.49976
3.5	0.500	0.500	0.500	0.500	0.500	0.49981	0.49981	0.49982	0.49983	0.49983
3.6	0.500	0.500	0.500	0.500	0.500	0.49987	0.49987	0.49988	0.49988	0.49989
3.7	0.49989	0.4999	0.4999	0.4999	0.49991	0.49991	0.49992	0.49992	0.49992	0.49992
3.8	0.49993	0.49993	0.49993	0.49994	0.49994	0.49994	0.49994	0.49995	0.49995	0.49995

3.9	0.49995	0.49995	0.49996	0.49996	0.49996	0.49996	0.49996	0.49996	0.49997	0.49997
4.0	0.49997	0.49997	0.49997	0.49997	0.49997	0.49997	0.49998	0.49998	0.49998	0.49998

Appendix E: Experiment Data

Experiments_all.xlsx

TRITA-ICT-EX-2016:14