NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

NATIONAL AUTHENTICATION FRAMEWORK IMPLEMENTATION STUDY

by

Mok Chuan-Hao

December 2009

Thesis Co-Advisors: Bert Lundy
J. D. Fulp

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE National Authentication Framework Implementation Study		**5. FUNDING NUMBERS** **NA**	
6. AUTHOR(S) Mok Chuan-Hao			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

The move towards e-government has seen many institutions put special focus on the need for security, especially that of authentication. Single-factor password-based systems have been proven inadequate in safeguarding online financial and e-government service transactions. Industry adoption of Two-Factor Authentication (2FA) has also been piecemeal. To mitigate these deficiencies, the Singapore Government, in 2008, put forth a Call-for-Collaboration (CFC) seeking industry and academic participation in defining a National Authentication Framework (NAF), with the dual aim of providing for a national-level 2FA system and broadening the market for authentication services, and, in so doing, provide the user with a better authentication experience. This thesis will detail, discuss, and compare the various token types and identity frameworks (PKI, SAML, WS-F, OpenID, and Infocard) that make up an authentication system, and make recommendations on the best combination of technologies, protocols, and standards that, when implemented, would not only fulfill the requirements of the CFC, but also position it well for future enhancement.

14. SUBJECT TERMS Authentication, Identity, OpenID, Infocard, SAML, WS-Federation, PKI, National Authentication Framework.			**15. NUMBER OF PAGES** 85
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

NATIONAL AUTHENTICATION FRAMEWORK IMPLEMENTATION STUDY

Mok Chuan-Hao
Captain, Singapore Army
B.S., Nanyang Technological University, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 2009

Author: Mok Chuan-Hao

Approved by: Dr. Bert Lundy
Thesis Co-Advisor

J. D. Fulp
Thesis Co-Advisor

Dr. Peter Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The move toward e-government has seen many institutions put special focus on the need for security, especially that of authentication. Single-factor password-based systems have been proven inadequate in safeguarding online financial and e-government service transactions. Industry adoption of Two-Factor Authentication (2FA) has also been piecemeal. To mitigate these deficiencies, the Singapore Government, in 2008, put forth a Call-for-Collaboration (CFC) seeking industry and academic participation in defining a National Authentication Framework (NAF), with the dual aim of providing for a national-level 2FA system and broadening the market for authentication services, and, in so doing, providing the user with a better authentication experience. This thesis will detail, discuss, and compare the various token types and identity frameworks (PKI, SAML, WS-F, OpenID, and Infocard) that make up an authentication system, and make recommendations on the best combination of technologies, protocols, and standards that, when implemented, would not only fulfill the requirements of the CFC, but also position it well for future enhancement.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION....1
A. THESIS BACKGROUND....1
B. AUTHENTICATION: A DEFINITION....1
C. NAF: IMPETUS, BENEFITS, AND CHALLENGES....3
D. THESIS OUTLINE....4

II. AUTHENTICATION USE CASE MODEL....5
A. OVERVIEW....5
B. USE CASE ACTORS....5
1. Authentication Operators....5
2. Service Providers....6
3. Users....6
C. AUTHENTICATION COMPONENT FUNCTIONS....6
1. User Registration....6
2. Token Issuance (and Management)....7
3. User Enrollment....7
4. Credential Verification....7
D. FUNCTIONAL MODEL DEVELOPMENT....8
1. Overview....8
2. Functional Models....8
a. Siloed....*9*
b. Centralized....*9*
c. Federated....*9*
3. Selection Criteria....10
a. Criterion #1....*10*
b. Criterion #2....*10*
c. Criterion #3....*11*
4. Analysis....11
a. Criterion #1....*11*
b. Criterion #2....*11*
c. Criterion #3....*11*
5. Interoperable Model....13
E. USE CASE MODEL....13
1. AO(USER)....14
2. AO(SP)....14
F. SUMMARY....15

III. AUTHENTICATION TOKENS....17
A. OVERVIEW....17
B. TOKEN TYPES....17
1. Memorized Secret Token....18
2. Pre-registered Knowledge Token....18
3. Look-up Secret Token....18

4. Out of Band Token 18
5. One-time Password 18
6. Cryptographic Token 19
7. Biometric Token 19
8. Hybrid 19
C. TOKEN THREAT ASSESSMENT 19
1. "Something You Know" Token 20
2. "Something You Have" Token 21
3. "Something You Are" Token 21
D. TOKEN TYPE SELECTION 21
E. FORM FACTOR TYPES AND SELECTION 23
1. Contact Multiprocessor Smart Card 23
2. USB Token 24
3. Contactless Hardware Device 24
4. Key-fob 25
F. SUMMARY 26
IV. IDENTITY FRAMEWORKS 27
A. OVERVIEW 27
B. IDENTITY FRAMEWORK (NON-REPUDIATION SUPPORTING) 27
1. Overview 27
2. Asymmetric Key Cryptography 28
a. Encryption and Decryption 28
b. Authentication 29
c. Digital Signature 30
3. Supporting PKI Standards 31
a. X.509 31
b. Transport Layer Security Protocol 31
4. Infrastructure 33
a. Certificate Issuance 34
b. Key Recovery 34
c. Certificate Revocation 35
C. IDENTITY FRAMEWORK (NON-REPUDIATION NON-SUPPORTING) 35
1. Overview 35
2. OpenID 36
3. Infocard 37
4. Security Assertion Markup Language 39
5. WS-Federation Language 40
6. Evaluation Criteria 42
a. Fit-for-purpose 42
b. Standards-based 42
c. Interoperable 43
d. Ease-of-use 43
7. Evaluation and Analysis 43
a. OpenID 43

b. Infocard 44
c. SAML 44
d. WS-F 44
D. SUMMARY 45
V. NAF IMPLEMENTATION MODEL 47
A. OVERVIEW 47
B. NAF SAML PROFILE 47
1. Overview 47
2. Web Browser SSO Profile 47
3. Proxy Authentication Operator 49
4. Transient Identifier 50
5. Response 51
C. USE CASE REALIZATION 51
D. SUMMARY 53
VI. CONCLUSION 55
A. OVERVIEW 55
B. LESSONS LEARNED: COMPROMISES 55
1. Co-existence 55
2. Non-repudiation 56
C. FUTURE RESEARCH 57
D. SUMMARY 57
LIST OF REFERENCES 59
INITIAL DISTRIBUTION LIST 63

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure 1. Functional Implementation Models. After [11] 8
Figure 2. Interoperable Functional Implementation Model 13
Figure 3. Authentication Use Case 14
Figure 4. Authentication Process 29
Figure 5. Digital Signature Generation and Verification 30
Figure 6. TLS Authentication Process 32
Figure 7. OpenID Authentication Process Flow Diagram. After [32] 36
Figure 8. CardSpace Authentication Flow Process. After [36] 38
Figure 9. SAML v2.0 Component Affiliations.After [37] 39
Figure 10. SAML Authentication Process Flow Diagram 40
Figure 11. WS-F authentication Process Flow Diagram [After 39] 42
Figure 12. Web Browser SSO Profile. After [45] 48
Figure 13. <AuthnRequest> Fields Extract. After [46] 50
Figure 14. <Response> Fields Extract. After [46] 51
Figure 15. Authentication Use Case 52

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Threats to Authentication Token Types. After [10].....20
Table 2. Authentication Framework Comparison.....45

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AKC	Asymmetric Key Cryptography
AO	Authentication Operator
ATM	Automated Teller Machine
CA	Certificate Authority
CIA	Confidentiality, Integrity, Availability
CFC	Call-for-collaboration
CRL	Certificate Revocation List
EOI	Evidence of Identity
EOR	Evidence of Relationship
GtoB	Government-to-business
GtoC	Government-to-citizens
HTTP	Hypertext Transfer Protocol
I-vard	Information Card
IDA	Infocomm Development Authority of Singapore
IA	Information Assurance
IT	Information Technology
NAF	National Authentication Framework
NEAF	National e-Authentication Framework
NRNS	Non-repudiation Non-supporting
NRS	Non-repudiation Supporting
OASIS	Organization for the Advancement of Structured Information Standards
OTP	One-time Password
PAN	Personal Area Network

PIN	Personal Identification Number
PKI	Public Key Infrastructure
PPI	Persistent Pseudonymous Identifiers
RA	Registration Authority
SAML	Security Assertion Markup Language
SFA	Single-factor Authentication
SMS	Short Messaging System
SOA	Service Oriented Architecture
SP	Service Provider
SSO	Single Sign-on
STS	Security Token Service
TLS	Transport Layer Security Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
WBSP	Web Browser SSO Profile
WS*	Web Services Standards Suite
WS-F	Web Services Federation Standard
XRI-ID	Extensible Resource Identifier
XRI-Res	Extensible Resolution Protocol
2FA	Two-factor Authentication

EXECUTIVE SUMMARY

The National Authentication Framework (NAF) Call-for-collaboration (CFC) was launched in 2008 as part of the Singapore Government's plan to provide for a national-level Two-factor Authentication (2FA) system, broaden the market for authentication services, and enhance the user experience. The CFC calls for proposals from industry and academia for a standards-based system that would permit the user the choice of Authentication Operator (AO) and token type used to access an online service. Two requirements listed will have significant impact on the final proposed implementation model: The first is that non-repudiation is to be a feature of the system, and the second is that all proposed technologies must have received recognition as a standard. These requirements serve as assessment criteria in the analysis and selection of token types and identity frameworks that would best serve the NAF.

The requirement for non-repudiation impacts the choice of token type: Only asymmetric cryptographic keys would, when implemented with Public Key Infrastructure (PKI) in place, offer non-repudiation through the application of digital signatures to online transactions. Other token types possess no such characteristic and are thus relegated to transactions that do not require non-repudiation. That the proposed model incorporates only recognized standards limits the choice of identity framework to the Security Assertion Markup Language (SAML) v2.0 standard. It is fortunate, though, that the SAML v2.0 standard also excels in all other assessment criteria, including being fit-for-purpose, supporting proxy authentication and maintaining user anonymity.

The proposed implementation model, incorporating both PKI and SAML v2.0 as the underlying identity framework, and the choice of the asymmetric cryptographic token for its supportability of non-repudiation is, in the author's opinion, the best possible combination of technologies that would not only fulfill the requirements of the NAF, but also position the NAF well for future enhancements. That last consideration is especially critical when considering the amount of resources expected to be expended on such a major effort.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

This work is dedicated to my wife, Melisse, who has been my greatest supporter.

I would also like to thank Dr Bert Lundy, and John Fulp, for their advice and guidance.

And thanks to God, who makes all things possible.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. THESIS BACKGROUND

The desire to reduce bureaucracy, and to better engage the general population, are two of the reasons that have led numerous governments to embark on programs that would see traditional services being made accessible online. The ten-year Intelligent National Masterplan (iN2015) to be implemented by the Infocomm Development Authority of Singapore (IDA) is one such program. To date, nearly 370 online Government-to-business (GtoB) and Government-to-citizen (GtoC) services have been made available [1], joining the substantial number of e-banking services already available. This proliferation, however, has led to a greater frequency of malicious activities. Because of this, as well as the increase in the sophistication of attacks being performed, there is a definite need for stronger forms of protection for online transactions. To increase this protection, among the many initiatives that are part of iN2015, is the development of a National Authentication Framework (NAF). The aim of the NAF is to [2]:

1. "Enable consistent strong authentication for end-users accessing key online services;" and
2. "Make the process of authenticating online identities more vigorous, thereby boosting online trust and confidence in individuals accessing the next generation of online services."

B. AUTHENTICATION: A DEFINITION

Authentication is defined as the verification of the "identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system" [3]. Authentication is a constituent component of *integrity*, itself defined as the process of "guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity" [3]. Integrity is a component of the Information Assurance (IA) *confidentiality*, *integrity*, and *availability* (CIA) triad which constitutes the core principles of information security [4].

Authentication is accomplished through one of the following: 1) demonstrating knowledge of a shared secret, or "something the user knows"; 2) presentation of an authorized token, or "something the user has"; or 3) presentation of a unique physical characteristic, or "something the user is." These proofs of identity are collectively known as *factors of identity*. The withdrawal of cash from an *automated teller machine* (ATM) illustrates the use of factors of identity. A user presents his ATM card token, "something the user has," either through swiping or insertion, followed by the input of his *personal identification number* (PIN), "something the user knows," onto a keypad in order to proceed with the transaction. In Japan, users are obliged to additionally present either the thumb or index finger to a scanner as a further form of identification, "something the user is." There have been efforts at identifying additional factors for authentication, including "someone the user knows" [5], but the latter is of limited use in non-contact situations.

The use of Single-factor Authentication (SFA) in the form of a password or PIN for authentication is prevalent in the majority of online transactions. The continued reliance on SFA has been advised against in view of the relative ease with which such authentication means can be subverted. A prominent case in Singapore involved the compromise of 21 Post Office Savings Bank Internet banking accounts in June 2002 [6]. This led to the voluntary conversion to a Two-factor Authentication (2FA) system in May 2003 [7] with the addition of a hardware One-time Password (OTP) token, a first in the region at the time. The Federal Financial Institutions Examination Council (FFIEC), having recognized the inadequacy of SFA, had also suggested that financial institutions adopt multi-factor authentication in order to mitigate the risks of account fraud and identity theft [8] in 2002.

Two-factor Authentication methods are aimed particularly at overcoming the shortcomings of password-only SFA methods, and are effective against passive forms of malicious attacks. Against active attacks, however, certain forms of 2FA unfortunately are not as effective [9]. The choice of biometrics, or "something the user is" type tokens, as a factor for remote authentication, is limited. Thus, biometric authentication is better suited for local use, whereby the user presents a livesample of his physical characteristic

for on-site comparison against his previously registered copy without going through a network [10]. The selection of the type of 2FA method is, then, particularly crucial in order to sustain secure online transactions.

C. NAF: IMPETUS, BENEFITS, AND CHALLENGES

Current access to online government services is through password-based SFA means, locally referred to as SINGPASS. This restricts the number of services available online to those that require relatively low assurance. In the financial sector, 2FA has been implemented to a large extent in a stove-piped manner, resulting in consumers having to manage separate tokens for each institution with which they have a relationship; e.g., a user may use a hardware OTP token for transactions with Institution A but rely on Out of Band *short messaging system* (SMS)–based tokens for transactions with Institution B. The number of tokens that may potentially be required to be in a user's possession will grow as more institutions adopt strong authentication mechanisms. The continuance of the current stove-piped system will make token management cumbersome.

