NIST Special Publication 800-63-1

# Electronic Authentication Guideline

*Recommendations of the National Institute of Standards and Technology*

**William E. Burr**
**Donna F. Dodson**
**Ray A. Perlner**
**W. Timothy Polk**
**Sarbari Gupta**
**Emad A. Nabbus**

National Institute of Standards and Technology
U.S. Department of Commerce

## I N F O R M A T I O N  S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

***December 8, 2008***

**U.S. Department of Commerce**
*Carlos M. Gutierrez, Secretary*

**National Institute of Standards and Technology**
*James M. Turner, Acting Director*

## Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

## Authority

This document has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), Securing Agency Information Systems, as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may also be used by non-governmental organizations on a voluntary basis and is not subject to copyright. (Attribution would be appreciated by NIST.)

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

**National Institute of Standards and Technology Special Publication 800-63-1, 97 pages (December 2008)**

## Abstract

This recommendation provides technical guidelines for Federal agencies implementing electronic authentication. The recommendation covers remote authentication of users over open networks. It defines technical requirements for each of four levels of assurance in the areas of identity proofing, registration, tokens, management processes, authentication protocols and related assertions.

KEY WORDS: Authentication, Authentication Assurance, Credentials Service Provider, Cryptography, Electronic Authentication, Electronic Credentials, Electronic Transactions, Electronic Government, Identity Proofing, Passwords, PKI, Public Key Infrastructure, Tokens.

## Acknowledgments

The authors, William Burr, Donna Dodson, Ray Perlner, and Tim Polk of the National Institute of Standards and Technology (NIST), and Sarbari Gupta, and Emad Nabbus of Electrosoft, would like to acknowledge the contributions of our many reviewers, including those from Enspier, Orion Security, and Mitre.

## Executive Summary

Electronic authentication (E-authentication) is the process of establishing confidence in user identities electronically presented to an information system. E-authentication presents a technical challenge when this process involves the remote authentication of individual people over a network, for the purpose of electronic government and commerce. This recommendation provides technical guidelines to agencies to allow an individual to remotely authenticate his or her identity to a Federal IT system. These guidelines address only traditional, widely implemented methods for remote authentication based on secrets. With these methods, the individual to be authenticated proves that he or she knows or possesses some secret information.

These technical guidelines supplement OMB guidance, *E-Authentication Guidance for Federal Agencies*, [OMB M-04-04] that defines four levels of authentication, Levels 1 to 4. These levels are defined in terms of the consequences of the authentication errors and misuse of credentials. Level 1 is the lowest assurance and Level 4 is the highest. The OMB guidance defines the required level of authentication assurance in terms of the likely consequences of an authentication error. As the consequences of an authentication error become more serious, the required level of assurance increases. The OMB guidance provides agencies with the criteria for determining the level of E-authentication assurance required for specific applications and transactions, based on the risks and their likelihood of occurrence of each application or transaction.

OMB guidance outlines a 5 step process by which agencies should meet their E-authentication assurance requirements:

1. Conduct a risk assessment of the government system.
2. Map identified risks to the appropriate assurance level.
3. Select technology based on E-authentication technical guidance.
4. Validate that the implemented system has met the required assurance level.
5. Periodically reassess the information system to determine technology refresh requirements.

This document provides guidelines for implementing the third step of the above process. After completing a risk assessment and mapping the identified risks to the required assurance level, agencies can select appropriate technology that, at a minimum, meets the technical requirements for the required level of assurance. In particular, the document states specific technical requirements for each of the four levels of assurance in the following areas:

- Identity proofing and registration of Applicants,
- Tokens (typically a cryptographic key or password) for proving identity,
- Token and credential management mechanisms used to establish and maintain token and credential information,

- Protocols used to support the authentication mechanism between the Claimant and the Verifier,
- Assertion mechanisms used to communicate the results of a remote authentication if these results are sent to other parties.

A summary of the technical requirements for each of the four levels is provided below.

**Level 1** - Although there is no identity proofing requirement at this level, the authentication mechanism provides some assurance that the same Claimant is accessing the protected transaction or data. It allows a wide range of available authentication technologies to be employed and allows any of the token methods of Levels 2, 3, or 4. Successful authentication requires that the Claimant prove through a secure authentication protocol that he or she controls the token.

Plaintext passwords or secrets are not transmitted across a network at Level 1. However this level does not require cryptographic methods that block offline attacks by an eavesdropper. For example, simple password challenge-response protocols are allowed. In many cases an eavesdropper, having intercepted such a protocol exchange, will be able to find the password with a straightforward dictionary attack.

At Level 1, long-term shared authentication secrets may be revealed to Verifiers. All assertions recognized within this guideline must indicate the assurance level of the initial authentication of the Subscriber. At Level 1, assertions and assertion references must be protected from manufacture/modification and reuse attacks.

**Level 2** – Level 2 provides single factor remote network authentication. At Level 2, identity proofing requirements are introduced, requiring presentation of identifying materials or information. A wide range of available authentication technologies can be employed at Level 2. For single factor authentication, Memorized Secret Tokens, Pre-Registered Knowledge Tokens, Look-up Secret Tokens, Out of Band Tokens, and Single Factor One Time Password Devices are allowed at Level 2. Level 2 also allows any of the token methods of Levels 3 or 4. Successful authentication requires that the Claimant prove through a secure authentication protocol that he or she controls the token. Online guessing, replay, session hijacking and eavesdropping, attacks are prevented. Protocols shall also be at least weakly resistant to man-in-the middle attacks as defined in Section 9.2.2.

Long-term shared authentication secrets, if used, are never revealed to any party except the Claimant and Verifiers operated by the Credentials Service Provider (CSP); however, session (temporary) shared secrets may be provided to independent Verifiers by the CSP. In addition to Level 1 requirements, assertions must be resistant to disclosure, redirection, capture and substitution attacks. Approved cryptographic techniques are required for all assertion protocols used at Level 2 and above.

**Level 3**- Level 3 provides multi-factor remote network authentication. At least two authentication factors are required. At this level, identity proofing procedures require verification of identifying materials and information. Level 3 authentication is based on

proof of possession of the allowed types of tokens through a cryptographic protocol. Multi-factor Software Cryptographic Tokens are allowed at Level 3. Level 3 also allows any of the token methods of Level 4. Level 3 authentication requires cryptographic strength mechanisms that protect the primary authentication token against compromise by the protocol threats for all threats at Level 2 as well as verifier impersonation attacks. Various types of tokens may be used as described in Section 7.

Authentication requires that the Claimant prove through a secure authentication protocol that he or she controls the token. The Claimant must first unlock the token with a password or biometric, or must use a secure multi-token authentication protocol to establish two-factor authentication (through proof of possession of a physical or software token in combination with some memorized secret knowledge). Long-term shared authentication secrets, if used, are never revealed to any party except the Claimant and Verifiers operated directly by the Credentials Service Provider (CSP); however, session (temporary) shared secrets may be provided to independent Verifiers by the CSP. In addition to Level 2 requirements, assertions shall be protected against repudiation by the Verifier.

**Level 4** – Level 4 is intended to provide the highest practical remote network authentication assurance. Level 4 authentication is based on proof of possession of a key through a cryptographic protocol. At this level, it is required that identity proofing be done in person. Level 4 is similar to Level 3 except that only "hard" cryptographic tokens are allowed, FIPS 140-2 cryptographic module validation requirements are strengthened, and subsequent critical data transfers must be authenticated via a key bound to the authentication process. The token shall be a hardware cryptographic module validated at FIPS 140-2 Level 2 or higher overall with at least FIPS 140-2 Level 3 physical security. By requiring a physical token, which cannot readily be copied, and since FIPS 140-2 requires operator authentication at Level 2 and higher, this level ensures good, two factor remote authentication. The PIV Card authentication key meets Level 4 token requirements.

Level 4 requires strong cryptographic authentication of all parties and all sensitive data transfers between the parties. Either public key or symmetric key technology may be used. Authentication requires that the Claimant prove through a secure authentication protocol that he or she controls the token. All protocol threats at Level 3 shall be prevented at Level 4. Protocols shall also be strongly resistant to man-in-the-middle attacks. Long-term shared authentication secrets, if used, are never revealed to any party except the Claimant and Verifiers operated directly by the Credentials Service Provider (CSP); however, session (temporary) shared secrets may be provided to independent Verifiers by the CSP. Strong Approved cryptographic techniques are used for all operations. All sensitive data transfers are cryptographically authenticated using keys bound to the authentication process.

At Level 4, bearer assertions shall not be used to establish the identity of the Claimant to the Relying Party. Holder-of-key assertions may be used, provided that the assertion contains a reference to a key that is possessed by the Subscriber and is cryptographically

linked to the Level 4 token used to authenticate to the Verifier. The Relying Party shall maintain records of the assertions it receives, to support non-repudiation.

## Table of Contents

1. Purpose ..... 1
2. Authority ..... 1
3. Introduction ..... 1
4. Definitions and Abbreviations ..... 6
5. E-Authentication Model ..... 13
5.1. Overview ..... 13
5.2. Subscribers, RAs and CSPs ..... 15
5.3. Tokens ..... 16
5.4. Electronic Credentials ..... 17
5.5. Verifiers ..... 18
5.6. Assertions ..... 18
5.7. Relying Parties ..... 19
5.8. Overall Authentication Solution Assurance Level ..... 19
5.9. Token and Credential Lifecycle ..... 20
6. Registration and Issuance ..... 22
6.1. Overview ..... 22
6.2. Registration and Issuance Threats ..... 23
6.2.1 Threat Mitigation Strategies ..... 24
6.3. Registration, and Issuance Assurance Levels ..... 25
6.3.1 Requirements per Assurance Level ..... 25
6.3.2 Mapping FPKI Certificate Policies to Registration Levels ..... 30
7 Tokens ..... 31
7.1 Overview ..... 31
7.1.1 Single-Factor versus Multi-factor Tokens ..... 32
7.1.2 Token Types ..... 32
7.1.3 Token Usage ..... 34
7.1.4 Multi-Stage Authentication Using Tokens ..... 35
7.2 Token Threats ..... 35
7.2.1 Threat Mitigation Strategies ..... 38
7.3 Token Assurance Levels ..... 39
7.3.1 Requirements per Assurance Level ..... 39
8 Token and Credential Management ..... 45
8.1 Overview ..... 45
8.1.1 Token and Credential Management Activities ..... 46
8.2 Token and Credential Management Threats ..... 48
8.2.1 Threat Mitigation Strategies ..... 50
8.3 Token and Credential Management Assurance Levels ..... 51
8.3.1 Requirements per Assurance Level ..... 51
8.3.2 Relationship of PKI Policies to E-Authentication Assurance Levels ..... 55
9 Authentication Process ..... 56
9.1 Overview ..... 56
9.2 Authentication Process Threats ..... 57
9.2.1 Other Threats ..... 58
9.2.2 Threat Mitigation Strategies ..... 59

9.2.3 Phishing and Pharming (Verifier Impersonation) ..... 62
9.3 Authentication Process Assurance Levels ..... 64
9.3.1 Threat Resistance per Assurance Level ..... 64
9.3.2 Additional Requirements per Assurance Level ..... 65
10 Assertions ..... 68
10.1 Overview ..... 68
10.1.1 Cookies ..... 71
10.1.2 Security Assertions Markup Language (SAML) ..... 72
10.1.3 Kerberos Tickets ..... 72
10.2 Assertion Threats ..... 73
10.2.1 Threat Mitigation Strategies ..... 75
10.3 Assertion Assurance Levels ..... 77
10.3.1 Threat Resistance per Assurance Level ..... 77
10.3.2 Requirements per Assurance Level ..... 78
10.3.2.1 Level 1 ..... 78
10.3.2.2 Level 2 ..... 78
10.3.2.3 Level 3 ..... 79
10.3.2.4 Level 4 ..... 80
11 References ..... 82
11.1 General References ..... 82
11.2 NIST ITL Bulletins ..... 83
11.3 NIST Special Publications ..... 83
11.4 Federal Information Processing Standards ..... 84
11.5 Certificate Policies ..... 84
Appendix A: Estimating Entropy and Strength ..... 86
Password Entropy ..... 86
A.1 Randomly Selected Passwords ..... 87
A.2 User Selected Passwords ..... 88
A.2 Other Types of Passwords ..... 91
A.3 Examples ..... 91
Appendix B: Mapping of Federal PKI Certificate Policies to E-authentication Assurance Levels ..... 96

# 1. Purpose

This recommendation provides technical guidelines to agencies for the implementation of electronic authentication (E-authentication).

# 2. Authority

This document has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), Securing Agency Information Systems, as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided A-130, Appendix III.

This guideline has been prepared for use by Federal agencies. It may be used by non-governmental organizations on a voluntary basis and is not subject to copyright. (Attribution would be appreciated by NIST.)

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official.

# 3. Introduction

Electronic authentication (E-authentication) is the process of establishing confidence in user identities electronically presented to an information system. E-authentication presents a technical challenge when this process involves the remote authentication of individual people over a network. This recommendation provides technical guidelines to agencies to allow an individual person to remotely authenticate his/her identity to a Federal IT system. This recommendation also provides guidelines for Verifiers, Relying Parties and Credential Service Providers.

These technical guidelines supplement OMB guidance, *E-Authentication Guidance for Federal Agencies*, [OMB M-04-04] that defines four levels of assurance Levels 1 to 4, in terms of the consequences of the authentication errors and misuse of credentials. Level 1 is the lowest assurance level and Level 4 is the highest. The guidance defines the required level of authentication assurance in terms of the likely consequences of an authentication error. As the consequences of an authentication error become more serious, the required

level of assurance increases. The OMB guidance provides agencies with criteria for determining the level of E-authentication assurance required for specific electronic transactions and systems, based on the risks and their likelihood of occurrence.

OMB guidance outlines a 5 step process by which agencies should meet their E-authentication assurance requirements:

1. *Conduct a risk assessment of the government system* – No specific risk assessment methodology is prescribed for this purpose, however the e-RA tool <http://www.cio.gov/eauthentication/era.cfm> is an example of a suitable tool and methodology, while NIST SP 800-30 offers a general process for Risk Assessment and Risk Mitigation.
2. *Map identified risks to the appropriate assurance level* – Section 2.2 of OMB M-04-04 provides the guidance necessary for agencies to perform this mapping.
3. *Select technology based on E-authentication technical guidance* – After the appropriate assurance level has been determined, OMB guidance states that agencies should select technologies that meet the corresponding technical requirements, as specified by this document. Some agencies may have existing technology in for E-authentication. Agencies should verify that the existing technology meets the requirements as specified in this document.
4. *Validate that the implemented system has met the required assurance level* – As some implementations may create or compound particular risks, agencies should conduct a final validation to confirm that the system achieves the required assurance level for the user-to-agency process. NIST special publication 800-53A provides guidelines for the assessment of the implemented system during the validation process. Validation should be performed as part of a certification and accreditation process as described in NIST SP 800-37.
5. *Periodically reassess the information system to determine technology refresh requirements* – The agency must periodically reassess the information system to ensure that the identity authentication requirements continue to be valid as a result of technology changes or changes to the agency's business processes. As with the initial validation process, agencies should follow the technical assessment guidelines specified in SP 800-53A. Annual information security assessment requirements provide an excellent opportunity for this.

Agencies may adjust the identity credential's level of assurance using additional risk mitigation measures. Easing identity credential assurance level requirements may increase the size of the enabled customer pool, but agencies must ensure that this does not corrupt the system's choice of the appropriate assurance level. Alternatively, agencies may consider partitioning the functionality of an E-authentication enabled application to allow less sensitive functions to be available at a lower level of authentication assurance, while more sensitive functions are available only at a higher level of assurance.

This document provides guidelines for implementing the third step of the above process. In particular, this document states specific technical requirements for each of the four levels of assurance in the following areas:

- Registration and identity proofing of Applicants;
- Tokens (typically a cryptographic key or password) for proving identity;
- Token and credential management mechanisms used to establish and maintain token and credential information;
- Protocols used to support the authentication mechanism between the Claimant and the Verifier;
- Assertion mechanisms used to communicate the results of a remote authentication if these results are sent to other parties.

The overall authentication assurance level is determined by the lowest assurance level achieved in any of the areas listed above.

These technical guidelines cover remote electronic authentication of human users to IT systems over a network. They do not address the authentication of a person who is physically present, for example, for access to buildings, although some credentials and tokens that are used remotely may also be used for local authentication. While these technical guidelines do, in many cases, establish requirements that Federal IT systems and service providers participating in authentication protocols be authenticated to Subscribers, they do not specifically address machine-to-machine (such as router-to-router) authentication, nor do these guidelines establish specific requirements for issuing authentication credentials and tokens to machines and servers when they are used in E-authentication protocols with people.

The paradigm of this document is that individuals are enrolled and undergo an identity proofing process in which their identity is bound to an authentication secret, called a token. Thereafter, the individuals are remotely authenticated to systems and applications over an open network, using the token in an authentication protocol. The authentication protocol allows an individual to demonstrate to a Verifier that he or she has or knows the secret token, in a manner that protects the secret from compromise by different kinds of attacks. Higher authentication assurance levels require use of stronger tokens (harder to guess secrets) and better protection of the token from attacks. This document covers authentication mechanisms that work by making the individual demonstrate possession and control of a secret. In order to improve the security of the authentication exchange, the Verifiers authenticate to the Claimant so that the latter has confidence that he or she is talking to the intended Verifier.

It may also be practical to achieve authentication by testing the personal knowledge of the individual (referred to as knowledge based authentication). As this information is private but not actually secret, confidence in the identity of an individual can be hard to achieve. In addition, the complexity and interdependencies of knowledge based authentication systems are difficult to quantify. However, knowledge based authentication techniques are included as part of registration in this document.

Biometric methods are widely used to authenticate individuals who are physically present at the authentication point, for example for entry into buildings. Biometrics do not constitute secrets suitable for use in the conventional remote authentication protocols addressed in this document. In the local authentication case, where the Claimant is observed and uses a capture device controlled by the Verifier, authentication does not require that biometrics be kept secret. The use of biometrics to "unlock" conventional authentication tokens, to prevent repudiation of registration, and to fully or partially verify that the same individual participates in all phases of the registration process is supported in this document.

This document identifies minimum technical requirements for remotely authenticating identity. Agencies may determine based on their risk analysis that additional measures are appropriate in certain contexts. In particular, privacy requirements and legal risks may lead agencies to determine that additional authentication measures or other process safeguards are appropriate. When developing E-authentication processes and systems, agencies should consult *OMB Guidance for Implementing the Privacy Provisions of the E-Government Act of 2002* [OMB M-03-22]. See the *Guide to Federal Agencies on Implementing Electronic Processes* for additional information on legal risks, especially those that are related to the need to satisfy legal standards of proof and prevent repudiation [DOJ 2000].

Additionally, Federal agencies implementing these guidelines should adhere to the requirements of Title III of the E-Government Act, entitled the Federal Information Security Management Act (FISMA), and the related NIST standards and guidelines. FISMA directs Federal agencies to develop, document, and implement agency-wide programs to provide information security for the information and information systems that support the operations and assets of the agency. This includes the certification and accreditation of IT systems that support E-authentication. It is recommended that non-Federal entities implementing these guidelines also follow equivalent standards of security management, certification and accreditation to ensure the secure operations of their E-authentication systems.

This document has been updated to reflect current (token) technologies and has been restructured to provide a better understanding of the E-authentication architectural model. Additional (minimum) technical requirements have been specified for the Credential Service Provider, protocols utilized to transport authentication information, and assertions if implemented within the E-authentication model.

The subsequent sections present a series of recommendations for the secure implementation of Registration Authorities, Credential Service Providers (CSPs), Verifiers, and Relying Parties. It should be noted that secure implementation of any one of these can only provide the desired level of assurance if the others are also implemented securely. Therefore, the following assumptions have been made in this guideline:

- RAs, CSPs, Verifiers, and Relying Parties are trusted entities. It is assumed that agencies implementing any of the above trusted entities have some

assurance that all other trusted entities with which it interacts are also implemented appropriately for the desired security level.

- It is assumed that there exists a process of certification through which agencies can obtain the above assurance for trusted entities which they do not implement themselves.
- A trusted entity is considered to be implemented appropriately if it complies with the recommendations laid out in this document and does not behave maliciously.
- While it is generally assumed that trusted entities will not behave maliciously, this document does contain some recommendations to reduce and isolate any damage done by a malicious or negligent trusted entity.