With the more interoperable NAF implemented, the Singapore government aims to put in place an infrastructure that not only strengthens the authentication process, particularly through the addition of a second authentication factor for online government services, but also streamlines the user experience for consumers. That is, a consumer may choose to possess a single token acceptable for all online transactions or may choose to be selective as to the number of tokens he or she possesses and also as to with which institutions each token will be affiliated. Any number of reasons may govern the consumer's choice of token type and affiliation, including that of per-transaction cost, hardware and software requirements, or user convenience. Even more important is that the token must be acceptable to the relying institution based on a predetermined level of assurance afforded by characteristics of the token type and its concomitant infrastructure and administration. This requirement for choice will engender the growth of a market in strong authentication services. With a larger consumer base, as compared to previous implementations, which were limited to a single institution's clientele, consumers will benefit through overall lowered costs as a result of economies of scale and competition.

Relying institutions will also no longer need to maintain their own private authentication infrastructure, but can instead outsource such to institutions better equipped for meeting the ever-changing nature of technology, as well as dealing with the associated risks. To enable this market, and to allow for a level-playing field, however, any implementation must be standards-based; and, hence, it is part of the NAF's scope to ensure that proposed solutions are interoperable.

D. THESIS OUTLINE

Based upon the requirements set forth by the CFC document for the NAF, coupled with an in-depth look into the current state of the art in terms of authentication methods and protocols, this thesis aims to suggest a framework model that would be suited not just to fulfilling the requirements, but to ensuring that the derived system is well positioned to meet future challenges. This will be achieved first by illustrating with a use case the interactions between the principal actors throughout the authentication process; this use case will serve as a baseline reference in Chapter II. An analysis of the various types of authentication tokens and associated form factors, as well as supporting authentication frameworks, will be detailed in Chapters III and IV, respectively. In Chapter V, the best-fit components identified in previous chapters will be amalgamated into a system in compliance with the requirements of the NAF, illustrated by a use case depicting how the proposed framework supports authentication. The thesis will conclude with Chapter VI, which details how system requirements affect the overall implementation design and the possible implications.

II. AUTHENTICATION USE CASE MODEL

A. OVERVIEW

From a user perspective, the authentication process as typified by the presentation of a user ID and password is a familiar one. The underlying exchange of data to enable authentication, however, is anything but simple. The complexity derives from the number of parties involved, inter-party interactions, and other associated factors. Prior to deriving the technical implementation, a use case model depicting these interactions will be developed to serve as a basis for the selection of the most appropriate supporting technologies. The use case itself will be developed from a functional model that details the functions of each actor in the use case.

B. USE CASE ACTORS

An actor represents a party to the transaction, e.g., a customer or a bank. The actors of an authentication can be specified based on their roles in the process [10], or as abstractions for the actual system users in a specified scenario [2]. Greater clarity is achieved through role separation; to achieve congruency with the CFC, however, the actors will be specified in terms of actual abstractions from the perspective of the provision of a service. Where necessary, the description of each actor will include the roles each encompasses. Although typical implementations of authentication mechanisms see the involvement of only two actors, i.e., the actor requesting a service and the actor providing the service, a third party may be necessary as an intermediary if one authentication mechanism is used by the one actor to access services provided by more than one other actor. The three principle actors in the authentication process are as follows.

1. Authentication Operators

An Authentication Operator (AO) is a provider of strong authentication services, inclusive of functional and managerial responsibilities, deemed necessary for the proper

administration of credentials. AOs are also otherwise known as "Credential Service Provider," "Credential Issuer," and "Verifier" [10], [11]. The term *credential* is used in this instance to define an "object that authoritatively binds an identity to a token possessed and controlled by a person" [10] or, in shorter parlance, a verified token, such as an authorized smart card issued by the bank. Verisign is but one example of an AO. In particular, Verisign provides digital certification services to Web sites.

2. Service Providers

A Service Provider (SP) is a provider of online services to service consumers. SPs rely on AOs to authenticate users prior to providing services to the users. SPs are alternatively termed "Relying Parties" [10]. In the context of this thesis, SPs are government agencies and other institutions, both public and private, that support online transactions.

3. Users

Users are subscribers to online services, e.g., citizens, residents. Users are most typically the initiators of an online transaction. Since it is the user's identity that requires verification, the user is alternatively also commonly described as a "Claimant" during the authentication process because the user is making a claim regarding his identity.

C. AUTHENTICATION COMPONENT FUNCTIONS

Several functions have been identified that are necessary to actualize the authentication processes. The functions have been adapted from the Australian National E-Authentication Framework (NEAF) [11]. The functions are executed by the actors at specific points of the authentication process, making up the Functional Implementation Model.

1. User Registration

The *user registration* function represents the processes associated with the initial creation of an electronic identity (e-identity) for a user, encompassing Evidence of

Identity (EOI) or Evidence of Relationship (EOR) processes, e.g., the selection of an *online identifier* (user ID) when registering for a new bank account at a bank's branch office by physically providing one's SSN and a valid driving license as proof of one's identity.

2. Token Issuance (and Management)

A *token* is "something that the Claimant possesses and controls used to authenticate the Claimant's identity" [10]. A token is provided to the user for subsequent online authentication transactions. No token is perpetual, and the issuing agency is responsible for ensuring the validity of the token throughout its life cycle and for any subsequent mitigation actions required, should a malfunction occur. Examples of a token include a sealed envelope containing the user's PIN, or an OTP key-fob. Greater detail on the types and characteristics of various tokens will be undertaken in the next chapter.

3. User Enrollment

User enrollment refers to the act of *binding* an authentication token to a known instance of a user identity within an Information Technology (IT) resource context, resulting in establishing a credential. For example, a credential is created when a user initially logs into the bank's Web site with a user ID and PIN, then registers the OTP key-fob by entering the unique key-fob serial number into the bank's Web page; those actions cause the unique key-fob to be associated with the user's account.

4. Credential Verification

Credential verification is the verification of an enrolled token, which takes place as a precursor to enabling the conduct of a transaction. It encompasses the issuance of a positive identity indicator, known as an *assertion*, to a requesting SP. For example, a bank's Web page will seek entry of the OTP from the user's key-fob (i.e., the token) for verification of the user's identity prior to servicing the user's request. The term *credential* is used in this context, as opposed to *token*: the token would have been enrolled and

bound to an identifier, prior to the need for verification. Validation, which relates to the checking of the status of the credential at the time of verification, is implied.

D. FUNCTIONAL MODEL DEVELOPMENT

1. Overview

This section defines a functional model specification incorporating the defined component functions. Three existing models—Siloed, Centralized, and Federated—are first described. Selection criteria are presented against which the models are analyzed. A fourth model, Interoperable, is thereafter developed based on the existing models, with augmentation based on the established selection criteria.

2. Functional Models

Three functional models have been identified in support of authentication—Siloed, Centralized, and Federated [11]. Each model differs primarily in the actor responsible for the provision of component functions. The details of the three models are captured in Figure 1, and explained below.

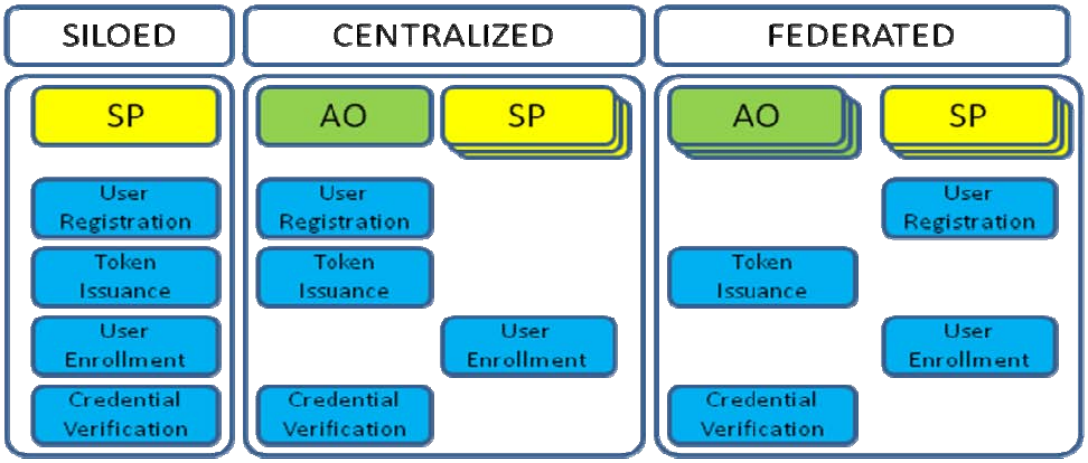

Figure 1. Functional Implementation Models. After [11]

a. Siloed

This model is representative of current authentication mechanisms being deployed by major financial institutions in Singapore. All component functions are provided by the SP. Individual institutions contract separately for the procurement and establishment of in-house proprietary authentication mechanisms [12], [13]. This results in individuals possessing multiple authentication tokens, one for each institution with which the individual has a relationship. As this model eliminates the need for AOs, it results in a simpler transactional process, as well as the fastest transactions. The need for in-house infrastructure, though, would require high initial capital outlay, as well as continuous life-cycle costs, which may prove to be a barrier to entry for all but the largest institutions. The model does not benefit from economies of scale.

b. Centralized

This model sees a user registering with an AO for the provision of a global identifier and a token. The user's credentials (global identifier and token) are subsequently enrolled with each SP. When a user requires a service, the user presents his credentials to the SP, which will subsequently be redirected to the AO for verification. As authentication services are outsourced, SPs no longer need bear the costs involved in maintaining in-house systems. The cost to both SP and the individual is reduced in light of the shared infrastructure. Individuals may also have a choice in the form factor and method of authentication. Concerns include the monopolistic dominance of a single provider, a single point of failure, increased transaction time, and privacy issues arising from having a global identifier registered across all SPs.

c. Federated

The Federated Model differs from the centralized model in that there is no requirement for a global identifier. The AO is responsible only for the issuance of the token. The user enrolls the provided token with the SP; the token and its attributes will be associated with the user's unique identifier with the SP, resulting in a credential. Each time the user requires a service, the user presents his or her credential to the SP, which

will request separate verification of the attached token from the AO. Positive verification is communicated back to the SP from the AO in the form of an assertion. This mechanism serves as the basis for Single Sign-on (SSO), which translates to greater user convenience because only a single authentication process is required to access multiple SPs, assuming that all relying SPs have a relationship with the same AO. The Federated Model provides an additional layer of privacy to the consumer. There is no global identifier, as with the Centralized Model, resulting in there being less risk of associating contents from different SPs to the same user. Initial transaction duration is expected to be longer as a result of the greater complexity of the SSO operation; but overall performance would be greatly enhanced.

3. Selection Criteria

With most systems already implementing the Siloed Model, the emphasis of the NAF would be the implementation of a system whereby more commonality could be achieved. The decision as to whether a centralized or federated model would be better suited to the NAF would be dependent upon the requirements as laid out in the CFC. Three important requirements have been identified.

a. Criterion #1

Cross-authentication of individuals/SPs is to be enabled across AOs, e.g., an individual A utilizing AO X will be able to utilize services provided by SP B, utilizing AO Y without a noticeable degradation in performance.

b. Criterion #2

Implemented systems should incorporate/build upon current systems, i.e., they should support SP-specific user ID and password/PIN as the first authentication factor in a 2FA implementation.

c. Criterion #3

Users will have a choice of AOs, token type, and form factor employed for authentication. Accordingly, each SP may determine the minimal level of assurance necessary to access its services, translating to which of the various methods would be acceptable to the SP as an authentication means.

4. Analysis

The characteristics of the centralized and federated models were compared based upon the criteria developed in the previous subsection. The results of the comparison are discussed below:

a. Criterion #1

In the Centralized Model, with only a single AO, there is no requirement for cross-authentication. For the Federated Model, the use of Persistent Pseudonymous Identifiers (PPI) may be necessary between AO X and AO Y, and between AO X and SP B, in order to ensure that the generated assertion can be mapped to the user at the SP end.

b. Criterion #2

In terms of criteron 2, for the Centralized Model, the use of a global identifier may require a PPI to map currently-existing IDs with SPs to the new global identifier. The nature of the Federated Model supports this requirement organically, since users can maintain their original unique IDs as registered with the SP.

c. Criterion #3

The strength of the Centralized Model lies in there being a single AO. The requirement that users and SPs have a choice in their AOs violates this characteristic of the centralized model. The Federated Model organically supports the use of different AOs and is, hence, well suited to fulfill this requirement.

From criterion 1 the lack of any need for cross-authentication puts the Centralized Model at an advantage over the Federated Model. From criterion 2, the inherent support of existing authentication processes gives the Federated Model the advantage. From criterion 3, the Centralized Model's inability to support multiple AOs, however, results in the Federated Model being the logical choice as the basis for the development of the use case. The Australian, Canadian, and New Zealand governments' choice of the Federated Model also lends weight to its selection [11], [14]. However, the SSO characteristic of the Federated Model may be considered excessive for the proposed NAF. There is no explicit requirement for an SSO functionality across domains, i.e., between that of the government and the private sector. Neither are there explicit restrictions on implementations that allow this functionality. Rather, it is the envisaged usage of the system by the government, as a means of imposing a second authentication factor to that of the single-factor system already in place, that predisposes any upcoming system toward that configuration, i.e., any user who is logged onto a non-governmental SP, and who would subsequently wish to use a government service, would not be able to use any inherent SSO functionality because the government service would require the entry of the first authentication factor that is separate from that already provided to the non-governmental SP. This configuration, using the NAF as the second authentication factor, is also in sync with the PIN-based single authentication factor system employed by local financial institutions. This system utilizes the same PIN both for ATM transactions and as a password to the user ID. This mechanism demands that the PIN remain under the control of the issuing organization, the SP, for privacy and security reasons. Any allowance for SSO among private and public entities is expected to require a level of agreement between all parties. That agreement is expected to require much more work than existing agreements among collections of educational institutions—currently, the most prolific users of SSO. That said, within the government, access to services provided by different agencies is expected to benefit from SSO; it is only at the public–private boundary that no compromise seems possible in the short run. Nevertheless, it would be shortsighted to put in place a system that does not have the potential for SSO, should the necessary agreements come into effect in the future.

5. Interoperable Model

It would be more accurate to define a modified Federated Model with a focus on interoperability at the AO level as the basis for the use case. This model, to be termed the Interoperable Model, inherits all of the characteristics of the Federated Model, less the SSO characteristic. The Interoperable Model, denoted by placing a superscript I to AO to indicate interoperability among the AOs, is captured in Figure 2.