# 4. Definitions and Abbreviations

| | |
|---|---|
| Active Attack | An attack on the authentication protocol where the Attacker transmits data to the Claimant or Verifier. Examples of active attacks include man-in-the-middle, impersonation, and session hijacking. |
| Address of Record | The official location where an individual can be found. The address of record always includes the residential street address of an individual and may also include the mailing address of the individual. In very limited circumstances, an Army Post Office box number, Fleet Post Office box number or the street address of next of kin or of another contact individual can be used when a residential street address for the individual is not available. |
| Approved | FIPS approved or NIST recommended. An algorithm or technique that is either 1) specified in a FIPS or NIST Recommendation, or 2) adopted in a FIPS or NIST Recommendation. |
| Applicant | A party undergoing the processes of registration and identity proofing. |
| Assertion | A statement from a Verifier to a Relying Party that contains identity information about a Subscriber. Assertions may also contain verified attributes. |
| Assurance | In the context of OMB M-04-04 and this document, assurance is defined as 1) the degree of confidence in the vetting process used to establish the identity of an individual to whom the credential was issued, and 2) the degree of confidence that the individual who uses the credential is the individual to whom the credential was issued. |
| Asymmetric Keys | Two related keys, a public key and a private key that are used to perform complementary operations, such as encryption and decryption or signature generation and signature verification. |
| Attack | An attempt to obtain a Subscriber's token or to fool a Verifier into believing that an unauthorized individual possess a Claimant's token. |
| Attacker | A party who acts with malicious intent to assault an information system. |
| Authentication | The process of establishing confidence in the identity of users or information systems. |
| Authentication Protocol | A defined sequence of messages between a Claimant and a Verifier that demonstrates that the Claimant has control of a valid token to establish his/her identity, and optionally, demonstrates to the Claimant that he or she is communicating with the intended Verifier. |
| Authenticity | The property that data originated from its purported source. |
| Bearer Assertion | An assertion that does not provide a mechanism for the Subscriber to prove that he or she is the rightful owner of the assertion. The Relying Party has to assume that the assertion was issued to the Subscriber who presents the assertion or the corresponding assertion reference to the Relying Party. |
| Bit | A binary digit: 0 or 1. |

| | |
|---|---|
| Biometrics | Automated recognition of individuals based on their behavioral and biological characteristics. In this document, biometrics may be used to unlock authentication tokens and prevent repudiation of registration. |
| Certification Authority (CA) | A trusted entity that issues and revokes public key certificates. |
| Certificate Revocation List (CRL) | A list of revoked public key certificates created and digitally signed by a Certification Authority. See [RFC 3280] |
| Challenge-Response Protocol | An authentication protocol where the Verifier sends the Claimant a challenge (usually a random value or a nonce) that the Claimant combines with a secret (such as by hashing the challenge and a shared secret together, or by applying a private key operation to the challenge) to generate a response that is sent to the Verifier. The Verifier can independently verify the response generated by the Claimant (such as by re-computing the hash of the challenge and the shared secret and comparing to the response, or performing a public key operation on the response) and establish that the Claimant possesses and controls the secret. |
| Claimant | A party whose identity is to be verified using an authentication protocol. |
| Credential | An object that authoritatively binds an identity (and optionally, additional attributes) to a token possessed and controlled by a person. |
| Credentials Service Provider (CSP) | A trusted entity that issues or registers Subscriber tokens and issues electronic credentials to Subscribers. The CSP may encompass Registration Authorities and Verifiers that it operates. A CSP may be an independent third party, or may issue credentials for its own use. |
| Cross Site Request Forgery (CSRF) | An attack in which a Subscriber who is currently authenticated to a Relying Party and connected through a secure session, browses to an attacker's website which causes the Subscriber to unknowingly invoke unwanted actions at the Relying Party. |
| Cryptographic Key | A value used to control cryptographic operations, such as decryption, encryption, signature generation or signature verification. For the purposes of this document, key requirements shall meet the minimum requirements stated in table 2 of NIST SP 800-57 part 1.<br>See also Asymmetric keys, Symmetric key. |
| Cryptographic Token | A token where the secret is a cryptographic key. |
| Data Integrity | The property that data has not been altered by an unauthorized entity. |
| Digital Signature | An asymmetric key operation where the private key is used to digitally sign an electronic document and the public key is used to verify the signature. Digital signatures provide authentication and integrity protection. |
| Eavesdropping Attack | An attack in which an Attacker listens passively to the authentication protocol to capture information which can be used in a subsequent active attack to masquerade as the Claimant. |
| Electronic Authentication (E-Authentication) | The process of establishing confidence in user identities electronically presented to an information system. |

| | |
|---|---|
| Electronic Credentials | Digital documents used in authentication that bind an identity or an attribute to a Subscriber's token. Note that this document distinguishes between credentials, and tokens (see below) while other documents may interchange these terms. |
| Entropy | A measure of the amount of uncertainty that an Attacker faces to determine the value of a secret. Entropy is usually stated in bits. See Appendix A. |
| Extensible Mark-up Language (XML) | Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them. |
| FIPS | Federal Information Processing Standard. |
| Guessing Entropy | A measure of the difficulty that an Attacker has to guess the average password used in a system. In this document, entropy is stated in bits. When a password has n-bits of guessing entropy then an Attacker has as much difficulty guessing the average password as in guessing an n-bit random quantity. The Attacker is assumed to know the actual password frequency distribution. See Appendix A. |
| Hash Function | A function that maps a bit string of arbitrary length to a fixed length bit string. Approved hash functions satisfy the following properties:<br>1. (One-way) It is computationally infeasible to find any input that maps to any pre-specified output, and<br>2. (Collision resistant) It is computationally infeasible to find any two distinct inputs that map to the same output. |
| Holder-of-Key Assertion | An assertion that contains a reference to a symmetric key or a public key (corresponding to a private key) possessed by the Subscriber. The Relying Party may require the Subscriber to prove |
| Identity | A unique name of an individual person. Since the legal names of persons are not necessarily unique, the identity of a person must include sufficient additional information (for example an address, or some unique identifier such as an employee or account number) to make the complete name unique. |
| Identity Proofing | The process by which a CSP and an RA validate sufficient information to uniquely identify a person. |
| Kerberos | A widely used authentication protocol developed at MIT. In "classic" Kerberos, users share a secret password with a Key Distribution Center (KDC). The user, Alice, who wishes to communicate with another user, Bob, authenticates to the KDC and is furnished a "ticket" by the KDC to use to authenticate with Bob. When Kerberos authentication is based on passwords, the protocol is known to be vulnerable to off-line dictionary attacks by eavesdroppers who capture the initial user-to-KDC exchange. |
| Man-in-the-Middle Attack (MitM) | An attack on the authentication protocol run in which the Attacker positions himself in between the Claimant and Verifier so that he can intercept and alter data traveling between them. |

| | |
|---|---|
| Message Authentication Code (MAC) | A cryptographic checksum on data that uses a symmetric key to detect both accidental and intentional modifications of the data. |
| Min-entropy | A measure of the difficulty that an Attacker has to guess the most commonly chosen password used in a system. In this document, entropy is stated in bits. When a password has n-bits of min-entropy then an Attacker requires as many trials to find a user with that password as is needed to guess an n-bit random quantity. The Attacker is assumed to know the most commonly used password(s). See Appendix A. |
| Network | An open communications medium, typically the Internet, that is used to transport messages between the Claimant and other parties. Unless otherwise stated no assumptions are made about the security of the network; it is assumed to be open and subject to active (e.g., impersonation, man-in-the-middle, session hijacking…) and passive (e.g., eavesdropping) attack at any point between the parties (Claimant, Verifier, CSP or Relying Party). |
| Nonce | A value used in security protocols that is never repeated with the same key. For example, challenges used in challenge-response authentication protocols generally must not be repeated until authentication keys are changed, or there is a possibility of a replay attack. Using a nonce as a challenge is a different requirement than a random challenge, because a nonce is not necessarily unpredictable. |
| Off-line Attack | An attack where the Attacker obtains some data (typically by eavesdropping on an authentication protocol run, or by penetrating a system and stealing security files) that he/she is able to analyze in a system of his/her own choosing. |
| On-line Attack | An attack against an authentication protocol where the Attacker either assumes the role of a Claimant with a genuine Verifier or actively alters the authentication channel. The goal of the attack may be to gain authenticated access or learn authentication secrets. |
| Online Guessing Attack | An attack in which an Attacker performs repeated logon trials by guessing possible values of the token authenticator. |
| Passive Attack | An attack against an authentication protocol where the Attacker intercepts data traveling along the network between the Claimant and Verifier, but does not alter the data (i.e. eavesdropping). |
| Password | A secret that a Claimant memorizes and uses to authenticate his or her identity. Passwords are typically character strings. |
| Personal Identification Number (PIN) | A password consisting only of decimal digits. |
| Personal Identity Verification (PIV) Card | A physical artifact (e.g., identity card, “smart” card) issued to an individual that contains stored identity credentials (e.g., photograph, cryptographic keys, digitized fingerprint representation) so that the claimed identity of the cardholder can be verified against the stored credentials by another person (human readable and verifiable) or an automated process (computer readable and verifiable). |

| | |
|---|---|
| Pharming | An attack in which an Attacker corrupts an infrastructure service such as DNS (Domain Name Service) causing the Subscriber to be misdirected to a forged Verifier/Relying Party, and revealing sensitive information, downloading harmful software or contributing to a fraudulent act. |
| Phishing | An attack in which the Subscriber is lured (usually through an email) to interact with a counterfeit Verifier, and tricked into revealing information that can be used to masquerade as that Subscriber to the real Verifier. |
| Possession and control of a token | The ability to activate and use the token in an authentication protocol. |
| Practice Statement | A formal statement of the practices followed by an authentication entity (e.g., RA, CSP, or Verifier); typically the specific steps taken to register and verify identities, issue credentials and authenticate Claimants. |
| Private Key | The secret part of an asymmetric key pair that is typically used to digitally sign or decrypt data. |
| Proof of Possession (PoP) Protocol | A protocol where a Claimant proves to a Verifier that he/she possesses and controls a token (e.g., a key or password) |
| Protected Channel | A session wherein messages between two participants are encrypted and integrity is protected using a set of shared secrets; A participant is said to be authenticated if the other participant can link possession of the session keys by the first participant to a long term cryptographic token and verify the identity associated with that token. |
| Public Key | The public part of an asymmetric key pair that is typically used to verify signatures or encrypt data. |
| Public Key Certificate | A digital document issued and digitally signed by the private key of a Certification Authority that binds the name of a Subscriber to a public key. The certificate indicates that the Subscriber identified in the certificate has sole control and access to the private key. See also [RFC 3280]. |
| Pseudonym | A Subscriber name that has been chosen by the Subscriber that is not verified as meaningful by identity proofing. |
| Registration | The process through which a party applies to become a Subscriber of a CSP and an RA validates the identity of that party on behalf of the CSP. |
| Registration Authority (RA) | A trusted entity that establishes and vouches for the identity of a Subscriber to a CSP. The RA may be an integral part of a CSP, or it may be independent of a CSP, but it has a relationship to the CSP(s). |
| Relying Party | An entity that relies upon the Subscriber's credentials or Verifier's assertion of an identity, typically to process a transaction or grant access to information or a system. |
| Replay Attack | An attack in which the Attacker is able to replay previously captured messages (between a legitimate Claimant and a Verifier) to masquerade as that Claimant to the Verifier or vice versa. |

| | |
|---|---|
| Salt | A non-secret value that is used in a cryptographic process, usually to ensure that the results of computations for one instance cannot be reused by an Attacker. |
| Secure Sockets Layer (SSL) | An authentication and security protocol widely implemented in browsers and web servers. SSL has been superseded by the newer Transport Layer Security (TLS) protocol; TLS 1.0 is effectively SSL version 3.1. |
| Security Assertion Mark-up Language (SAML) | An XML-based security specification developed by OASIS for exchanging authentication (and authorization) information between trusted entities over the Internet. |
| Session Hijack Attack | An attack in which the Attacker is able to insert himself or herself between a Claimant and a Verifier subsequent to a successful authentication exchange between the latter two parties. The Attacker is able to pose as a Subscriber to the Verifier or vice versa to control session data exchange. |
| SAML Authentication Assertion | A SAML assertion that conveys information about a successful act of authentication that took place for a subject. |
| Shared Secret | A secret used in authentication that is known to the Claimant and the Verifier. |
| Social Engineering | The act of deceiving an individual into revealing sensitive information by associating with the individual to gain confidence and trust. |
| Subject | The person whose identity is bound in a particular credential. |
| Subscriber | A party who has received a credential or token from a CSP. |
| Symmetric Key | A cryptographic key that is used to perform both the cryptographic operation and its inverse, for example to encrypt and decrypt, or create a message authentication code and to verify the code. |
| Token | Something that the Claimant possesses and controls (typically a key or password) used to authenticate the Claimant's identity. |
| Token Authenticator | The value that is provided to the protocol stack to prove that the Claimant possesses and controls the token. Protocol messages sent to the Verifier are dependant upon the token authenticator, but they may or may not explicitly contain it. |
| Transport Layer Security (TLS) | An authentication and security protocol widely implemented in browsers and web servers. TLS is defined by [RFC 2246] and [RFC 3546]. TLS is similar to the older Secure Sockets Layer (SSL) protocol, and TLS 1.0 is effectively SSL version 3.1. NIST SP 800-52, *Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations* specifies how TLS is to be used in government applications. |
| Tunneled Password Protocol | A protocol where a password is sent through a protected channel to a cryptographically authenticated Verifier. For example, the TLS protocol is often used with a Verifier's public key certificate to (1) authenticate the Verifier to the Claimant, (2) establish an encrypted session between the Verifier and Claimant, and (3) transmit the Claimant's password to the Verifier. The encrypted TLS session protects the Claimant's password from eavesdroppers. |

| | |
|---|---|
| Verified Name | A Subscriber name that has been verified by identity proofing. |
| Verifier | An entity that verifies the Claimant's identity by verifying the Claimant's possession of a token using an authentication protocol. To do this, the Verifier may also need to validate credentials that link the token and identity and check their status. |
| Verifier Impersonation Attack | A scenario where the Attacker impersonates the Verifier in an authentication protocol, usually to capture information that can be used to masquerade as that Claimant to the real Verifier. |
| Zero-knowledge Password Protocol | A password based authentication protocol that allows a claimant to authenticate to a Verifier without revealing the password to the Verifier. Examples of such protocols are EKE, SPEKE and SRP. |

# 5. E-Authentication Model

## *5.1. Overview*

In accordance with [OMB M-04-04], E-authentication is the process of establishing confidence in user identities electronically presented to an information system. Systems can use the authenticated identity to determine if that individual is authorized to perform an electronic transaction. In most cases, the authentication and transaction take place across an open network such as the Internet; however, in some cases access to the network may be limited and access control decisions may take this into account.

E-authentication begins with *registration*. An *Applicant* applies to a *Registration Authority (RA)* to become a *Subscriber* of a *Credential Service Provider (CSP)* and, as a Subscriber, is issued or registers a secret, called a *token*, and a *credential* that binds the token to a name and possibly other attributes that the RA has verified. The token and credential may be used in subsequent authentication events.

The Subscriber's name may either be a *verified name* or a *pseudonym*. A verified name is associated with the identity of a real person and before an Applicant can receive credentials or register a token associated with a verified name, he or she must demonstrate that the identity is a real identity, and that he or she is the person who is entitled to use that identity. This process is called *identity proofing*, and is performed by an RA that registers Subscribers with the CSP. At Level 1, since names are not verified, names are always assumed to be pseudonyms. Level 2 credentials and assertions must specify whether the name is a verified name or a pseudonym. This information assists *Relying Parties*, that is, parties who rely on the name or other authenticated attributes, in making access control or authorization decisions. Only verified names are allowed at Levels 3 and 4.

In this document, the party to be authenticated is called a *Claimant* and the party verifying that identity is called a *Verifier*. When a *Claimant* successfully demonstrates possession and control of a token in an on-line authentication to a *Verifier* through an *authentication protocol,* the Verifier can verify that the Claimant is the Subscriber. The Verifier passes on an assertion about the identity of the Subscriber to the Relying Party. That assertion includes identity information about a Subscriber, such as the Subscriber name, an identifier assigned at registration, or other Subscriber attributes that were verified in the registration process (subject to the policies of the CSP and the needs of the application). Where the Verifier is also the Relying Party, the assertion may be implicit. In addition, the Subscriber's identifying information may be incorporated in credentials (public key certificates) made available by the Claimant. The Relying Party can use the authenticated information provided by the Verifier/CSP to make access control or authorization decisions.

Authentication simply establishes identity, or in some cases verified personal attributes (for example the Subscriber is a US Citizen, is a student at a particular university, or is assigned a particular number or code by an agency or organization), not what that identity

is authorized to do or what access privileges he or she has; this is a separate decision. Relying parties, typically government agencies, will use a Subscriber's authenticated identity and other factors to make access control or authorization decisions. In many cases, the authentication process and services will be shared by many applications and agencies, but the individual agency or application is the Relying Party that must make the decision to grant access or process a transaction based on the specific application requirements. These guidelines provide technical recommendations for the process of authentication, not authorization.

The various entities and interactions that comprise the E-authentication model are illustrated below in Figure 1. The dashed box on the left shows the registration, credential issuance, maintenance activities, and the interactions between the Subscriber/Claimant, the Registration Authority and the CSP. The interactions are as follows:

1. An individual Applicant applies to an RA through a registration process.
2. The RA identity proofs that Applicant.
3. On successful identity proofing, the RA sends the CSP a registration confirmation message.
4. A secret token and a corresponding credential are established between the CSP and the new Subscriber.
5. The CSP maintains the credential, its status, and the registration data collected. The Subscriber maintains his or her token.

The dashed box on the right side of Figure 1 shows the entities and the interactions related to using a token and credential to perform E-authentication. When the Subscriber needs to authenticate to perform a transaction, he or she becomes a Claimant to a Verifier. The interactions are as follows:

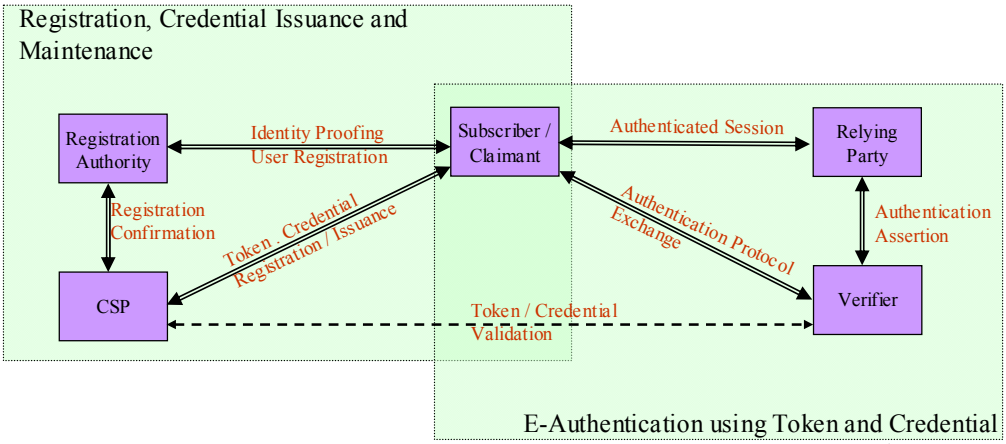
1. The Claimant proves to the Verifier that he or she possesses and controls the token through an authentication protocol.
2. The Verifier interacts with the CSP to validate the token and credential and confirm that the Claimant is a Subscriber of the CSP.
3. If the Verifier is separate from the Relying Party (application), the Verifier provides[1] an assertion about the Claimant to the Relying Party, which uses the information in the assertion to make an access control or authorization decision.
4. An authenticated session is established between the Claimant and the Relying Party.

In some cases the Verifier does not need to directly communicate with the CSP to complete the authentication activity (e.g., the use of digital certificates). Therefore, the dashed line between the Verifier and the CSP represents a logical link between the two

[1] Many assertion protocols require assertions to be forwarded through the Claimant's local system before reaching the Relying Party. For Details, see Section 10.

entities rather than a physical link. In some implementations, the Verifier, Relying Party and the CSP functions may be distributed and separated as shown in Figure 1; however, if these functions are co-resident on the same platform, the interactions between the components are local messages between applications running on the same system rather than protocols over shared untrusted networks.

Figure 1 - E-Authentication Architectural Model




### 5.2. Subscribers, RAs and CSPs

In the conceptual E-authentication model, a Claimant in an authentication protocol is a Subscriber to some CSP. At some point, an Applicant registers with an RA, which verifies the identity of the Applicant, typically through the presentation of paper credentials and by records in databases. This process is called identity proofing. The RA, in turn, vouches for the identity of the Applicant (and possibly other verified attributes) to a CSP. The Applicant then becomes a Subscriber of the CSP.

The CSP establishes a mechanism to uniquely identify each Subscriber and the associated tokens and credentials issued to that Subscriber. The CSP registers or gives the Subscriber a token to be used in an authentication protocol and issues credentials as needed to bind that token to the identity, or to bind the identity to some other useful verified attribute. The Subscriber may be given electronic credentials to go with the token at the time of registration, or credentials may be generated later as needed. Subscribers have a duty to maintain control of their tokens and comply with the responsibilities to the CSP. The CSP maintains registration records for each Subscriber to allow recovery of registration records.