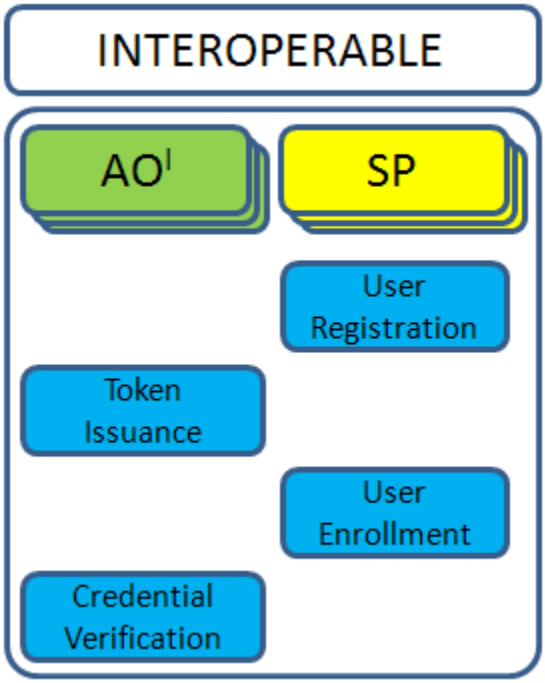

Figure 2. Interoperable Functional Implementation Model

E. USE CASE MODEL

Based upon the Interoperable Model, a baseline use case model is developed to illustrate the authentication process. The use case, in which both user and SP share a common AO, is reminiscent of the Centralized Model and is considered trivial. The use case developed will be based on the condition whereby user and SP subscribe to different AOs for the provision of authentication services. The differentiation of AOs is as follows.

1. AO(USER)

Defined as the AO to which the user is subscribed, the AO(USER) is responsible for credential issuance and management for the user; synonymous with the *AO(Issuing)* term used in the CFC.

2. AO(SP)

Defined as the AO that the SP has selected for the provision of credential verification services; AO(SP) is synonymous with the *AO(Front)* term used in the CFC.

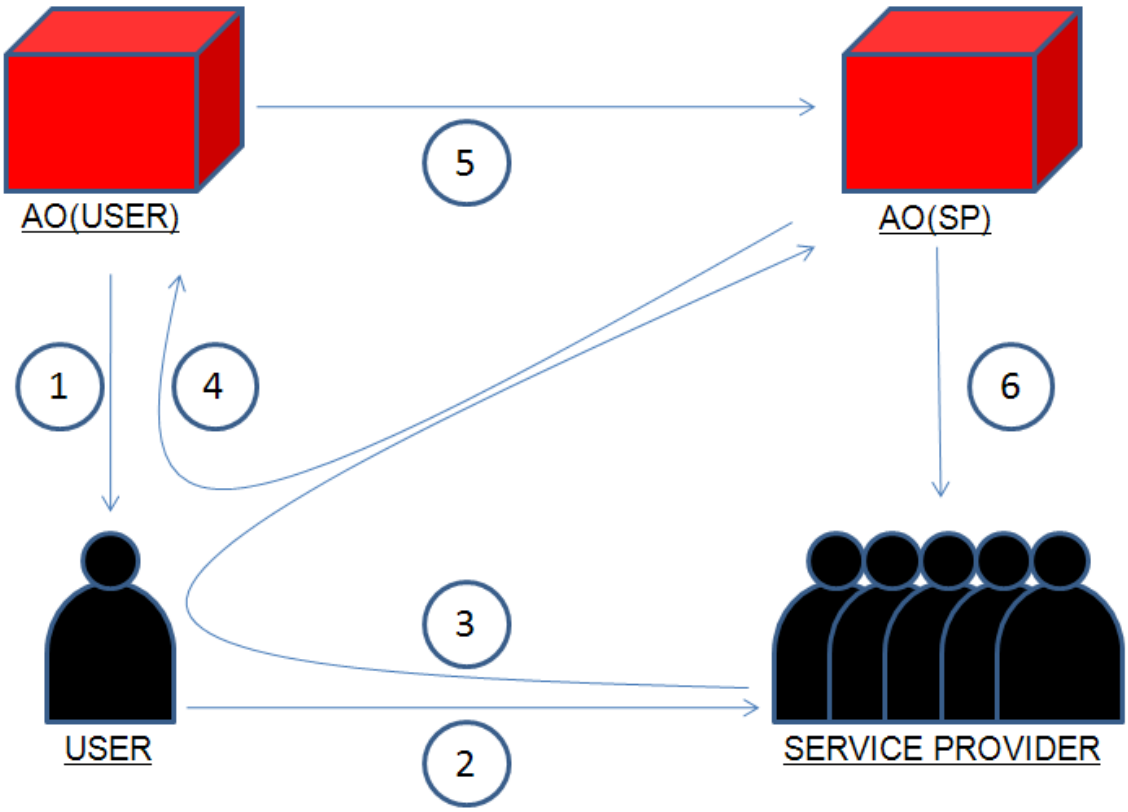

Figure 3. Authentication Use Case

The authentication use case takes places in the following steps, as depicted in Figure 3:

1. Credential is issued to subscribed user.
2. Pre-registered user (enrolled) presents first factor to SP for authentication.
3. SP directs user to AO(SP) for second factor authentication.

4. AO(SP) directs user to AO(User) for presentation of second factor.
5. AO(User) verifies presented second factor; issues assertion to AO(SP).
6. AO(SP) transmits assertion to SP, verifying identity of user.

F. SUMMARY

In this chapter, a baseline use case model, detailing the interactions between the various actors as part of the authentication process, has been developed that will serve as a reference for the technical implementation model. The technical implementation model will be developed in later chapters of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. AUTHENTICATION TOKENS

A. OVERVIEW

In Chapter II, we touched on how a token would be used as part of the authentication process. As previously mentioned, a token is either "something you have," "something you know," or "something you are" that can be used to positively identify the user. This being the case, the choice of the kind of token is cardinal to the implementation of any authentication system. Each token possesses a distinct set of characteristics that will render its use advantageous under certain conditions and yet still susceptible to certain threats. In this chapter, the different types of tokens and their characteristics will be presented, with particular interest paid to the kind of threats to which each token type is susceptible. The chapter will continue with a survey of the current form factors that implement/store tokens. This chapter will conclude with a recommendation on the form factor and token that would best support the requirements of the NAF.

B. TOKEN TYPES

A survey of literature providing guidelines for the implementation of an authentication system yielded varied classifications for the types of tokens. Factor-based lists [14] classify tokens in accordance with the underlying factor type. While simple and elegant, there is no clear distinction among the variety of tokens within a single factor. Functional-type lists [10], [15] often combine a token type and a form factor, obfuscating the relationship of the two while resulting in a lengthy catalogue. The list of token types presented below, borrowing heavily from the NIST SP 800-63 [10], provides a succinct and comprehensive listing of the current token types, exclusive of the form factor which will be used to implement the token. Arranged loosely in order of factor type are the "something you know" Memorized Secret Token and Pre-Registered Knowledge Token; "something you have" Look-up Secret Token, Out of Band Token, One-time Password Token, and Cryptographic Token; and "something you are" Biometric Token. The final

token, while not strictly a token type, combines different factors into a Hybrid Token. The definitions of each token type are as follows.

1. Memorized Secret Token

This token type is a shared secret between the user and the verifier that the user memorizes, e.g., a password.

2. Pre-registered Knowledge Token

The Pre-registered Knowledge Token is a set of prompts/responses that has been established between the user and the verifier. The user responds with the appropriate answer to a prompt from the verifier, e.g., a response of “Lee” to the question, “What is your mother’s maiden name?”

3. Look-up Secret Token

A database of one or more shared secrets between the user and the verifier is defined as a Look-up Secret Token. The user responds with the appropriate secret to a prompt from the verifier, e.g., a code book.

4. Out of Band Token

An Out of Band Token is a secret sent from the verifier to the user through a pre-established secondary communications medium; the user subsequently submits that secret into the primary channel for authentication, e.g., the transmission of a PIN to a user’s cell phone through SMS, which the user subsequently enters into the appropriate field on the verifier’s Web site.

5. One-time Password

This token type is a time-limited password obtained from a device the user possesses, hence the term “one-time,” e.g., a PIN generated by a key fob, which is subsequently entered into the appropriate field on the verifier’s Web site. Authentication is achieved when the entered PIN is the same as that generated at the verifier’s end.

6. Cryptographic Token

A persistent symmetric or asymmetric cryptographic key stored in or generated by either hardware or software means is a Cryptographic Token. For example, the cryptographic key is used to encrypt a challenge issued by the verifier and thereafter submitted back as a response. The verifier decrypts the response to obtain the originally issued challenge and, if it matches what was previously issued, effectively authenticates the user since only the user would have the correct key to encrypt the challenge.

7. Biometric Token

This token type is a distinguishing physiological or behavioral characteristic presented for verification against a database, e.g., a thumb-print image captured on a reader.

8. Hybrid

A combination of two or more token types used for authentication would be considered by the author as a Hybrid token. While not strictly a token type, hybrid tokens consist of multiple tokens, often with differing factor types, yielding what is known as a multi-factor token, e.g., the use of a biometric token to unlock a smart card containing the user's unique private cryptographic key.

C. TOKEN THREAT ASSESSMENT

The strength of a token and its use as part of an authentication system is very much dependent upon its resilience in the face of current threats. Each token type is characterized by strengths and weaknesses, which must be assessed for suitability of each for use in the NAF. Table 1, based upon a list similar to the one found in the NIST SP 800-63 standard, lists the most current threats to authentication tokens. Appended to each threat are the token types that are most susceptible to that threat. Hybrid tokens are not included in the list. The following discussion looks at each token factor type in turn.

Table 1. Threats to Authentication Token Types. After [10]

Threat	Description	Example	Susceptible Token
Theft	A token with a physical manifestation is stolen by an Attacker.	Hardware cryptographic device stolen; cell phone stolen; OTP device stolen.	3,4,5,6
Duplication	The user's token has been copied with or without his or her knowledge.	Password written on paper disclosed; passwords stored in electronic file copied.	1,2,3,4
Eaves-dropping	The token secret or authenticator is revealed to the Attacker as the Subscriber is submitting the token to send over the network.	Shoulder surfing of passwords; keystroke logging on keyboard; PIN captured from PIN pad device; fingerprint data captured from reader.	1,2,3,4,5,7
Emissions analysis	The token is exposed using analytical methods outside the authentication mechanism.	Differential power analysis on stolen hardware cryptographic token.	4,5,6,7
Phishing or pharming	The token secret or authenticator is captured by fooling the Subscriber into thinking the Attacker is a Verifier or Relying Party.	Password revealed by Subscriber to website impersonating as the Verifier.	1,2,3,5
Social engineering	The Attacker establishes a level of trust with a Subscriber in order to convince the Subscriber to reveal his or her token or token secret.	Token revealed by Subscriver in telephone inquiry from masquerading system administrator.	1,2,3,4,5
Guessing / Brute-Force Attack	The Attacker attempts to obtain the token by systematically trying a large number of permutations of the key set.	Online/offline dictionary attacks to guess passwords.	1,2,3,4

1. "Something You Know" Token

"Something you know" tokens are susceptible to most forms of attacks. Their inherent susceptibility is exacerbated by other conditions: 1) User behavioral weaknesses expose the token to social engineering attacks. 2) Limits on users' recollection abilities not only result in creation of passwords of limited character complexity and number but also subject those passwords to repeated use in different domains. 3) Vulnerability to cross-references allows the actual password or knowledge token to be implied or deduced from open source material; e.g., a password based a user's birth date can be obtained easily from the information on a user's FaceBook home page. The weakness of this token

type was one of the main factors for the push for a 2FA system, in which the second factor belongs to one of the two remaining token factor types.

2. "Something You Have" Token

Being physically manifested, "something you have" tokens are susceptible to simple theft. Beyond that, the vulnerabilities of the token types within this category differ greatly. A Look-up Secret Token is subject to most of the vulnerabilities inherent in "something you know"–type tokens, though it does afford greater protection should there be limited reuse of the token; limited reuse would overcome the replay attack vulnerability. An Out of Band Token provides greater protection since it requires the attacker to have access to the secondary communications medium used to transmit the token. A phishing Web site may not have the necessary data, e.g., cell phone number, to send the token to in order to complete the sham even if a user unwittingly encounters a phishing site. An OTP token takes this protection further by limiting the validity of the token in the time domain, usually not more than 30 seconds. A Cryptographic Token, while perpetual, is the least vulnerable token due in part to the generally concealed nature of the token: the user has rare access to the exact characters of the token; and the length of the token, typically many times longer than that of a password or OTP pass-phrase, makes brute force attacks infeasible.

3. "Something You Are" Token

The use of biometrics has already been restricted to local mechanisms in Chapter I. That said, biometrics may still be subjected to attacks even in this limited setting, such as by the lifting of an imprint left on the reader following a subscriber's use.

D. TOKEN TYPE SELECTION

The previous section expounded on the vulnerabilities of the various token types. While vulnerability remains a key determinant in the choice of token type, a key characteristic that tokens selected for the NAF must possess is the ability to support non-repudiation.

Non-repudiation, while not a feature of a token, has been explicitly listed as a requirement for the NAF in support of certain government online services. To realize non-repudiation, any message exchanged between the user and the SP is required to have in place a means of proving undeniably the authenticity of its source and its destination, i.e., a sender cannot deny sending the message, nor can the receiver deny having received the message. This requirement influences the choice of token type because only certain tokens and their associated mechanisms satisfy the conditions necessary for non-repudiation. Chief among these conditions would be that the element associating an actor to a message be irrevocably unique at the time of the transaction. This condition effectively eliminates all schemes that are based on token types whereby a user demonstrates knowledge or possession of a secret shared with the verifier. If this secret is leveraged as a means toward non-repudiation—e.g., by attaching an encrypted digest of the message to an outgoing e-mail as a form of a digital signature using a shared symmetric key—there is a chance that either the user or verifier may falsely claim that a message originated from the other party instead of themselves, since both the user and the verifier have equal access to the same symmetric encryption key and, hence, are both able to form the digital signature. Hybrid schemes that generate a "signature" based upon a combination of transaction attributes is still subject to replication by an entity, usually the verifier, with knowledge of the component attributes of the signature [16]. This leaves asymmetric cryptographic schemes as the only reasonable choice for solutions that require non-repudiation. The private key, which only the user possesses and uses to digitally sign any outgoing messages, may be irrevocably traced back to the user should the need arise, assuming a robust Public Key Infrastructure (PKI) is in place. Security is further enhanced if cryptographic operations that generate the digital signature are performed by a hardware device unique to the user, as compared to software variants, which are subject to risks inherent particularly to software design. The generation of digital signatures with the use of an asymmetric key is discussed in detail in the next chapter.