There is always a relationship between the RA and CSP. In the simplest and perhaps the most common case, the RA and CSP are separate functions of the same entity. However, an RA might be part of a company or organization that registers Subscribers with an independent CSP, or several different CSPs. Therefore a CSP may have an integral RA,

or it may have relationships with multiple independent RAs, and an RA may have relationships with different CSPs as well.

Section 6 provides recommendations for the registration, identity proofing and issuance processes.

### *5.3. Tokens*

Tokens generally are something the Claimant possesses and controls that may be used to authenticate the Claimant's identity. In these guidelines, every token contains a secret. In E-authentication, the Claimant authenticates to a system or application over a network by proving that he or she has possession of a token. The token produces an output called an authenticator and this output is used in the authentication process to prove that the Claimant possesses and controls the token (refer to Section 7.1 for more details). Therefore, a token used for E-authentication must be protected. The token, for example, may be a piece of hardware (the token) which contains a cryptographic key (the token secret); the key is protected by encrypting it with a password. When used, the cryptographic key produces an output (the authenticator) which is used in the authentication process to authenticate the Claimant. An impostor must steal the encrypted key (by stealing the hardware) and learn the password to use the token.

Authentication systems are often categorized by the number of factors that they incorporate. The three factors often considered as the cornerstone of authentication are:

- *Something you know* (for example, a password)
- *Something you have* (for example, an ID badge or a cryptographic key)
- *Something you are* (for example, a thumb print or other biometric data)

Authentication systems that incorporate all three factors are stronger than systems that only incorporate one or two of the factors. The system may be implemented so that multiple factors are presented to the Verifier, or some factors may be used to protect a secret that will be presented to the Verifier. For example, consider a hardware device that holds a cryptographic key. The key might be activated by a password or the hardware device might include a biometric capture device and uses a biometric to activate the key. Such a device is considered to effectively provide two factor authentication, although the actual authentication protocol between the Verifier and the Claimant simply proves possession of the key.

The secrets are often based on either *public key pairs* (asymmetric keys) or *shared secrets*. A *public key* and a related private key comprise a public key pair. The *private key* is used by the Claimant as a token. A Verifier, knowing the Claimant's public key through some credential (typically a *public key certificate*), can use an authentication protocol to verify the Claimant's identity, by proving that the Claimant has control of the associated private key token (*Proof of Possession*).

Shared secrets are either *symmetric keys* or passwords. In a protocol sense, all shared secrets are similar, and can be used in similar authentication protocols; however, passwords, since they are often committed to memory, are something the Claimant knows, rather than something he or she has. Passwords, because they are committed to memory, usually do not have as many possible values as cryptographic keys, and, in many protocols, are vulnerable to network attacks that are impractical for keys. Moreover the entry of passwords into systems (usually through a keyboard) presents the opportunity for very simple keyboard logging or "shoulder surfing" attacks. Therefore keys and passwords demonstrate somewhat separate authentication properties (something you know rather than something you have). Passwords often have lesser resistance to network attacks. However, when using either public key pairs or shared secrets, the Subscriber has a duty to maintain exclusive control of his or her token, since possession and control of the token is used to authenticate the Claimant's identity.

Biometrics are unique personal attributes that can be used to verify the identity of a person who is physically present at the point of verification. They include facial features, fingerprints, DNA, iris and retina scans, voiceprints and many other characteristics. This publication recommends that biometrics be used in the registration process to later prevent a Subscriber who is in fact registered from repudiating the registration, to help identify those who commit registration fraud, and to unlock tokens. Biometric characteristics are not recommended for use directly as tokens in this document.

Section 7 provides guidelines on the various types of tokens that may be used for electronic authentication.

### *5.4. Electronic Credentials*

Paper credentials are documents that attest to the identity or other attributes of an individual or entity called the subject of the credentials. Some common paper credentials include passports, birth certificates, driver's licenses, and employee identity cards. The credentials themselves are authenticated in a variety of ways: traditionally perhaps by a signature or a seal, special papers and inks, high quality engraving, and today by more complex mechanisms, such as holograms, that make the credentials recognizable and difficult to copy or forge. In some cases, simple possession of the credentials is sufficient to establish that the physical holder of the credential is indeed the subject of the credentials. More commonly, the credentials contain biometric information such as the subject's description, a picture of the subject or the handwritten signature of the subject, which can be used to authenticate that the holder of the credentials is indeed the subject of the credentials. When these paper credentials are presented in-person, authentication biometrics contained in those credentials can be checked to confirm that the physical holder of the credential is the subject.

Electronic identity credentials bind a name and perhaps other attributes to a token. This recommendation does not prescribe particular kinds of electronic credentials. There are a variety of electronic credential types in use today, and new types of credentials are constantly being created. At a minimum, credentials include identifying information that permits recovery of the records of the registration associated with the credentials and a

name that is associated with the Subscriber. In every case, given the issuer and the identifying information in the credential, it must be possible to recover the registration records upon which the credentials are based. Electronic credentials may be general-purpose credentials or targeted to a particular Verifier. Some common types of credentials are:

- X.509 public key identity certificates that bind an identity to a public key;
- X.509 attribute certificates that bind an identity or a public key with some attribute;
- Kerberos tickets that are encrypted messages binding the holder with some attribute or privilege.

Electronic credentials may be stored as data in a directory or database. These credentials may be digitally signed objects (e.g., X.509 certificates), in which case their integrity may be verified. In this case, the directory or database may be an untrusted entity, since the data it supplies is self-authenticating. Alternatively, the directory or database server may be a trusted entity that authenticates itself to the Relying Party or Verifier. When the directory or database server is trusted, unsigned credentials may simply be stored as unsigned data.

Section 8 provides guidelines for token and credential management activities that are applicable to electronic authentication.

### 5.5. Verifiers

In any authenticated on-line transaction, the Verifier must verify that the Claimant has possession and control of the token that verifies his or her identity. A Claimant authenticates his or her identity to a Verifier by the use of a token and an authentication protocol. This is called *Proof of Possession (PoP)*. Many PoP protocols are designed so that a Verifier, with no knowledge of the token before the authentication protocol run, learns nothing about the token from the run. The Verifier and CSP may be the same entity, the Verifier and Relying Party may be the same entity or all three may be separate entities. It is undesirable for Verifiers to learn shared secrets unless they are a part of the same entity as the CSP that registered the tokens. Where the Verifier and the Relying Party are separate entities, the Verifier shall convey the result of the authentication protocol to the Relying Party. The object created by the Verifier to convey this result is called an assertion.

Section 9 provides guidelines for the various types of protocols used to authenticate parties within the E-authentication model.

### 5.6. Assertions

Assertions can be used to pass information about the Claimant or the E-authentication process from the Verifier to a Relying Party. Assertions contain, at a minimum, the name of the Claimant, as well as identifying information that permits recovery of registration

records. A Relying Party trusts an assertion based on the source, the time of creation, and attributes associated with the Claimant. Conversely, when sending assertions across an open network, the Verifier is responsible for ensuring that any sensitive Subscriber information contained in the assertion can only be extracted by a Relying Party that it trusts to maintain the information's confidentiality.

Examples of assertions include:

- *Cookies* – character strings placed in a web browser's memory, are available to websites within the same Internet domain as the server that placed them in the web browser. Cookies are used for many purposes and may be assertions or may contain pointers to assertions.[2]
- *SAML Assertions* – specified using a mark up language intended for describing security assertions, can be used by a Verifier to make a statement to a Relying Party about the identity of a Claimant. SAML assertions may optionally be digitally signed.
- *Kerberos Tickets* – which allow a ticket granting authority to issue session keys to two authenticated parties using symmetric key based encapsulation schemes.

Section 10 provides guidelines applicable to electronic authentication for assertions.

### 5.7. Relying Parties

A Relying Party relies on results of an on-line authentication to establish the identity or attribute of a Subscriber for the purpose of some transaction. Relying parties will use a Subscriber's authenticated identity and other factors to make access control or authorization decisions. The Verifier and the Relying Party may be the same entity, or they may be separate entities. If they are separate entities, the Relying Party normally receives an assertion from the Verifier. The Relying Party ensures that the assertion came from a Verifier trusted by the Relying Party. The Relying Party also processes any additional information in the assertion, such as personal attributes or expiration times.

Section 10 provides guidelines for the assertions that may be used by Relying Parties to establish confidence in the identities of Subscribers when the Relying Party and the Verifier are not co-located.

### 5.8. Overall Authentication Solution Assurance Level

The overall authentication solution assurance level is based on the low watermark of the assurance levels for each of the components of the solution. For instance, to achieve an overall assurance level of 3:

[2] There are specific requirements that agencies must follow when implementing cookies. See OMB Memorandum M-03-22, OMB Guidance for Implementing the Privacy Provisions of the E-Government Act of 2002, available at: http://www.whitehouse.gov/omb/memoranda/m03-22.html.

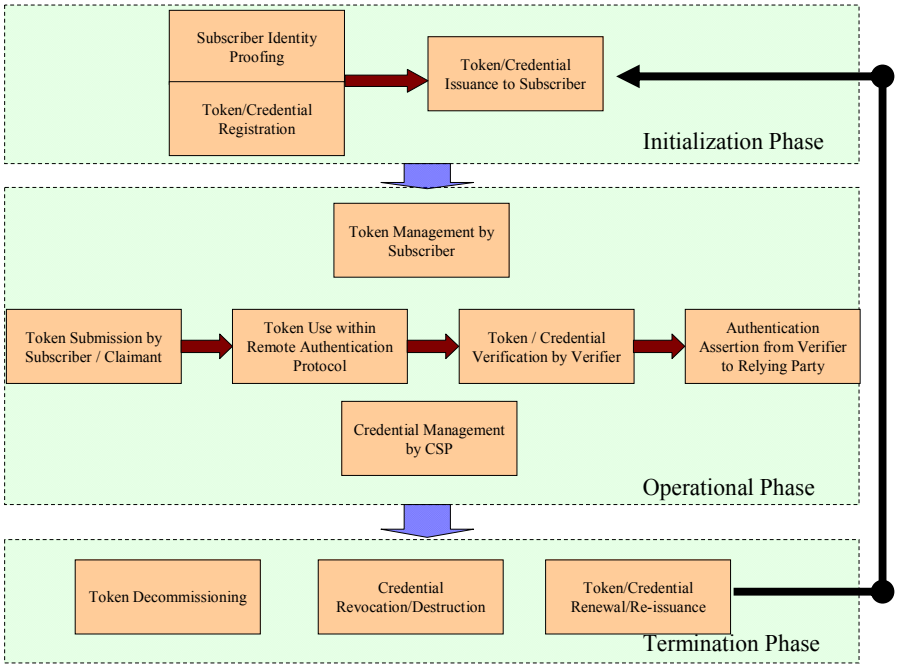
- The registration and identity proofing process shall, at a minimum, use Level 3 processes or higher.
- The token (or combination of tokens) used shall have an assurance level of 3 or higher.
- The authentication protocols used shall have a Level 3 assurance level or higher.
- The token and credential management processes shall use a Level 3 assurance level or higher.
- Assertions (if used) shall have a Level 3 assurance or higher.

If a system uses a token for authentication that has Level 2 assurance, but uses other mechanisms that have Level 3 assurance, the Attacker will likely focus on gaining access to the token since it is easier to attack an area meeting assurance Level 2 rather than attacking areas meeting assurance Level 3. (See Sections 6 through 10 for information on assurance levels for each area.)

### *5.9. Token and Credential Lifecycle*

Figure 2 illustrates three major phases within the lifecycle of a token: an initialization phase, an operational phase, and a termination phase. This model facilitates the identification of the threats and attacks that arise in each of the life cycle functions.

Figure 2 - Token and Credential Lifecycle




The initialization phase results in the successful registration of Subscriber and the establishment of the token and credential. This phase includes identity proofing, token

and credential registration, and token and credential issuance and delivery. In some cases, interim tokens (that provide a limited degree of authenticated access) may be issued to a Subscriber even before the identity proofing step is completed (e.g., in the case of the PIV Card). Based on the results of the completed identity proofing step, the Subscriber's interim token may either be revoked or upgraded to a long term token with full degree of authenticated access.. Once the initialization phase is complete, the token enters the operational phase.

During the operational phase, various functions are performed in parallel. The Subscriber manages his or her token and keeps it secure. The CSP manages the credential data and status and securely maintains the information collected during the Initialization Phase. The token is used repeatedly by the Subscriber (who is acting as a Claimant) to authenticate to remote Verifiers. Each time the token is used for remote authentication, the following functions occur:

1. The Claimant submits his or her token for authentication;
2. The token is used in a remote authentication protocol between the Claimant and the Verifier;
3. The Verifier validates the token and credential to authenticate the Claimant (at this point the Claimant has proven to the Verifier that he or she is a Subscriber);
4. The Verifier may then provide an Identity Assertion to the Relying Party that provides a service to the Subscriber.

The final phase of the token and credential lifecycle is the termination phase, during which the token may be decommissioned (cleared of any data so it cannot be used for a future authentication session). In the termination phase, the token and credential may be renewed or re-issued; or, the credential may be revoked and/or destroyed. Typically, the Subscriber authenticates to the CSP using his or her existing, unexpired token and credential in order to request re-issuance of a new token and credential. If the Subscriber fails to request token and credential re-issuance prior to their expiration or revocation, he or she will be required to repeat the registration process to obtain a new token and credential.

# 6. Registration and Issuance

## *6.1. Overview*

In the registration process, an Applicant undergoes identity proofing by a trusted Registration Authority (RA). If the RA is able to verify the Applicant's identity, the CSP registers or gives the Applicant a token and issues a credential as needed to bind that token to the identity or some related attribute. The Applicant is now a Subscriber of the CSP and may use the token as a Claimant in an authentication protocol. This section describes the requirements for registration and for token and credential issuance.

The RA can be a part of the CSP, or the RA can be a separate and independent entity; however, a trusted relationship always exists between the RA and CSP. The RA or CSP maintain records of the registration. The RA and CSP can provide services on behalf of an organization or may provide services to the public. The processes and mechanisms available to the RA for identity proofing may differ as a result. Where the RA operates on behalf of an organization, the identity proofing process may be able to leverage a pre-existing relationship (e.g., the Applicant is employee or student.) Where the RA provides services to the public, the identity proofing process is generally limited to confirming publicly available information and previously issued credentials.

The registration and identity proofing process is designed, to a greater or lesser degree depending on the assurance level, to ensure that the RA/CSP knows the true identity of the Applicant. Specifically, the requirements include measures to ensure that:

- A person with the Applicant's claimed attributes exists, and those attributes are sufficient to uniquely identify a single person;
- The Applicant whose token is registered is in fact the person who is entitled to the identity;
- It is difficult for the Claimant to later repudiate the registration and dispute an authentication using the Subscriber's token.

An Applicant may appear in person to register, or the Applicant may register remotely. Somewhat different processes and mechanisms apply to identity proofing in each case. Remote registration is limited to Levels 1 through 3.

After successful identity proofing of the Applicant, the CSP is responsible for token and credential issuance (additional CSP responsibilities are discussed further in Section 8). Issuance includes establishing the authentication token and a corresponding credential for the Subscriber and delivering the token and/or credential to the Subscriber. Depending on the type of token being used, the CSP will either create a new token and supply the token to the Subscriber, or require the Subscriber to register a token that the Applicant already possesses or has newly created. In either case, the mechanism for transporting the token from the token origination point to the other party may need to be secured to ensure that the confidentiality and integrity of the newly established token is maintained.

The CSP is also responsible for the creation of a credential that binds the Subscriber's identity to his or her token and potentially other attributes. Optionally, the CSP may also need to establish and include other attributes about the Subscriber within the credential, such as his or her organizational affiliation, policies or constraints for token use.

In models where the registration and identity proofing take place separately from credential issuance, the CSP is responsible for verifying the identity of the person who is being issued the credential. In this model, issuance must be strongly bound to registration and identity proofing so that an Attacker cannot pose as a new Subscriber and attempt to collect a token/credential meant for the actual Subscriber. This attack, and similar attacks, can be thwarted by the methods described at the end of Section 6.3.1, which describes which techniques are considered appropriate for establishing the necessary binding at the various assurance levels.

### *6.2. Registration and Issuance Threats*

There are two general categories of threats to the registration process: impersonation and either compromise or malfeasance of the infrastructure (RAs and CSPs). This recommendation concentrates on addressing impersonation threats. Infrastructure threats are addressed by normal computer security controls (e.g., separation of duties, record keeping, independent audits, etc.) and are outside the scope of this document[3]. The threats to the issuance process include impersonation attacks and threats to the transport mechanism for the token and credential issuance. The table below lists the threats related to registration and issuance.

Table 1 - Registration and Issuance Threats

| Activity | Threat/Attack | Example |
|---|---|---|
| Registration[4] | Impersonation of claimed identity | An Applicant claims an incorrect identity by using a forged driver's license. |
| | Repudiation of registration | A Subscriber denies registration, claiming that they did not register that token. |
| | | |
| Issuance | Disclosure | Password created by the CSP for a Subscriber is copied by an attacker as it is transported from the CSP to the Subscriber during token establishment. |
| | Tampering | New password created by the Subscriber is modified by an attacker as it is being submitted to the CSP during token establishment phase. |

[3] See NIST SP800-53, *Recommended Security Controls For Federal Information Systems* for appropriate security controls.

[4] Some impostors may attempt to register as any Subscriber in the system and other impostors may wish to register as a specific Subscriber.

| Activity | Threat/Attack | Example |
|---|---|---|
| | Unauthorized issuance | A person claiming to be the Subscriber (but in reality is not the Subscriber) is issued credentials for that Subscriber. |

### *6.2.1 Threat Mitigation Strategies*

Registration fraud can be deterred by making it more difficult to accomplish or increasing the likelihood of detection. This recommendation deals primarily with methods for making impersonation more difficult, however it does prescribe certain methods and procedures that may help to prove who carried out an impersonation. At each level, methods are employed to determine that a person with the claimed identity exists, the Applicant is the person who is entitled to that identity and the Applicant cannot later repudiate the registration. As the level of assurance increases, the methods employed provide increasing resistance to casual, systematic and insider impersonation. The table below lists strategies for mitigating threats to the registration and issuance processes.

Table 2 - Registration and Issuance Threat Mitigation Strategies

| Activity | Threat/Attack | Mitigation Strategy |
|---|---|---|
| Registration | Impersonation of claimed identity | Registration Authorities request documentation that proves the identity of the Applicant and makes it more difficult for imposters to successfully pass the identity proofing step. Government issued documents such as driver's licenses, passports and social security cards presented by the Applicant assert the identity of the Applicant. |
| | | Provide non-government issued documentation (e.g. electricity bills in the name of the Applicant with the current address of the Applicant printed on the bill, or a credit card bill) to help in achieving a higher level of confidence in the identity of the Applicant. |
| | Repudiation of registration | Have the Applicant sign a form acknowledging participation in the registration activity. |
| | | |
| Issuance | Disclosure | Issue token in person, by physically mailing it in a sealed envelope to a secure location, or through the use of a communication protocol that protects the confidentiality of the session data. |
| | Tampering | Issue credentials in person, by physically mailing storage media in a sealed envelope, or through the use of a communication protocol that protects the integrity of the session data. |
| | | Establish a procedure that allows the Subscriber to authenticate the CSP as the |

| Activity | Threat/Attack | Mitigation Strategy |
|---|---|---|
| | | source of any token and credential data that he or she may receive. |
| | Unauthorized issuance | Establish procedures to ensure that the individual who receives the token is the same individual who participated in the registration procedure. |

## 6.3. Registration, and Issuance Assurance Levels

The following sections list the NIST recommendations for registration and issuance for the four levels corresponding to the OMB guidance. As noted in the OMB guidance, Levels 1 and 2 recognize the use of anonymous credentials. When anonymous credentials are used to imply membership in a group, the level of proofing shall be consistent with the requirements for the identity credential of that level. Explicit requirements for registration processes for anonymous credentials are not specified, as they are unique to the membership criteria for each specific group.

### 6.3.1 Requirements per Assurance Level

At Level 2 and higher, records of registration shall be maintained either by the RA or by the CSP, depending on the context. Either the RA or the CSP shall maintain a record of each individual whose identity has been verified, and the steps taken to verify his or her identity, including the evidence required in the sections below. The CSP shall be prepared to provide records of identity proofing to Relying Parties as necessary. The identity proofing and registration process shall be performed according to a written policy or *practice statement* that specifies the particular steps taken to verify identities.
If the RA and CSP are remotely located, and communicate over a network, the entire registration transaction between the RA and CSP shall be cryptographically authenticated using an authentication protocol that meets the requirements for the assurance level of the registration, and any secrets transmitted shall be encrypted using an Approved encryption method.

The CSP shall be able to uniquely identify each Subscriber and the associated tokens and the credentials issued to that Subscriber. The CSP shall be capable of conveying this information to Verifiers and Relying Parties. At Level 1, the name associated with the Subscriber is provided by the Applicant and accepted without verification. At Level 2, the name associated with the Subscriber may be pseudonymous but the RA or CSP shall know the actual identity of the Subscriber. In addition, pseudonymous Level 2 credentials must be distinguishable from Level 2 credentials that contain meaningful names.