From the previous discussion on authentication threats, it is clear that Cryptographic Tokens would provide the best protection against most current threats.

When the need for non-repudiation is factored in, the only choice would be an Asymmetric Key Cryptographic Token. That said, users may still choose to leverage on token types that do not support non-repudiation for authentication to SPs that do not require it. This is especially pertinent in view of the relatively larger infrastructure footprint required of the user to support non-repudiation, with a direct impact on mobility. For this subset of transactions, the remaining token types that are of the "something the user has" form may be utilized as the second authentication factor, particularly OTP and Out of Band Tokens.

E. FORM FACTOR TYPES AND SELECTION

A survey of various authentication operators yielded a large variety of authentication form factors. A majority of solutions that claim to apply strong authentication principles implement a variation of an OTP mechanism [17], e.g., RSA Secure-ID, VASCO Digipass, Aladdin eToken. The major advantage afforded by this form of authentication is its connectionless nature. Unfortunately, this method does not ensure non-repudiation because both the user and verifier would possess the same OTP. Continuing from the previous discussion, only hardware-based, asymmetric key cryptography-supporting solutions would be sufficiently strong to withstand current threats and yet meet the requirements for non-repudiation. Market trends also show a steady increase in the acceptance of PKI-based smart card solutions. This chapter discusses various form factors that implement OTP, Out of Band, and Asymmetric Key Cryptographic Tokens.

1. Contact Multiprocessor Smart Card

A smart card is "a credit-card sized card with an embedded processor and memory that can receive input, process it, and provide output" [18]. It operates by performing cryptographic operations on inbound data with the user's private key and then outputs the encrypted message back to its source. The smart card has been in use by major institutions, including the U.S. Department of Defense. The cost of a basic smart-card solution is US$23 per individual (smart card US$3[19], smart card reader

US$20[20]). The smart card form factor is also familiar to users since all would already be in possession of a government-issued ID card. It is not too remote to expect such an ID card to be upgraded to smart card specifications as a means toward nationwide implementation. Yet, while a smart card solution is fairly cost competitive, the requirement for an additional reader may render the solution less attractive in light of the current trends towards portability.

2. USB Token

A USB token is essentially a key fob containing a built-in USB port with the same processing functionality as a smart card. This form factor was designed with the aim of eliminating the need for a separate smart card reader. Users need only attach the key fob to any available USB port for authentication to be effectively carried out. The security of this form factor is strengthened through adherence to the FIPS 140-2 standard, safeguarding the cryptographic keys held within through tamper proofing and associated methods. This form factor is also congruent with the OTP key fob, which is currently issued by most major banking institutions as their means of strong authentication, in terms of look and feel and, hence, would be most familiar to users of online banking. The price per unit is expected to be less than US$30 per unit [21]. The USB token can be further enhanced with the additional requirement of the need for a second token, usually biometric or PIN-based, to unlock the cryptographic keys stored within. This addition of a token will require an enhancement to the standard smart card/key fob form factor, usually through the addition of a key pad or a biometric scanner. This enhancement is also available to the smart card form factor.

3. Contactless Hardware Device

The USB token, while not requiring a separate reader, still requires direct connection via a USB port. The advent of Personal Area Network (PAN) technologies, such as Bluetooth and RFID, makes wireless connectivity possible. Consumer devices such as smartphones can be leveraged as a means of storing the necessary keys and performing the desired cryptographic operations. The smartphone can also be utilized to

implement Out of Band token-based solutions, particularly by the entering of an OTP token, received via SMS from the AO, into the AO's authentication page by the user. Another use of the smartphone implements biometrics in the form of voice authentication as a third authentication factor whereby a user receives a call on his phone (second out-of-band token) and he is asked to read out his PIN. The system authenticates the user based on the unique characteristics of his voice [22]. Most smartphones enable connectivity via Bluetooth, though the smartphone may require the installation of an additional trusted cryptographic module.

Another contactless hardware device is a contactless smart card. As the name suggests, it has essentially the same functionality as a smart card but with an add-on wireless means of connectivity—radio frequency. The performance characteristics of the contactless smart are specified by the ISO/IEC 144443 standard; among other features, it defines the communications distance at 10 cm. Large-scale implementations include the Octopus Card in Hong Kong and the Ezylink card in Singapore. While the cost of a contactless smart card is competitive, the lack of a reader in most consumer products means this form factor may require significant investment to implement. The cost of a contactless reader is expected to be around US$90.00.

The key issue regarding the use of contactless solutions is security. Wireless communications means are associated with numerous security concerns. The complexity of smartphones, their dual-use nature for non-secured communications, and the relative lack of security inherent in smartphone operating systems [23] increase the chances of subversion and subsequent compromise of the embedded cryptographic keys.

4. Key-fob

A key-fob is a device similar in look and feel to a key-chain ornament but containing electronics capable of issuing a unique OTP upon activation, visible to the user through a liquid crystal display. Such systems place no additional demands on users, such as needing to use and/or install hardware and software, except for the need to have the key-fob in their possession. It is easily revoked from the AO end and can be easily replaced. The key fob also benefits from being the dominant form factor in current use.

Each key fob costs about US$5.00 [24].Variants of a key fob include the addition of a keypad so that the user may enter a PIN to unlock the generated OTP. Such a mechanism is similar to the enhanced smart card previously mentioned.

With each form factor having its own strengths and weaknesses, and in line with the government's aim of providing for choice, it may be best if the decision of which form factor to adopt be left to the consumer, based on his or her own lifestyle. Each form factor possesses distinct characteristics in terms of footprint, infrastructure investment, and security concerns. A caveat is that the selected form factor must be able to implement the correct token type for the requested service.

F. SUMMARY

In this chapter, a summary of the various token types in regards to current threats has been presented. An asymmetric cryptography-capable token has been selected as the best means for authentication while ensuring non-repudiation, whereas an OTP or Out of Band token-based system would suffice for systems that do not require non-repudiation. In the next chapter, we will look at the identity frameworks that will be leveraged on to authenticate the user to the SP.

IV. IDENTITY FRAMEWORKS

A. OVERVIEW

In Chapter II, a use case was developed as a baseline reference for the implementation of the National Authentication Framework (NAF). Key components of the process are those of authentication and the transmission of the assertion from the Authentication Operator (AO) to the Service Provider (SP), indicating concurrence as to the identity of the user. While authentication in itself is a fairly well-developed mechanism, it is the transmission of the assertion that poses a challenge in the implementation of the authentication framework. The possible involvement of more than a single AO is an issue that has yet to be addressed specifically [25]. The realization of both an authentication and an assertion transmission is dependent upon the choice of the supporting identity framework. The underlying protocols and standards that make up each framework vary in their support of the selected token type and associated authentication mechanisms. The key differentiating factor lies in whether a framework is Non-repudiation Supporting (NRS) or Non-repudiation Non-supporting (NRNS). It has been established in Chapter III that both forms will co-exist as part of the overall NAF. This chapter will explore current frameworks that support either of the two forms, evaluate their respective suitabilities, and draw a conclusion about the best-fit framework for support of the NAF.

B. IDENTITY FRAMEWORK (NON-REPUDIATION SUPPORTING)

1. Overview

The use of asymmetric key cryptography for authentication is supported by a collection of entities known as Public Key Infrastructure (PKI). These entities provide the means, not just for authentication, but also for ensuring non-repudiation, one of the key

requirements of the NAF. This section will begin with an overview as to how asymmetric key cryptography operates, will provide a list of critical supporting standards, and will conclude with a description of how these standards are leveraged to enable NRS.

2. Asymmetric Key Cryptography

In Asymmetric Key Cryptography (AKC), as the name suggests, different keys are used to encrypt and decrypt the data of interest. Supporting algorithms that offer AKC include the Rivest-Shamir-Adleman algorithm and the elliptic curve algorithm. While each algorithm differs in the means of key generation, the mechanism in which the key-pairs are utilized remains the same and will be discussed in this section. The use of AKC for authentication and generating digital signatures is also discussed.

a. Encryption and Decryption

The mechanism for employing encryption and decryption with AKC is as follows: Each user has a key-pair, consisting of a public key and a private key. The public key is freely available, while the private key is kept only by the user. Assume that user A, Alice, wishes to send a message M to user B, Bob. Alice wishes to ensure that no person other than Bob will be able to read the message. Upon generation of the message M, Alice uses Bob's public key, KB, to encrypt the message, resulting in M_{KB}. Bob, upon receipt of M_{KB}, uses his personal private key, KB^{-1}, to decrypt M_{KB} and successfully obtains the message. Because only Bob possesses the requisite private key, only Bob can decrypt the message; and no one else, not even Alice, who used Bob's public key to encrypt the message, is able to retrieve the message from its encrypted form. Bob's public key is publicly available in the form of a digital certificate, the structure of which will be covered shortly. In effect, each person would have a private key, and a public key, together termed a public-private key-pair.

b. Authentication

Using AKC for authentication usually features a challenge-response protocol. The process is illustrated in Figure 4.

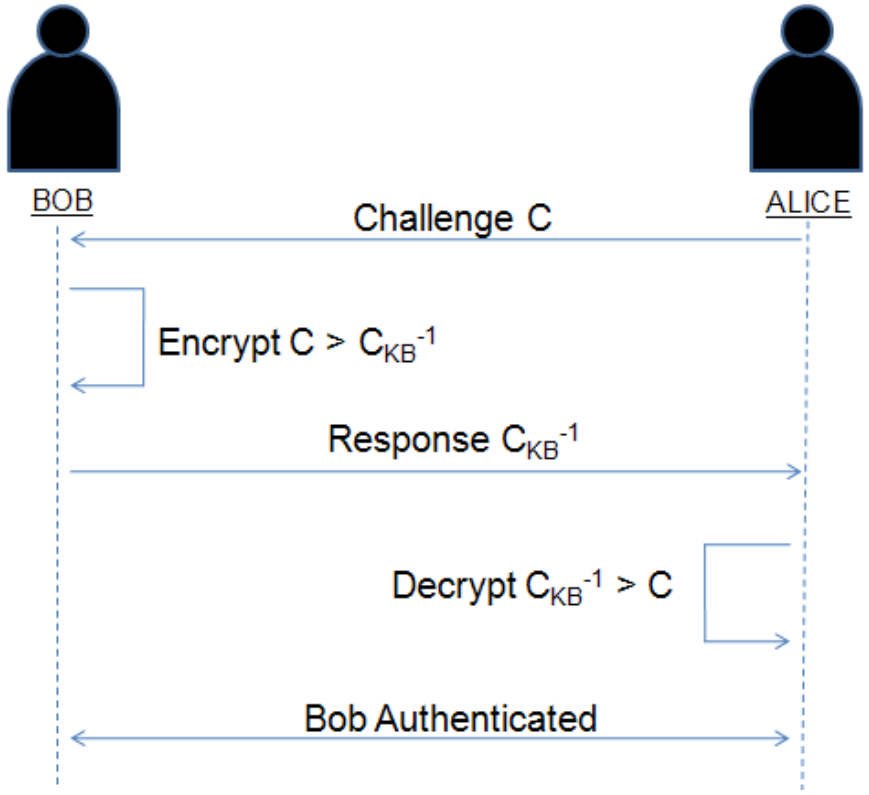

Figure 4. Authentication Process

Bob wishes to assess a resource controlled by Alice. Alice issues a challenge C to Bob. Bob uses his private key KB^{-1} to encrypt the challenge C_{KB}^{-1} and sends the encrypted challenge, now termed a response, back to Alice. Alice utilizes Bob's public key to decrypt the response. If Bob's identity (as provided by Bob's identity being "bound" to the public key used by Alice) is authentic, the response will decrypt accurately to reveal the challenge. Bob's identity is verified because only he should have possession of the private key that corresponds to the freely available public key used to decrypt the response.

c. Digital Signature

A digital signature is the primary means by which non-repudiation is effected. Essentially, any message that is accompanied by a digital signature can be trusted in terms of its source, since the digital signature is derived from a sender's private key and the message itself. The digital signature generation and verification process is depicted in Figure 5.

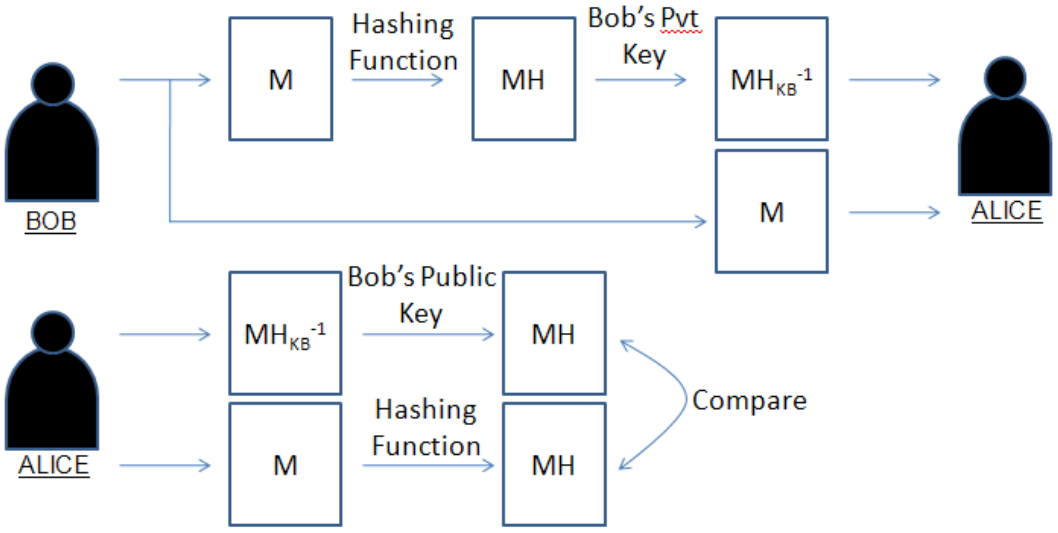

Figure 5. Digital Signature Generation and Verification

The structure of the digital signature is simple. Bob intends to send a message M to Alice. After composing the message, Bob hashes the message, resulting in a hashed message MH, and subsequently encrypts the hash with his private key, resulting in MH_{KB}^{-1}. This encrypted hash is sent together with the original message to Alice. A hash is the result of the performance of a hashing function, such as SHA-1 or MD5, on the message, resulting in a message digest that cannot be "reversed" back to the original message by any existing means, even by the message originator. Alice, upon receipt of the message, first hashes the accompanying message, then proceeds to use Bob's public key to decrypt the encrypted hashed message that Bob sent along with the original message. If the decrypted hashed message and the new hash match, then the sender's identity is established.