At Level 3 and above, the name associated with the Subscriber shall be meaningful. At all levels, personal identifying information collected as part of the registration process shall be protected from unauthorized disclosure or modification.

The following text establishes registration requirements specific to each level. There are no level-specific requirements at Level 1. Both in-person and remote registration are

permitted for Levels 2 and 3. Explicit requirements are specified for each scenario in Levels 2 and 3. Only in-person registration is permitted at Level 4.

At Level 2 and higher, the Applicant supplies his or her full legal name, an address of record, and date of birth, and may, subject to the policy of the RA or CSP, also supply other individual identifying information. Detailed level-by-level identity proofing requirements are stated in Table 3 below.

Table 3 - Identity Proofing Requirements by Assurance Level

| | **In-Person** | **Remote** |
|---|---|---|
| **Level 2** | | |
| Basis for issuing credentials | Possession of a valid current primary Government Picture ID that contains Applicant's picture, and either address of record or nationality (e.g. drivers license or Passport) | Possession of a valid Government ID (e.g. a driver's license or Passport) number and a financial account number (e.g., checking account, savings account, loan or credit card) with confirmation via records of either number. |
| RA actions | Inspects photo-ID, compare picture to Applicant, record ID number, address and DoB. If ID appears valid and photo matches Applicant then:<br>a) If ID confirms address of record, authorizes or issues credentials and sends notice to address of record, or;<br>b) If ID does not confirm address of record, issues credentials in a manner that confirms address of record. | • Inspects both ID number and account number supplied by Applicant (e.g. for correct number of digits). Verifies information provided by Applicant including ID number OR account number through record checks either with the applicable agency or institution or through credit bureaus or similar databases, and confirms that: name, DoB, address other personal information in records are on balance consistent with the application and sufficient to identify a unique individual.<br>• Address confirmation and notification:<br>a) Sends notice to an address of record confirmed in the records check or;<br>b) Issues credentials in a manner that confirms the address of record supplied by the Applicant; or<br>c) Issues credentials in a manner that confirms the ability of the Applicant to receive telephone communications or e-mail at number or e-mail address associated with the Applicant in records. Any secret sent over an unprotected channel shall be reset upon first use. |
| **Level 3** | | |
| Basis for issuing credentials | Possession of verified current primary Government Picture ID that contains Applicant's picture and either address of record or nationality (e.g. drivers license or passport) | Possession of a valid Government ID (e.g. a driver's license or Passport) number and a financial account number (e.g., checking account, savings account, loan or credit card) with confirmation via records of both |

| | In-Person | Remote |
|---|---|---|
| | | numbers. |
| RA actions | Inspects Photo-ID and verify via the issuing government agency or through credit bureaus or similar databases. Confirms that: name, DoB, address and other personal information in record are consistent with the application. Compares picture to Applicant, record ID number, address and DoB. If ID is valid and photo matches Applicant then:<br>a) If ID confirms address of record, authorize or issue credentials and send notice to address of record, or;<br>b) If ID does not confirm address of record, issues credentials in a manner that confirms address of record. | • Verifies information provided by Applicant including ID number AND account number through record checks either with the applicable agency or institution or through credit bureaus or similar databases, and confirms that: name, DoB, address and other personal information in records are consistent with the application and sufficient to identify a unique individual.<br>• Address confirmation:<br>a) Issues credentials in a manner that confirms the address of record supplied by the Applicant; or<br>b) Issues credentials in a manner that confirms the ability of the Applicant to receive telephone communications at a number associated with the Applicant in records, while recording the Applicant's voice or using equivalent alternative means to establish non-repudiation. |
| **Level 4** | | |
| Basis for issuing credentials | In-person appearance and verification of two independent ID documents or accounts, meeting the requirements for Level 3 (in-person and remote), one of which must be current primary Government Picture ID that contains Applicant's picture, and either address of record or nationality (e.g. drivers license or passport), and a new recording of a biometric of the Applicant at the time of application. | Not Applicable |
| RA actions | • *Primary Photo ID:*<br>Inspects Photo-ID and verify via the issuing government agency, compares picture to Applicant, record ID number, address and DoB.<br>• *Secondary Government ID or financial account*<br>a) Inspects Photo-ID and if apparently valid, compare picture to Applicant, record ID number, address and DoB, or;<br>b) Verifies financial account number supplied by Applicant through record checks or through credit bureaus or similar databases, and confirms that: name, DoB, address other personal information in records are on balance consistent with the application and sufficient to identify a unique individual. | Not Applicable |

| | In-Person | Remote |
|---|---|---|
| | • *Record Current Biometric*<br>Records a current biometric (e.g. photograph or fingerprints) to ensure that Applicant cannot repudiate application.<br>• *Confirm Address*<br>Issues credentials in a manner that confirms address of record. | |

At Level 2, employers and educational institutions who verify the identity of their employees or students by means comparable to those stated above for Level 2 may elect to become an RA or CSP and issue credentials to employees or students, either in-person by inspection of a corporate or school issued picture ID, or through on-line processes, where notification is via the distribution channels normally used for sensitive, personal communications.

Federal law, including the Bank Secrecy Act and the USA Patriot Act, impose a duty on financial institutions to "know their customers" and report suspicious transactions to help prevent money laundering and terrorist financing. Many financial institutions are regulated by Federal Agencies such as the Office of the Comptroller of the Currency (OCC) or other members of the Federal Financial Institutions Examination Council (FFIEC) and the Securities and Exchanges Commission (SEC). These regulators normally require the intuitions to implement a Customer Identification Program.

The following provisions apply to Federally regulated financial institutions, brokerages and dealers subject to such Federal regulation, that implement such a Customer Identification Program:

- At Level 2 such institutions may issue credentials to their customers via the mechanisms normally used for on-line banking or brokerage credentials, and may use on-line banking or brokerage credentials and tokens as Level 2 E-authentication credentials and tokens, provided they meet the provisions of Sections 7 through 10 for Level 2.
- At Level 3 such institutions may issue credentials to their customers via the mechanisms normally used for on-line banking or brokerage credentials, and may use on-line banking or brokerage credentials and tokens as Level 3 E-Authentication credentials and tokens, provided:
  1. The customers have been customers in good standing for a period of at least 1 year prior to the issuance of E-auth credentials, and
  2. The credentials and tokens meet the provisions of Sections 7 through 10 for Level 3.
- At Level 3 or 4 such institutions may issue credentials to their customers via the mechanisms normally used for on-line banking or brokerage credentials,

and may use on-line banking or brokerage credentials and tokens as Level 3 or 4 E-authentication credentials, provided:

1. The customers have appeared in-person before a representative of the financial institution, and the representative has inspected a Government issued primary Photo-ID and compared the picture to the customer.
2. The credentials and tokens meet all additional provisions of Section 6 as well as all provisions in Sections 7 through 10 for Level 3 or 4 as appropriate.

In some contexts, agencies may choose to use additional knowledge based authentication methods to increase their confidence in the registration process. For example, an Applicant could be asked to supply non-public information on his or her past dealing with the agency that could help confirm the Applicant's identity.

The sensitive data collected during the registration and identity proofing stage must be protected at all times (e.g. transmission and storage) to ensure its security and privacy. Additionally, the results of the identity proofing step (which may include background investigations of the Applicant) have to be protected to ensure source authentication, confidentiality and integrity.

It is important to note that registration, identity proofing, and token and credential issuance represent different goals of the same process. In many cases, however, this process may be broken up into a number of separate physical encounters and electronic transactions. (Two electronic transactions are considered to be separate if they are not part of the same protected session.) In these cases, the following methods shall be used to ensure that the same party acts as Applicant throughout the process:

- At Level 1, there is no specific requirement, however some effort should be made to uniquely identify and track applications.
- At Level 2, the Applicant shall identify himself/herself in any new electronic transaction (beyond the first transaction or encounter) by presenting a temporary secret which was established during a prior transaction or encounter, or sent to the Applicant's phone number, email address, or physical address of record. The Applicant shall identify himself/herself in person by either using a secret as described above, or through the use of a biometric that was recorded during a prior encounter.
- At Level 3, the Applicant shall identify himself/herself in each new electronic transaction by presenting a temporary secret which was established during a prior transaction or encounter, or sent to the Applicant's physical address of record. The Applicant shall identify himself/herself in person by either using a secret as described above, or through the use of a biometric that was recorded during a prior encounter. Temporary secrets shall not be reused.
- At Level 4, the Applicant shall identify himself/herself in person through the use of a biometric that was recorded during a prior encounter.

A common reason for breaking up the registration process as described above is to allow the subscriber to register or download software tokens in two or more different computing environments. This is permissible as long as the tokens individually meet the appropriate assurance level. However, if the exact number of tokens to be issued is not agreed upon early in the registration process, then the tokens should be distinguishable so that Verifiers will be able to detect whether any suspicious activity occurs during the first few uses of a newly issued token.

### *6.3.2 Mapping FPKI Certificate Policies to Registration Levels*

The identity proofing and certificate issuance processes specified in the Federal PKI Certificate Policies [FCBA1, FBCA2, FBCA3] are considered equivalent to the requirements specified in the preceding section in accordance with Appendix B.

However, agencies are not limited to relying upon only those certificates by CAs cross-certified with the Federal Bridge CA. At Level 2, agencies may choose to rely on any CA that has been determined to meet the identity proofing and registration requirements stated in the general requirements, Section 6.3. At Levels 3 and 4, PKI credentials shall be issued by a CA cross-certified with the Federal Bridge CA under one of the certificate policies identified above, or a policy mapped to one of those policies.

# 7 Tokens

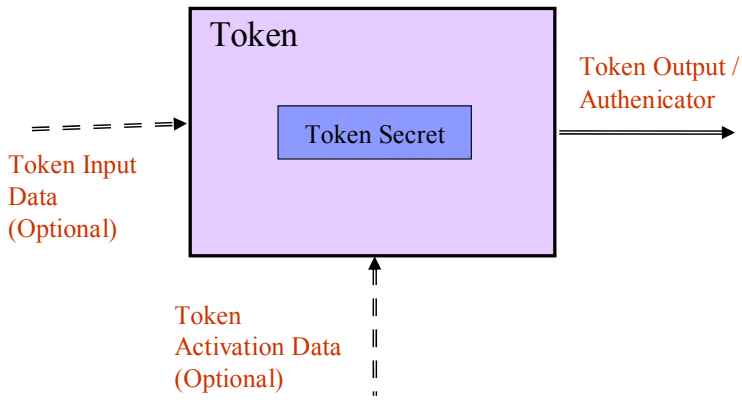
## 7.1 Overview

*A token is something that the Claimant possesses and controls (typically a key or password) used to authenticate the Claimant's identity*. A token incorporates one or more of the three factors of authentication - something you know, something you have, and something you are.