3. Supporting PKI Standards

A suite of established and recognized standards supports the implementation of PKI. Chief among the standards is the standard that addresses the format of the digital certificate and the protocol used to support authentication via smart card-type tokens. These and other standards are discussed in the rest of this section.

a. X.509

The X.509 standard (RFC 5280) defines one possible format of a digital certificate. The digital certificate is the basis for the authenticity of a freely available public key. The digital certificate is composed primarily of the name of the user and the AO, the user's public key, the validity period of the public key, the identifier for the type of cryptographic algorithm used, and the digital signature of the AO [26]. The AO's digital signature signifies the validity of the information provided in the certificate, notably the binding of the user's identity to the user's public key. X.509 also specifies the format of the Certificate Revocation List (CRL). The CRL is a time-stamped list of all certificates that have been revoked and can no longer be relied upon. The CRLs are signed by the issuer and, like public keys, are available publicly. Prior to reliance on any public key found within a submitted digital certificate, a check should be made against the most recently published CRL to ensure that the certificate a) has a valid AO signature, b) has not expired, c) does not appear on a CRL, and d) is being used as intended.

b. Transport Layer Security Protocol

The most widely used protocol for securing e-commerce transactions is the Transport Layer Security Protocol (TLS). Designed to provide "privacy and data integrity between two communicating applications" [27], TLS is widely accepted and well supported. Transaction security is achieved first through conducting mutual authentication and thereafter establishing a secure channel for the exchange of data. Current use of TLS is restricted to server authentication, since most users do not possess

the necessary tokens for user authentication. With the NAF in place, users can be authenticated to the SP using TLS via their public-private key-pairs. The authentication mechanism is illustrated in Figure 6.

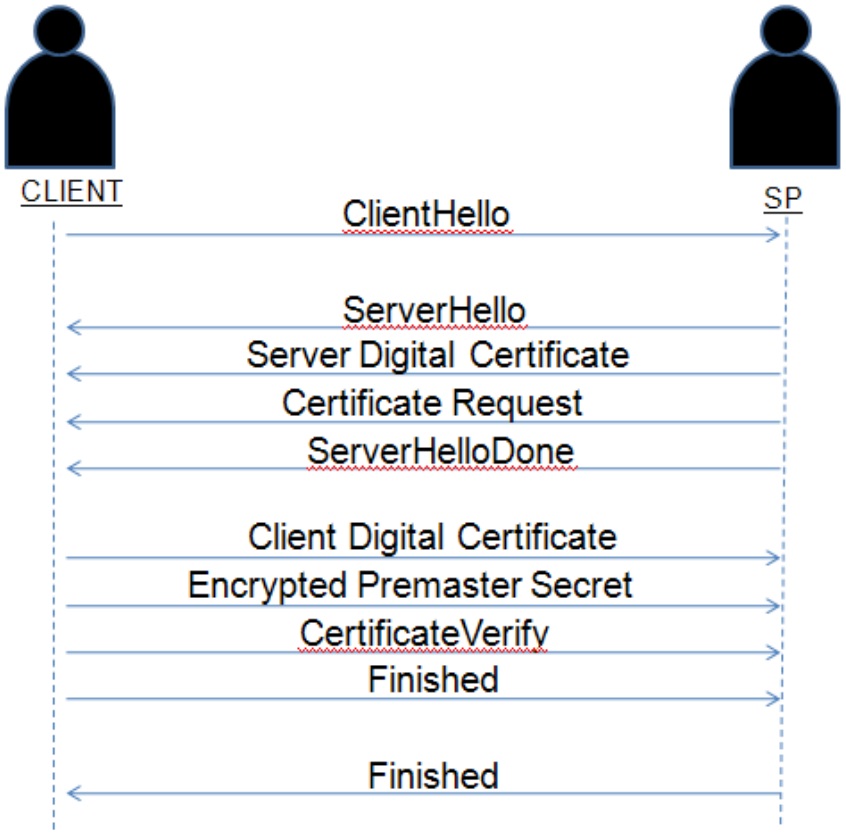

Figure 6. TLS Authentication Process

The user sends to the SP a message, *ClientHello*, indicating the user's intention to establish a connection, shown in step one. The *ClientHello* message contains the version of TLS in use, a client_random number, R_C, a list of applicable cipher suites, and a list of compression methods. The SP server, upon receipt of the message, responds in kind with a chain of messages. The first message, *ServerHello*, contains a field indicating the TLS version in use by the server, a server_random number, R_S, the cipher suite selected by the server, and the compression method selected by the server, represented by the respective standardized cipher suite and compression method IDs. This is followed by the X.509-formatted server digital certificate. If the SP wishes to authenticate the user, a *CertificateRequest* message is appended as well. This message

chain from the server, as shown in step two, is concluded with the message *ServerHelloDone*. The user, upon detection of *ServerHelloDone*, proceeds to transmit its own message chain to the SP, commencing with the user's X.509-formatted digital certificate. The second message in the chain is a server public key–encrypted user-generated 48-byte pseudorandom number premaster secret, S. This is followed by a *CertificateVerify* message, which is a user digitally signed hash of all previous messages used in the handshaking process. The final message in the chain, the *Finished* message, is substantially different from the other messages in the chain: its contents are the encrypted hash of all previous messages utilizing a symmetric key, K, derived from the 48-byte pseudorandom number, S, the client_random number, R_C, and the server_random number, R_S; hence, $K=f(S,R_C,R_S)$. The user-message chain is shown in step three. The server likewise responds to the user with its own *Finished* message, a hashed message digest of all previous transacted messages also encrypted with K. Note that key K, known only to the user and the SP, will be used to encrypt and decrypt all further communications between the two, ensuring confidentiality. Mutual authentication is assured in a few ways. Firstly, the 48-byte pseudorandom number S used to generate the key K was encrypted with the AO-certified SP server's public key. The SP server will only be able to generate the correct K if it possesses the corresponding private key which allows the server to decrypt and obtain the 48-byte pseudorandom seed number S. The user is likewise authenticated by the server through verification of the digital signature submitted as part of the *CerificateVerify* message. The accurate exchange of the *Finished* messages is proof that both the user and the server have derived the correct K from the list of transactions. If at any time any of the transmissions in the process should fail, the connection is immediately terminated.

4. Infrastructure

The implementation of PKI is not just dependent on the identification of the correct standards, but also on the implementation of these standards in a manner that ensures security. Examples of sub-par implementations of otherwise trustworthy

standards include WEPs implementation of RC4. In PKI, two key functions have been identified, *certificate issuance* and *certificate revocation.*

a. Certificate Issuance

The issuance of certificates to both users and SPs (collectively termed *client* henceforth in this section, both being clients to the AO) alike is key to the implementation of PKI. It is the trust that both parties place in the AO having correctly executed the necessary identity background checks prior to issuance of a certificate that is the basis of PKI. In this regard, the AO may play the role of the Certification Authority (CA), or the Registration Authority (RA), or both. The latter is responsible for the generation of the asymmetric key-pair, maintenance of the validity of the key-pair, and issuing the key-pair to the dependent client, while the former creates the corresponding digital certificate. For a client to receive certification, the client must positively prove his identity, often in person to the RA. Due to the importance of the CA, the operations of a CA are often governed by legislation [28].

b. Key Recovery

Once a client's public-private key-pair is in use, the client may face issues of having misplaced the private key. In such situations, there is a need to have in place mechanisms for key back-up and/or recovery. This responsibility remains with the CA/RA; a mutual agreement with the client about the proper procedures to follow in such an event should be established during issuance. The notion of a private key being kept by a second party other than the client may prove to be the undoing of the reliability of the said key for non-repudiation. A work-around involves each client having two sets of key-pairs: one set for confidentiality, i.e., encryption/decryption of data purposes; and one set for integrity, i.e., digital signature purposes. The private key used for decryption of data may then be kept by a trusted third party.

c. Certificate Revocation

While the validity of a certificate is checked during an authentication transaction, there may be events that lead to the premature invalidation of a public key, rendering a digital certificate void. Beyond natural expiration of a key and the need for regeneration, there is also a requirement that a mechanism be in place that ensures that the relying party in a transaction be able to retrieve data stating that the said public key has not been invalidated prematurely. This check on invalidation is made through comparison against a CRL, generally maintained by the issuing CA. Should a certificate be found to be on a CRL, dependence on the certificate is immediately suspect. Proper distribution infrastructure must be put in place for the proper dissemination of the CRLs to the relying parties. Protocols that support revocation include the Online Certificate Status Protocol, and the Simple Certificate Validation Protocol [29]. The format of the CRL is defined by the X.509 standard.

C. IDENTITY FRAMEWORK (NON-REPUDIATION NON-SUPPORTING)

1. Overview

The current state-of-the-art in NRNS frameworks can be divided into two opposing camps [30], user-centric and federated. User-centric frameworks, as the name suggests, place the user in a position optimized for user control of the authentication process: the user is the initiator as well as conduit for the authentication process information flow. In this form, user privacy is greatly enhanced. Implementations that can be classified as user-centric are OpenID and a collective group known as Infocard. Federated frameworks were designed to provide SSO functionality within a circle of pre-established trust relationships, or a federation. The user continues to play an active, though relatively diminished, role in federated frameworks: interactions between AOs and SPs may pass over the user in favor of the already established connections between the two. The two leading standards supporting federation are the Security Assertion Markup Language (SAML) v2.0 and Web Services Federation (WS-Federation) standards. Each standard is detailed below.

2. OpenID

The OpenID framework was developed based upon the principle whereby a user may authenticate through presentation of a Web resource that the user controls [31]. Simply put, authentication is dependent upon the user proving ownership of the said resource [32]. OpenID is an evolving framework with widespread adoption on the Internet; notable AOs include those of Google and Yahoo, with Facebook and MySpace as implementers. Its popularity stems from its simplicity of use; no additional software or hardware is required—at least, not in its present form—and there is no requirement for a password. Currently, in v2.0 of its implementation, responsibility for development of OpenID is undertaken by the community-based OpenID Foundation. Built upon OASIS standards (XRI, XRDS), OpenID itself has yet to achieve widespread peer recognition as a standard. However, it has attained U.S. approval for use as an identity authentication framework in support of low-risk transactions; all derived data from the authentication are to be treated as unreliable [33]. This assurance classification is due in part to the documented susceptibility of the framework to phishing and man-in-the-middle attacks [30].

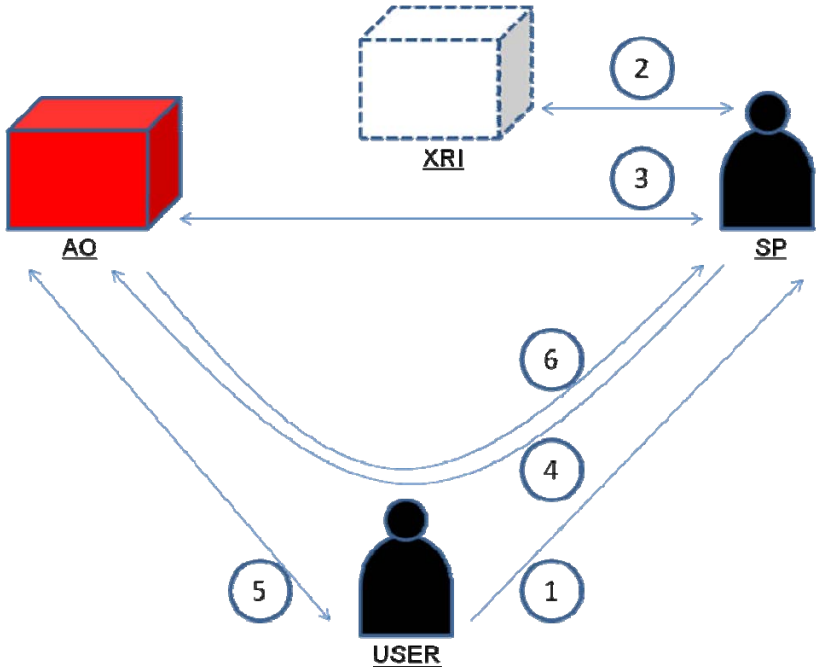

Figure 7. OpenID Authentication Process Flow Diagram. After [32]

Figure 7 depicts the authentication process in OpenID. A Uniform Resource Locator (URL) address or Extensible Resource Identifier (XRI-ID) is used as a user's identifier. A user may obtain the identifier through pre-registration with an AO. Once at the SP's login portal, the user presents the said identifier, shown in step one. In step two, the SP evokes the XRI Resolution Protocol (XRI-Res) to discover the identity of the supporting AO based on the submitted identifier, followed by the establishment of a shared secret between the two, as shown in step three. The user is then re-directed to the AO, as in step four, followed by authentication with the AO, as in step five. Note that this authentication in itself can leverage any of the token types previously mentioned in Chapter II. Once verified, the user is redirected back to the SP with an assertion from the AO, as in step six [32].

3. Infocard

Infocard refers to a collection of proposed implementations that are based on the notion of the submission of an Information Card (I-card) to enable authentication. The I-cards are stored in and selected from a repository known as a *card selector,* software that manages all of a user's I-cards. Leading implementations of the Infocard methodology include Microsoft's CardSpace and the InCommon framework. To date, CardSpace has been included with recent versions of Windows [34], while the InCommon framework is undergoing reviews for use for e-government purposes in the United States [35]. CardSpace itself leverages upon the Web Services (WS*) suite of standards as its infrastructure, though CardSpace itself has yet to be accepted as an industry standard [36]. With CardSpace as the dominant implementation for the Infocard framework, it will be used as the basis for the rest of the discussion.

As can be seen in Figure 8, in CardSpace, each time a user accesses a service requiring authentication, the SP will reply with a request for a security token, as shown in step one. The user selects the appropriate security I-card from the *card selector* software and presents this card as the security token to the SP, as shown in step two. Two types of

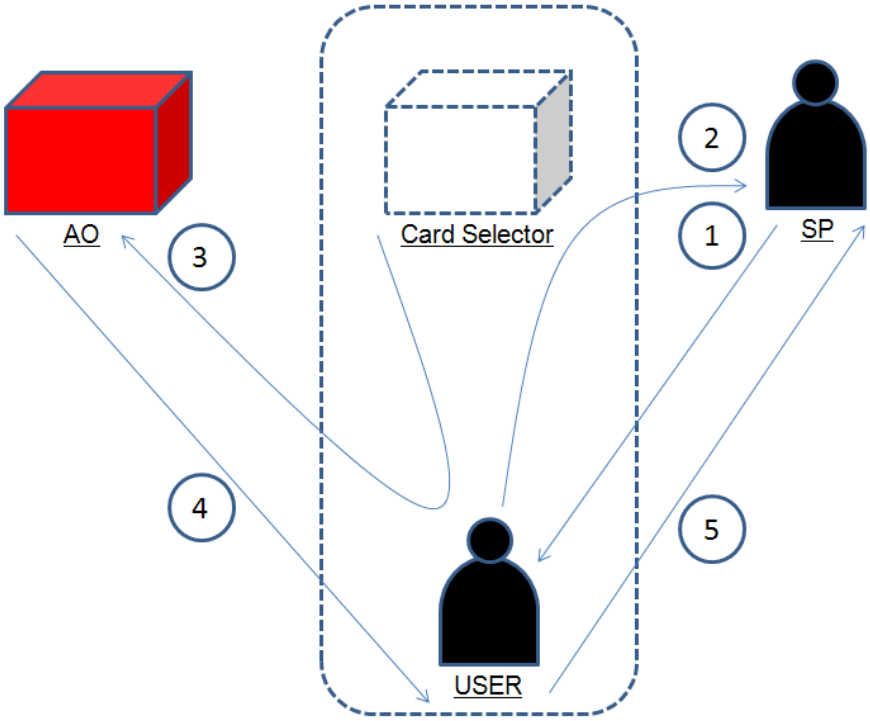

Figure 8. CardSpace Authentication Flow Process. After [36]

cards may be submitted. A Personal Card is a self-asserted/created card containing details that will facilitate authentication, such as a user ID and password combination. A Managed Card is one that is provided by an AO; it typically requires a secondary authentication step to be undertaken between the AO and the user, as shown in step three, before a token is made available by the AO, shown in step four. The said token is reviewed by the user and then forwarded to the SP, shown in step five. If the token is accepted by the SP, the user is then granted access to the service, completing the authentication process.

Because the card selector is a single repository for all of a user's digital identities, it is a prime target for the determined attacker. However, it is also claimed that the card selector may decrease its susceptibility to phishing through warnings should the user transact with a new service, or through customized tokens that are only useful to the SP and no others, thereby preventing a replay attack [35].

4. Security Assertion Markup Language

SAML, at its core, is not strictly an authentication framework. It is, as the name suggests, a standard that dictates how assertions are to be formatted for transmission between parties in a form that is non-proprietary. This can even be leveraged by user-centric frameworks. Building upon this base specification, though, are components that, when put together, define an eponymous composite standard for authentication. First defined in November 2002, the standard has since seen a major revision, emerging as SAML v2.0 in 2005. This new version incorporates inputs from then-leading implementations of Federated Identity, including those of the Liberty Alliance and the Internet2 Shibboleth project. It is the only specification to have received recognition as a federated identity management standard by the Organization for the Advancement of Structured Information Standards (OASIS).

The components previously mentioned, as captured in Figure 9, collectively define the relationship and possible interactions between the user, AO, and SP. At its core is the assertion component, the primary construct that serves to affirm the identity of a user. Each assertion contains authentication, attribute, and authorization decision statements; each statement is composed of data necessary for authentication sandwiched between standardized XML tags. The protocol component defines the SAML protocols that are leveraged upon for interactions via a request/response mechanism. The bindings

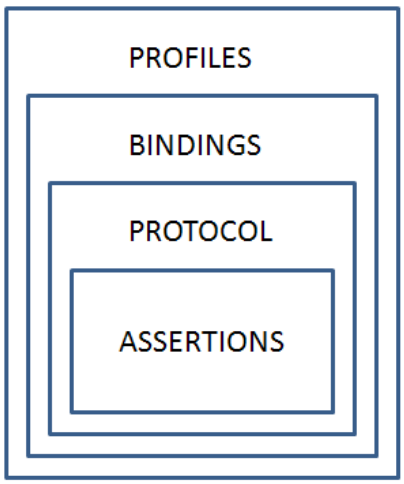

Figure 9. SAML v2.0 Component Affiliations.After [37]

component provides the linkage between SAML protocols and the underlying non-SAML transport layer protocols. The profile component specifies how combinations of assertion, protocol, and binding components are amalgamated to support a specific function. It is this profile component that provides a tangible expression of how SAML can be leveraged to support authentication, this expression differentiates SAML from non-SAML frameworks that only use the base assertion component in their specifications. The Web Browser SSO Profile will be used to illustrate SAML's use.

As is shown in Figure 10 step 1, the user visits the SP site to access a service. As the user has not been authenticated, the user is redirected to the AO for authentication, depicted by step 2. In step 3, the user authenticates to the AO through a pre-established authentication means. Step 4 sees the AO build an assertion that signifies that the user's identity has been affirmed upon verification. The user is thereafter directed back to the SP by the AO with the assertion in tow, completing the authentication process, as in step 5.

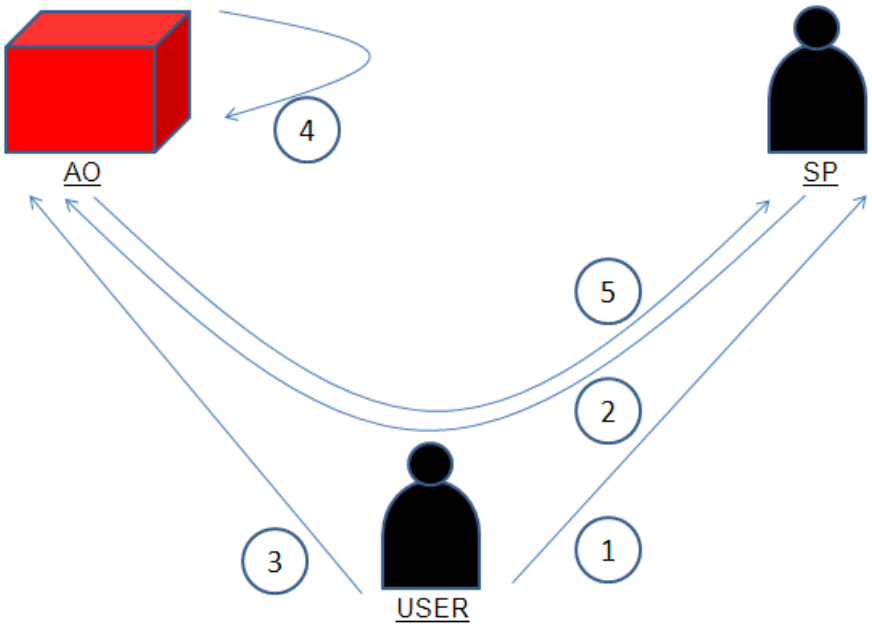

Figure 10. SAML Authentication Process Flow Diagram

5. WS-Federation Language

WS-Federation Language (WS-F) was developed to provide SSO-federated functionality as part of the WS* suite of specifications. It defines how authentication,

authorization, attribute, and pseudonym services can be effected via reliance on an underlying Security Token Service (STS) [38]; STS itself is a specification of WS-Trust, a mechanism developed for the management of security tokens or assertions. The services that WS-F offers are similar to those of SAML but optimized for a Web-services environment. The WS* suite of standards was developed jointly by a consortium of industry heavyweights, including Microsoft, IBM, and BEA, to meet the challenges of Internet 2.0 and the growing interest in systems developed based upon the Service-Oriented Architecture (SOA). While core specifications upon which WS-F is reliant have achieved OASIS recognition (WS-Security, WS-SecurityPolicy, WS-Trust), WS-F itself has yet to achieve standardization. As WS-F relies on STS, the WS-Trust model will be used to illustrate how authentication is performed using the WS-F standard.

Figure 11 depicts the STS authentication process [39]. A user requires a service from an SP residing in a separate trust domain. To access the desired service, he willrequire a security token from his AO(User)/STS provider and an access token from the AO(SP)/STS provider that governs/resides in the target trust domain which the SP recognizes. The user proceeds to obtain his security token from AO(User), as seen in step one. This is achieved through a pre-determined authentication means. Because both AOs have an on-going trust relationship, the security token presented to the other AO by the user is accepted in step two, and an access token is granted to the user in step three. The access token is subsequently presented to the SP in step 4 for access to the specified resource.

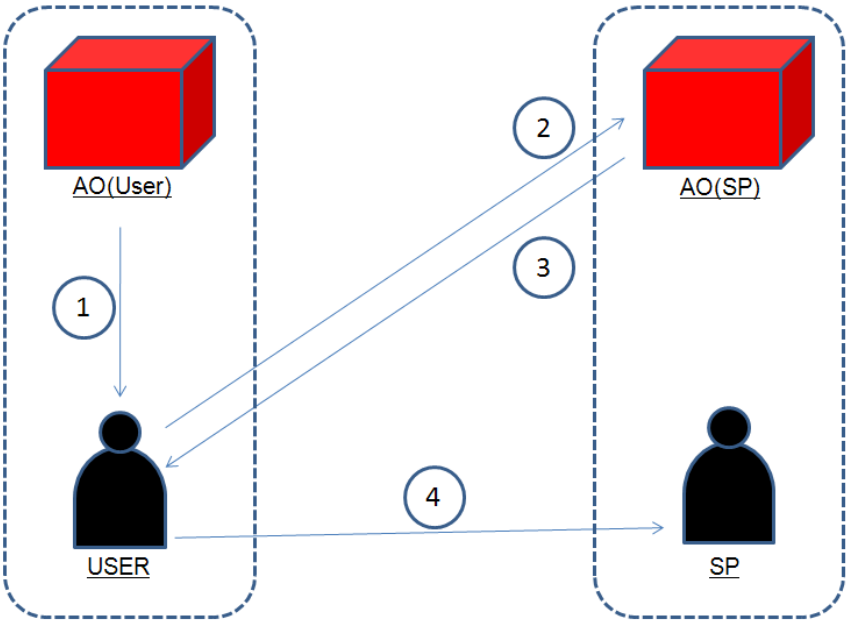

Figure 11. WS-F authentication Process Flow Diagram [After 39]

6. Evaluation Criteria

To determine the best-fit framework in support of the NRNS component of the NAF, the following criteria have been derived, based upon requirements set forth by the NAF.

a. Fit-for-purpose

The selected framework must be fit-for-purpose, supporting the deployment characteristics of the NAF. These include supportability for NRNS tokens, and use of the NAF as a second authentication factor to the established SINGPASS infrastructure.

b. Standards-based

The selected framework must be based upon established standards. The availability of a certification service is an advantage. Differentiation is made with dependence upon a standard and actual accreditation.

c. Interoperable

The selected framework must be able to accommodate and integrate with existing and future standards/frameworks.

d. Ease-of-use

The selected framework must be usable for the majority of the population, presenting the least inconvenience in terms of hardware and software reliance.

7. Evaluation and Analysis

Each framework was assessed based on the stated criteria, with the results summarized in Table 2.

a. OpenID

OpenID, as a complete authentication protocol standard, will add an additional layer of complexity when integrated into the existing SINGPASS system. A user will not only have to supply his SINGPASS password but additionally will have to input the required URL/XRI-based identifier and any subsequent token required of the authentication process proper. Current implementations rely upon passwords for authentication. Although this does not exclude the option of using other tokens, the dearth of any actual implementations will continue to render OpenID suspect in this area. While OpenID is based upon XRI, OpenID itself, as a standard, lacks third-party accreditation. This deficiency may be mitigated by the open-source nature of the standard, akin to JAVA, whereby its advancement is dependent upon a community of developers. In terms of interoperability, not being XML-based, as compared to SAML and WS-F, may limit its extensibility as well as interoperability. The additional complexity involved should OpenID be adopted may nullify its simplicity and ease-of-use, as will the possible use of hardware tokens for authentication. URLs and XRIs are also not easily memorized and may require a secondary management tool, an I-card being one option.

b. Infocard

Hardware tokens do not have a software-based alternative that can be implemented as an I-card, nor can hardware tokens be supported as a Managed Card. In its current implementation, Infocard only supports the X.509 and Kerberos standards [36]. This demands a separate framework be implemented in addition to Infocard to manage hardware tokens. That Infocard is built upon WS* standards allows for some interoperability. The usability and corresponding ease-of-use of a card selector is dependent upon the actual interface presented. The requirement for the installation of card selector software is a substantial additional requirement when compared to the other frameworks but may well serve as a convenient tool for the management of digital certificates as a managed card, complementing PKI.

c. SAML

SAML does not specify the authentication method but provides a means whereby assertions can be transmitted between AOs, thus supporting one of the key required characteristics of the use case in Chapter II. It is an established standard, endorsed by OASIS, and adopted by numerous countries [40]–[42] as part of their identity management infrastructure. Being XML-based, SAML is expected to be interoperable with like systems. An official certification body also exists to ensure products comply with the SAML standard. There exist specifications that integrate SAML with foundational WS* standards other than WS-F. The use of SAML is transparent to the user, leveraging on currently existing HTTP request/redirect protocol for data exchange.

d. WS-F

WS-F provides similar functionality to SAML; although it has yet to be ratified by OASIS, it is in the process. The complexity involved in realizing WS-F, being reliant on numerous foundational building blocks, permits it greater extensibility, not least of which is its Web services nature. Products offering WS-F will also implement SAML; that implementation does remove the differences between the two to a degree

[43]. The use of WS-F is likewise transparent to the user, but is more suited to applications that evoke Web services, an architecture that has yet to be fully established, though is expected to be further developed in the future [44].

Table 2. Authentication Framework Comparison

	OpenID	**Infocard**	**SAML**	**WS-F**
Fit-for-purpose			Y	Y
Standards-based	Y	Y	Y	Y
Accredited	Open-Source		Y	In-process
Interoperable			Y	Y
Ease-of-Use		Dependent	Y	Y

From the results of the analysis, it is evident that SAML is the framework of choice, with WS-F as a credible alternative.

D. SUMMARY

In this chapter, the characteristics of various identity frameworks have been detailed, discussed, and compared as a means of selecting the best framework to implement the National Authentication Framework. Inherent characteristics of the selected frameworks will serve as the basis for infrastructure deployment, system implementation, and future development, underlining the importance of ensuring that the selected framework not only meets current requirements but also will be sufficiently extensible to meet future needs. Public Key Infrastructure is a well-established framework that will provide for non-repudiation. For non-repudiation non-supporting authentication, SAML is the framework of choice. The next chapter will delve into greater detail as to how SAML will realize the baseline use case.

THIS PAGE INTENTIONALLY LEFT BLANK

V. NAF IMPLEMENTATION MODEL

A. OVERVIEW

Chapter IV established the basis for the selection of SAML as the standard for the implementation of the non-repudiation non-supporting infrastructure of the NAF. The realization of the use case as specified in Chapter II will be based upon features organic to SAML. This chapter will provide details as to how each selected feature will realize a certain aspect of the use case, and will conclude with an illustration of how the various features converge to realize the use case.

B. NAF SAML PROFILE

1. Overview

At its highest layer of abstraction, SAML specifies profiles to fulfill certain functionalities required during authentication. The Web Browser SSO Profile (WBSP), which dictates the interactions between the user, SP, and AO in an authentication process, will first be described, as per the prescribed standard as endorsed by OASIS. Thereafter, key features and attributes of SAML necessary to enhance the functionality of the WBSP for the NAF will be discussed.

2. Web Browser SSO Profile

The SAML v2.0 Technical Overview [45] describes how a user authenticates to an SP via the WBSP. Underlying the transaction is the ubiquitous Hypertext Transfer Protocol (HTTP), wherein its relationship with SAML is specified as part of the SAML Bindings abstraction. The choice of an SP-initiated process is in sync with current practice whereby a user navigates to the SP page to initiate authentication. This is opposed to an AO-initiated process, where the user navigates first to an AO, prior to accessing the SP. The WBSP, as extracted from the SAML standard, is detailed below (Figure 12).

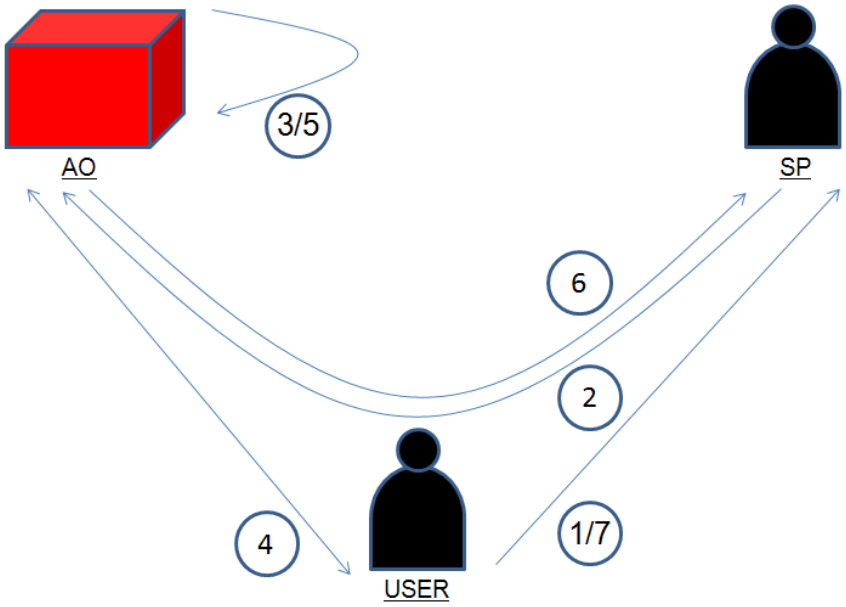

Figure 12. Web Browser SSO Profile. After [45]

A user attempts to access a resource but is denied by the SP pending authentication, shown in step one. In step two, the SP sends an HTTP Redirect response to the user's Web browser, specifying the Uniform Resource Identifier (URI) of the AO, appended with an <AuthnRequest> message encoded as a URL query variable. The user's browser, upon receipt of the HTTP Redirect response, issues an HTTP Get request to the AO's authentication service, with the same <AuthnRequest> appended. Step three sees the AO interpret the <AuthnRequest> to determine the specific requirements of the SP for authentication and thereafter proceeds to authenticate the user through a token presented by the user, as shown in step four. Upon positive authentication in step five, the AO proceeds to build an assertion and returns this assertion, placed in a <Response> message as an HTML form to the user's browser. This HTML form is subsequently sent to the user as a redirect to be forwarded to the relying SP as an HTTP POST request, depicted in step six. The SP validates the authenticity of the assertion and thereafter proceeds to permit access to the resource as requested by the user, in step seven.

3. Proxy Authentication Operator

The <AuthnRequest> message, as depicted in the previous section, is the key message sent by the SP to initiate an authentication request to an AO. The <AuthnRequest> message contains substantial information specifying the authentication context required in order for the user to gain access to the requested service. Critical contextual factors include the level or strength of the authentication method, as specified by the <RequestedAuthnContext> subfield, the SP-defined user's transactional ID as specified in the <NameIDPolicy> subfield, and the list of recognized AOs with regards to the SP, as specified in the <Scoping> subfield. A notable characteristic of the use case developed in Chapter II is the ability of the SP's AO (AO(SP)) to seek user authentication from the user's AO (AO(User)). This capability is realized by specifications enclosed within the <Scoping> subfield, which allows for the actual authentication of the user to be transferred to a secondary AO, a transaction known as *proxying*.

Since AO(User) may not be similar to the AO(SP), there is a need for the AO(SP) to seek an assertion from AO(User) and, thereafter transfer the assertion generated by the AO(User) to the SP. This is realized through AO(SP) generating an <AuthnRequest> message of its own and once more redirecting the user to AO(User). The assertion generated from AO(User) is first received by the AO(SP), which proceeds to extract the assertion from the HTTP Post request, repackage the assertion into a form recognizable by the SP, and thereafter forward the assertion once more as an HTTP Post request. This functionality is only possible if certain conditions are met within the <Scoping> subfield. The first is for the <ProxyCount> subfield within that of <Scoping> to be set to more than 0. This figure, as obtained from the SP, specifies the number of proxy operations allowed by the SP for authentication. The user's AO must also reside in a list documented in the <IDPList> subfield as the list of AOs that the SP trusts to provide authentication. The token used by the AO also has to conform to the minimal standard imposed by the <RequestedAuthnContext> element as stated in the <AuthnContextClassRef> and its <Comparison> subfield. The <AuthnContextClassRef> defines the type of token with the least amount of assurance acceptable for the authentication. The <Comaprison> field has

a value of “minimum,” indicating that the actual authentication token used cannot be lower in assurance than that set in <AuthnContextClassRef>. The relationship between the messages and subfields described in this paragraph are summarized in Figure 13.

<AuthnRequest>
<saml : Subject>
<NameID>
<SPNameQualifier>
<SPProvidedID>
<NameIDPolicy>
<Format>
<RequestedAuthnContext>
<AuthnContextClassRef>
<Comparison>
<Scoping>
<ProxyCount>
<IDPList>

Figure 13. <AuthnRequest> Fields Extract. After [46]

4. Transient Identifier

An identifier is attached to an <AuthnRequest> to associate the user, or subject, of the authentication. To ensure user privacy, a one-time use transient identifier is proposed over the use of a persistent identifier. A persistent identifier, while limited to only between that of the SP and the AO, may allow for the unwarranted collation of past and present authentication attempts by the user. A transient identifier, being single-use, is only valid for one instance of an authentication transaction and requires regeneration for each subsequent transaction. With reference to Figure 13, the transient identifier is a pseudorandom number generated by the SP and appended to an <AuthnRequest> via the <SPProvidedID> field. The SP also sets the <SPNameQualifier> with its own unique ID, and specifies that the identifier used is a transient one in <Format>. The setting of the <Format> field allows the receiving AO to know that the <SPProvidedID> is transient in nature and cannot be relied upon for subsequent authentication requests, where a new identifier will be expected. In a proxying relationship, the same <SPProvidedID> is expected to remain constant for the instance of the authentication transaction.

5. Response

The AO, having verified a user's submitted token, issues the assertion in a <Response> message, which is subsequently redirected to the requesting SP. A <Response> message is essentially a carrier for the assertion. The assertion will contain the following attributes as captured in Figure 14: <Version> states the version of the assertion, by default being 2.0. The <ID> is a unique one-time randomly generated identifier for the specific assertion, minimally 128 bits in length. <IssueInstant> states the time that the assertion was generated. <Issuer> captures the identifier of the submitting AO. <ds:Signature> is required for authentication of the assertion itself, originating from a credible source through the application of a digital signature. <Subject> contains the transient identifier issued originally by the requesting SP.

<Response>
<Assertion>
<Version>
<ID>
<IssueInstant>
<Issuer>
<ds:Signature>
<Subject>

Figure 14. <Response> Fields Extract. After [46]

C. USE CASE REALIZATION

This section depicts the basis use case, developed in Chapter II, being realized by components of SAML v2.0.

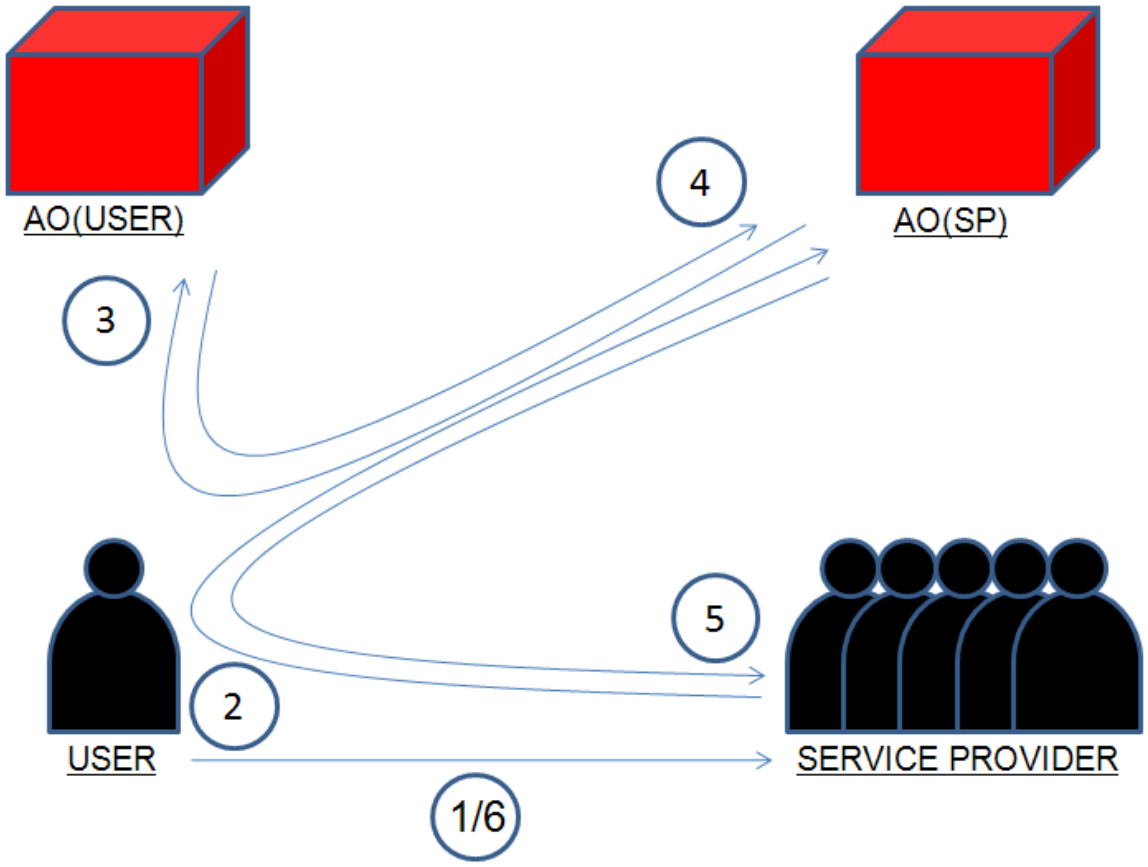

Figure 15. Authentication Use Case

1. Conversation #1: The user submits the first legacy authentication token (SINGPASS password) to the SP, which proceeds to authenticate this first token. Should the token be invalid, the connection is immediately terminated.

2. Conversation #2: Should the token be valid, the SP issues an HTTP Redirect response to the user's browser (Browser(U)), directing the user to AO(SP) with an <AuthnRequest> message appended. The AO(SP) authentication server receives the <AuthnRequest>; from the analysis of the <IDPList> field, it determines the acceptable AOs that the SP authorizes for the generation of an assertion and generates a Web Page that allows the user to select the appropriate authorized AO. Note that the AO(SP) would be in this list as well because the AO(SP) would not have any indicator regarding

whether any particular user has a relationship with it, and AO(SP) therefore cannot automatically initiate the establishment of a second-factor authentication process with the user.

3. Conversation #3: The user selects the appropriate AO(User). If AO(User) is AO(SP), then AO(SP) will continue with authentication and the generation of the assertion. If AO(User) is another entity, the AO(SP) will now transit into the role of a proxying AO and issue an HTTP Redirect response once more to Browser(U) and direct the user to AO(User) with an <AuthnRequest> appended.

4. Conversation #4: The user proceeds to submit his token and any other necessary attributes (e.g., identifier unique to AO(User)) to the AO(User) for authentication. The AO(User), having verified the identity of the user, will generate an assertion in a <Response> message and return the user to AO(SP) via an HTML form.

5. Conversation #5: AO(SP) extracts the assertion and repackages it into a form unique to itself and the relying SP, and returns the assertion in a <Response> message via an HTML form to the SP. Note that AO(User) remains as the originator of the assertion. AO(SP) merely edits the necessary fields in the <Response> message to enable communication between it and the SP.

6. Conversation #6: The SP, having received the assertion, proceeds to allow the user access to the requested service.

D. SUMMARY

In this chapter, it has been shown that SAML has many built-in features that support the realization of the use case. Among the more pertinent features are the ability to proxy an authentication request, to define a transient identifier, and to dictate the AOs and token type assurance levels suitable to the supported transaction. That SAML also defines profiles encompassing supporting protocols shows how it can be easily realized with current technology. SAML is sufficiently robust to meet the requirements of future challenges, being organically highly extensible. The choice of SAML as the basis standard ensures that the NAF will be effectively and efficiently realized.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION

A. OVERVIEW

The NAF is an ambitious project, not just in terms of scale, but also in terms of scope. There has yet to be a similar effort at implementing a national two-factor authentication system with the amount of choice available to the user. In so doing, several important lessons can be derived which will be elaborated upon in this chapter. The chapter will conclude with suggestions for further research into this novel effort.

B. LESSONS LEARNED: COMPROMISES

In stating the requirements for the NAF, two elements stand out that have had broad implications on the eventual implementation model. Firstly, the NAF is to be implemented in concert with the operational single-factor authentication system. Secondly, non-repudiation is a criterion for the system. This section will detail the implications of these two system requirements.

1. Co-existence

The legacy password-based SINGPASS authentication system, still currently in use, will continue to serve as the first authentication factor. Support for SINGPASS remains the domain of a single operator, contracted by the government for its operation. While the operation of SINGPASS remains distinct from the SAML-based second-factor authentication, its continued existence implies that, in the short term, the NAF would play, at best, a complimentary role. That SINGPASS, if kept in its current form, diminishes the practicality of implementing a non-SAML NAF framework; for instance, the implementation of OpenID (non-SAML) would require the user to input a second what-you-know identifier in addition to that already required for SINGPASS. The benefits of the NAF to the user are also diminished, as the user would still be required to use the services of both the government-dictated Authentication Operator (AO) for SINGPASS, and another AO for the second factor. The operator for SINGPASS is also

placed at a competitive advantage over its peers. Being the long-standing solution, it not only has the advantage of reputation, but will be well placed to offer synergistic services incorporating both the first and second authentication factors in spite of any technical and operational differences between SINGPASS and the NAF. While the continuance of SINGPASS may have very much been a contractual issue, that the government continues to leverage on a proprietary legacy system while proposing and pushing for an enterprise one may also diminish the acceptance of the NAF in the eyes of possible private-sector users, particularly the banking industry, which already has 2FA authentication systems in place. This may result in the authentication landscape remaining status quo for the user except for the addition of yet another token for purposes of government services. It is recognized that SINGPASS would play a role during the transition phase as part of the roll-out of the NAF. Once the implementation issues for the NAF have been resolved and the schedule put in place for its roll-out, however, it would be helpful for the government to announce the road map for the eventual dissolution of SINGPASS and for the full transition to the NAF, wherein both tokens used for authentication would be a result of the user's choice. That would provide the impetus for wider adoption of the NAF and the full realization of the system's stated objective.

2. Non-repudiation

Among the requirements set forth for the NAF is that of non-repudiation for access to certain services. This requirement for non-repudiation, though entirely logical, immediately limits the type of token used to that of the asymmetrical cryptographic type. With such a requirement, each user of government services will be required to have in his possession an asymmetric cryptographic token, making such a token the default choice for users and thus limiting the take-up rate of other token types. This brings up the relative costs of using other token types, which may make owning other tokens financially and practically infeasible. This remains, in public policy parlance, a wicked problem. The requirement for non-repudiation cannot be rescinded easily, as doing so would result in an Integrity issue. Yet continuing with the existence of the requirement for non-repudiation would limit the market for non-asymmetric cryptographic token

types. It is then best left to the academic community to devise an alternate token type that also provides non-repudiation as a direct competitor to that of an asymmetric cryptographic token.

C. FUTURE RESEARCH

As of December 2009, the NAF is in its consultative stage whereby comments and proposals are being sought from both industry and academia. One area in which work will be needed is a study of the actual implementation model that is adopted and the basis for the selection of the various underlying technologies that support the implemented model. Next, a post-implementation review of the implemented system would also yield important lessons that would be useful for other organizations that may wish to embark on a similar type project.

D. SUMMARY

The aim of this thesis was to propose an implementation model for the NAF. A baseline use case was first developed to outline the relationships between the user, the SP and the AO, to be used as the main reference in the application of supporting technologies. Thereafter, a survey of current token types was conducted, with each token type being assessed for its suitability for use in the NAF, susceptibility to security threats, and provision of non-repudiation being major criteria. The asymmetric cryptographic token type has been selected for use where non-repudiation is required, and the OTP and Out of Band token has been selected for use where non-repudiation is not required. Following, five leading identity frameworks were assessed for their suitability. The frameworks were assessed based on whether they were fit-for-purpose, were standards-based, were interoperable, and were easy-to-use. SAML emerged as the framework of choice where non-repudiation was not required, while PKI would serve as the framework where non-repudiation was required. How SAML would be used to meet the requirements of the AO was also described in detail, particularly the ability for proxy authentication and the use of transient identifiers. It was observed that several requirements for the NAF have had an asymmetric effect on the eventual implementation

model, particularly the need for co-existence with a legacy system and the need for non-repudiation. It has been suggested that the legacy system be retired as early as possible and that an alternative token type, which would provide non-repudiation, be worked on. As part of future research, it would be extremely instructive to continue to follow the development of the NAF to its conclusion, because much can be learned and applied to subsequent exercises of the same nature.

LIST OF REFERENCES

[1] Infocomm Development Authority of Singapore. (2008, June). Fact Sheet: National Authentication Framework. Available: http://www.ida.gov.sg (accessed June 20, 2009).

[2] Infocomm Development Authority of Singapore. (2008, October). National Authentication Framework (NAF) Call-for-Collaboration Main Public Document. Available: http://www.ida.gov.sg (accessed June 20, 2009).

[3] *Recommended Security Controls for Federal Information Systems and Organizations*, SP 800-53-3, August 2009.

[4] S. Harris, *CISSP: Exam Guide*, 4th ed., New York: McGraw-Hill, 2008, pp. 1145.

[5] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth-factor authentication: Somebody you know," in *Proc of the 13th ACM Conf on Computer and Communications Security*, 2006, pp. 168–78.

[6] R. Sreenivasan and M. Tan, "Cracking the online vault," October 2002. Available: http://www.lawgazette.com.sg/2002-10/Oct02-focus2.htm (accessed July 10, 2009).

[7] DBS, "DBS banking achieves another first in the region," May 2003 Available: http://www.dbs.com.sg/newsroom/2003/Pages/press030513.aspx (accessed July 10, 2009).

[8] Federal Financial Institutions Examination Council, "Authentication in an Internet Banking Environment," October 2005. Available: http://www.ffiec.gov/pdf/authentication_guidance.pdf (accessed July 12, 2009).

[9] B. Schneier, "The failure of Two-Factor Authentication," March 2005. Available: http://www.schneier.com/blog/archives/2005/03/the_failure_of.html (accessed June 27,2009).

[10] *Electronic Authentication Guideline Rev. 1*, SP 800-63, December 2008.

[11] Australian Government Information Management Office. (2009, January). Better Practice Guideline 3: Implementation Models. Available: http://www.finance.gov.au/e-government/index.html (accessed June 25, 2009).

[12] VASCO, "Sovereign bank enables and protects its customers with VASCO's DIGIPASS 260," October 2009. Available: http://www.vasco.com (accessed October 20, 2009).

[13] VASCO, “The Swedes bank on e-security,” October 2009. Available: http://www.vasco.com (accessed October 20, 2009)

[14] State Services Commission. (2004, April). Authentication for e-Government: Best practice framework for authentication. Available: http://www.e.govt.nz/ (accessed June 25, 2009).

[15] Australian Government Information Management Office. (2009, January). National e-Authentication Framework. Available: http://www.finance.gov.au/e-government/index.html (accessed June 25, 2009).

[16] VASCO, “E-signatures for transaction validation and document signing,” November 2009. Available: http://www.vasco.com/solutions/solutions/e-signatures.aspx (accessed October 20, 2009).

[17] R. Ayoub, “An overview and competitive analysis of the OTP market,” June 2009. Available: http://www.rsa.com/press_release.aspx?id=10278 (accessed August 9, 2009).

[18] Verisign, “National PKI: The foundation of trust in governmental programs,” March 2009. Available http://www.verisign.com/authentication/index.html (accessed August 14, 2009).

[19] Cardlogix, “Microprocessor cards,” November 2009. Available: http://www.cardlogix.com/products/cards/smart/microprocessor.asp (accessed October 10, 2009).

[20] Cardlogix, “Card readers & terminals,” November 2009. Available: http://www.cardlogix.com/products/readers/readers.asp?mfg=Omnikey&action=Go (accessed October 10, 2009).

[21] Almex Ltd, “Smart card products,” October 2009. Available: http://www.smartcardsource.com/cards.html (accessed October 10, 2009).

[22] British Telecommunications, “21CN voice authentication,” October 2009. Available: http://www.globalservices.bt.com/InsightsDetailContentAction.do?Record=21cn_article_all_en-gb (accessed October 7, 2009).

[23] J. Gold Associates, “Choosing an enterprise-class wireless operating system: A comparison of Blackberry, iPhone, and Windows Mobile,” February 2009. Available: http://www.jgoldassociates.com/recentresearch.html (accessed July 17, 2009).

[24] J. Evers, “PayPal to offer password key fobs to users,” January 2007. Available: http://news.cnet.com/2100-7355_3-6149722.html (accessed September 7, 2009).

[25] R. Perlman and C. Kaufman, "User-centric PKI," presented at *IDtrust '08: Proc of the 7th Symp on Identity and Trust on the Internet.* Available: http://doi.acm.org/10.1145/1373290.1373300 (accessed November 20, 2009).

[26] *Internet X.509 Public Key Infrastructure certificate and Certificate Revocation List (CRL) Profile*, RFC5280, May 2008.

[27] *The Transport Layer Security (TLS) protocol version 1.2*, RFC5246, August 2008.

[28] Infocomm Development Authority of Singapore, "Salient features of the electronic transactions act (CA regulations)," October 2009. Available: http://www.ida.gov.sg/home/index.aspx (accessed November 20, 2009).

[29] C. Adams, S. Lloyd, and C. Adams, *Understanding PKI : Concepts, Standards, and Deployment Considerations*, 2nd ed., Boston: Pearson, 2003.

[30] S. Rieger, "User-centric identity management in heterogeneous federations," in *Fourth International Conf on Internet and Web Applications and Services*, Venice, Italy, 2009, pp. 527–532.

[31] D. Reed, L. Chasen, and W. Tan, "OpenID identity discovery with XRI and XRDS," in *Proc of the 7th Symposium on Identity and Trust on the Internet*, Gaithersburg, MD, 2008. Available: http://doi.acm.org/10.1145/1373290.1373294 (accessed September 23, 2009).

[32] D. Recordon and D. Reed, "OpenID 2.0: A platform for user-centric identity management," in *Proc of the Second ACM Workshop on Digital Identity Management*, Alexandria, VA, 2006. Available: http://doi.acm.org/10.1145/1179529.1179532 (accessed September 23, 2009).

[33] T. McBribe, D. Silver, M. Tebo, C. Louden, and J. Bradley, "Federal identity, credential and access management: OpendID profile 2.0," September 2009, http://www.idmanagement.gov/ (accessed September 23, 2009).

[34] W. Alrodhan and C. Mitchell, "Improving the security of cardspace," in *EURASIP Journal on Information Security*, February 2009 Available: http://www.hindawi.com/journals/is/2009/167216.html (accessed September 25, 2009).

[35] D. Thibeau and R. Drummond, "Open trust frameworks for open government: enabling citizen involvement through open identity technologies," August 2009. Available: http://openid.net/government/ (accessed September 25, 2009).

[36] V. Bertocci, G. Serack, and C. Baker, *Understanding windows cardspace: an Introduction to the concepts and challenges of digital identities*, Boston, Pearson, 2008.

[37] J. Hughes and E. Maler, “Security Assertion Markup Language (SAML) 2.0 technical overview,” February 2005. Available: http://www.oasis-open.org (accessed November 5, 2009).

[38] M. Goodner, M. Hondo, A. Nadalin, M. McIntosh, and D. Schmidt, “Understanding WS-Federation,” May 2007. Available: http://msdn.microsoft.com/en-us/library/bb498017.aspx (accessed November 7, 2009).

[39] H. Lockhart, S. Andersen, J. Bohren, Y. Sverdlov, M. Hondo, H. Maruyama, A. Nadalin, N. Nagaratnam, T. Boubez, K. Morrison, C. Kaler, A. Nanda, D. Schmidt, D. Walters, H. Wilson, L. Burch, D. Earl, S. Baja, and H. Prafullchandra, “Web Services Federation Language (WS-Federation),” December 2006. Available: http://www.ibm.com/developerworks/ (accessed November 7, 2009).

[40] Liberty Alliance, “The Finnish national board of taxes makes a business case for e-authentication,” September 2009. Available: http://www.projectliberty.org/ (accessed November 9, 2009).

[41] S. Nielsen, “Memorandum on the reasons for selecting SAML 2.0 as recommended standard for federated identity and access management in the Danish public sector,” February 2006. Available: http://www.itsk.dk (accessed November 5, 2009).

[42] D. Silver, T. McBride, M. Tebo, T. Farrales, S. Lazerowich, and C. Louden, “E-authentication federation adopted schemes,” May 2007. Available: http://www.idmanagement.gov/eauthentication/documents/EAuthFederationAdoptedSchemes.pdf (accessed November 7, 2009).

[43] Organization for the Advancement of Structured Information Standards, “Microsoft 'Geneva' framework supports SAML 2.0, WS-F and WS-trust,” October 2008. Available: http://xml.coverpages.org/ni2008-10-29-a.html (accessed November 5, 2009).

[44] Computer Economics, "SOA adoption surges," January 2009. Available: http://www.computereconomics.com/article.cfm?id=1423 (accessed November 8, 2009).

[45] N. Ragouzis, J. Hughes, R. Philpott, E. Maler, P. Madsen and T. Scavo, “Security Assertion Markup Language (SAML) v2.0 technical overview,” March 2008. Available: http://www.oasis-open.org/specs/ (accessed November 5, 2009).

[46] S. Cantor, J. Kemp, R. Philpott and E. Maler, “Assertions and protocols for the OASIS Security Assertion Markup Language (SAML) v2.0,” March 2005. Available: http://www.oasis-open.org/specs/ (accessed November 5, 2009).

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Peter Denning
   Chairman, Department of Computer Science
   Naval Postgraduate School
   Monterey, California

4. Dr. Bert Lundy
   Naval Postgraduate School
   Monterey, California

5. J. D. Fulp
   Naval Postgraduate School
   Monterey, California

6. Mok Chuan-Hao
   Singapore Armed Forces
   Republic of Singapore

7. Dr. Yeo Tat Soon
   Director, Temasek Defence Systems Institute
   National University of Singapore
   Republic of Singapore

8. Ms Tan Lai Poh
   Assistant Manager, Temasek Defence Systems Institute
   National University of Singapore
   Republic of Singapore