Examples of tokens include a memorized password, a one-time password device, or a smart card with an embedded cryptographic key. In the context of these guidelines, every token contains a secret. Figure 3 below depicts an abstract model for a token. The outer box is the token itself, which may have a physical manifestation (such as a smart card, or a hardware one-time password device), an electronic manifestation (such as a software cryptographic key) or it may only exist in human memory (such as a password). The inner box represents the actual secret that is inherent within the token. For tokens that have a physical or electronic manifestation, the secret is embedded within the token. For tokens that do not have a physical or electronic manifestation, the token and token secret are synonymous, and the inner and outer boxes merge.

In the model below, the output of the token is the *authenticator*, which is the value that is provided to the protocol stack for transmission to the Verifier to prove that the Claimant possesses and controls the token. For some token types such as passwords, the value of the authenticator is identical to the token secret. For other token types, the authenticator is a function of input data such as a nonce or a challenge sent by the Verifier and the token secret. This is shown in Figure 3 below as optional token input data. Additionally, certain tokens require activation (such as through the entry of a PIN or biometric) in order to generate the authenticator. This is shown in the figure below as optional token activation data.

Figure 3 - Token Model




The *authenticator* is generated through the use of the token. In the trivial case, the authenticator may be the token *secret* itself (e.g., where the token is a password). In the

general case, an authenticator is generated by performing a mathematical function using the token secret and one or more optional token input values (a nonce or challenge):

*Authenticator = Function (<token secret> [, <nonce>] [, <challenge>] )*

In some cases, the function cannot be computed unless activation data is supplied to unlock the token. The activation data itself is not used in the computation of the authenticator, but it serves as a gating mechanism.

Tokens may present physical interfaces (such as a pin pad to enter the activation data for a token) or electronic interfaces (such as a token which can be plugged into a USB drive). Where the token interfaces are electronic, there may be opportunities to enforce access restrictions upon the token.

### 7.1.1 Single-Factor versus Multi-factor Tokens

Tokens are characterized by the number and types of authentication factors that they use. For example, a password is a token that is something you know, a biometric is something you are, and a cryptographic identification device is something you have. Tokens may be single or multi-factor tokens as described below:

- *Single-factor Token* – A token that uses one of the three factors to achieve authentication. For example, a password is something you know, and can be used to authenticate the holder to a remote system.
- *Multi-factor Token* – A token that uses two or more factors to achieve authentication. For example, a private key on a smart card that is activated via PIN is a multi-factor token. The PIN is something you know and the smart card is something you have.

### 7.1.2 Token Types

These guidelines recognize the following types of tokens for E-authentication.

- *Memorized Secret Token* – A secret shared between the Subscriber and the CSP that the Subscriber memorizes and uses to authenticate his or her identity to the Verifier. Memorized Secret Tokens are typically character strings (e.g. passwords and passphrases) or numerical strings (e.g. PINs,)
- *Pre-registered Knowledge Token* – A set of prompts/responses that the Subscriber establishes during the registration process with the CSP. Typically, the prompts and responses are such that they are easy for the Subscriber to recall from memory, and difficult for others to obtain. During E-authentication, the Claimant is asked to provide the appropriate responses to a subset of the prompts. Authentication is based on the accuracy of the responses provided by the Claimant. An example of this type of token would be establishing prompts such as “What was the first car you ever owned?” and requiring the answer to contain the year, make, model and color. Based on the

accuracy of the responses supplied by the Claimant, the Verifier may consider the authentication attempt successful. Another example is prompting a Claimant to select an image or set of images that the Subscriber memorizes during the registration phase; the Subscriber then has to identify the correct images from a set(s) of similar images.

- *Look-up Secret Token* – One or more secrets shared between the Claimant and the CSP, which are stored on a physical or electronic medium held by the Claimant. The Claimant uses the token to look up the appropriate secret(s) needed to respond to a prompt from the Verifier. These tokens may be printed on a paper or plastic medium or stored in electronic form. For example, a Claimant may be asked by the Verifier to provide a specific subset of the numeric or character strings printed on a card in table format. If the Claimant is able to provide the correct response, the Verifier successfully authenticates the Subscriber.
- *Out of Band Token* – The combination of a physical device or system with a uniquely addressable identifier, and a secret authenticator that is transmitted to the device by a verifying party. The device is owned or controlled by the Claimant, and supports communication over a channel that is separate from the primary channel for E-authentication. The secret authenticator transmitted to the device is valid for one time use and expires within minutes. An Out of Band Token may be used to support E-authentication through a protocol that ties the proof of possession/control of the Out of Band Token to the primary E-authentication activity. For example, a Claimant attempts to log into a website and receives a text message on his or her cellular phone, PDA, pager, or land line (which has to be pre-registered with the CSP during the registration phase) with a random authenticator that has to be typed in as a part of the online authentication protocol.
- *Single-factor (SF) One Time Password (OTP) Device* – A hardware device that supports the spontaneous generation of one time passwords. This device has an embedded secret that is used as the seed for generation of one time passwords, and does not require activation through a second factor. Authentication is accomplished by providing an acceptable one time password and thereby proving possession of the device.
- *Single-factor (SF) Cryptographic Device* – a hardware device that performs cryptographic operations on input provided to the device. This device does not require activation through a second factor of authentication. This device uses embedded symmetric or asymmetric cryptographic keys. Authentication is accomplished by proving possession of the device.

- *Multi-factor (MF) Software Cryptographic Token* – A cryptographic key that is stored on disk or some other "soft" media and requires activation through a second factor of authentication. Authentication is accomplished by proving possession and control of the key.

- *Multi-factor (MF) One-Time Password (OTP) Device* – A personal hardware device that generates "one-time" passwords for use in authentication and which requires activation through a second factor of authentication. The second factor of authentication may be achieved through some kind of integral entry pad, an integral biometric (e.g., fingerprint) reader or a direct computer interface (e.g., USB port). The one-time password is typically displayed on the device and manually input to the Verifier as a password, although direct electronic input from the device to a computer is also allowed in some cases.
- *Multi-factor (MF) Cryptographic Device* – A hardware device that contains a protected cryptographic key and which requires activation through a second authentication factor. Authentication is accomplished by proving possession of the device and control of the key.

### *7.1.3 Token Usage*

For E-authentication, a single token may be used or a combination of two or more tokens (multi-token scheme) may be used, as described below.

- *Single token authentication* – Only one token is required to verify the identity of the Claimant. For example, when a Claimant attempts to log into a password protected website, the Claimant has to use a username and password. The password is the token and once entered, the user is allowed access; therefore, a single token has allowed the user to authenticate to the website.
- *Multi-token authentication* – Two or more tokens are required to verify the identity of the Claimant. A Relying Party that requires a Claimant to enter a password token and use a single factor cryptographic device is an example of multi-token authentication. A multi-token authentication scheme is additionally considered to be multi-factor if authentication requires tokens which, in combination, include at least two of the three authentication factors.

A PIN activated smart card with a private key is a single-token, multi-factor authentication scheme. A combination of a password and a callback to a registered cell phone is a valid multi-token, multi-factor authentication scheme. A password only scheme represents single-token, single-factor authentication.

When a multi-factor token or a multi-token authentication scheme is being used, the security properties of each factor or of each token are considered additive in nature. If one factor of a multi-factor scheme or one token of a multi-token scheme has the desired properties for a given assurance level, it is considered sufficient.

Some E-authentication schemes use one token to gain access to a second token. These are considered single token schemes since all that is needed to gain access is the initial token. Therefore, when this model is used, the compound solution is only as strong as the token with the lowest assurance level.

### 7.1.4 Multi-Stage Authentication Using Tokens

Some cryptographic mobility solutions allow full or partial cryptographic keys to be stored on an online server, and downloaded to the Claimant's local system after suitable authentication using a password or passphrase. The downloaded keys are protected such that activation data (such as a user password) is required in order to apply the key. For example, a software cryptographic token may be downloaded to a Claimant's local system after the Claimant successfully authenticates to a remote token server system using a passphrase; subsequently, the Claimant can use the downloaded software cryptographic token to authenticate to a remote Verifier for E-authentication. This type of solution is considered only as strong as the password provided by the Claimant unless an additional authentication factor is required to successfully download the token.

In certain circumstances, it may be desirable to raise the assurance level of an E-authentication session between a Subscriber and a Relying Party in the middle of the application session. Thus, it is possible to combine the first stage authentication prior to launching a session, with a second stage authentication to elevate the assurance level for the same session. This guideline also recognizes the use of electronic transaction receipts or confirmation mechanisms using additional token(s) to achieve the required assurance level for the requested service. In these multi-stage authentication scenarios, the level of assurance of the combination of the two stages can be no higher than that possible through a multi-token authentication scheme using the same set of tokens. Table 7 describes the highest level of assurance achievable through a combination of two token types.

Consider that a Subscriber has successfully authenticated at assurance Level 2 using a Memorized Secret Token, and is interacting with a Relying Party to obtain services at Level 2. At some point, the Subscriber requests services from the Relying Party that require authentication at assurance Level 3. The Relying Party then generates a transaction record (for the requested transaction or service) and sends it to the Subscriber using a Pre-registered Out of Band Token (such as a cell phone). If the Subscriber is able to prove possession and control of the out of band token by receipt and confirmation of the transaction record (either in-band or out of band) the Relying Party considers the Subscriber authenticated at Level 3 and proceeds to provide the requested service.

## 7.2 Token Threats

An Attacker who can gain control of a token will be able to masquerade as the token's owner. Threats to tokens can be categorized based on attacks on the types of authentication factors that comprise the token:

- *Something you have* may be stolen from the owner or cloned by the Attacker. For example, an Attacker who gains access to the owner's computer might copy a software token. A hardware token might be stolen or duplicated.
- *Something you know* may be disclosed to an Attacker. The Attacker might guess a password or PIN. Where the token is a shared secret, the Attacker

could gain access to the CSP or Verifier and obtain the secret value. An Attacker may install malicious software (e.g., a keyboard logger) to capture this information. Finally, an Attacker may determine the secret through off-line attacks on network traffic from an authentication attempt.

- *Something you are* may be replicated. An Attacker may obtain a copy of the token owner's fingerprint and construct a replica.

This document assumes that the owner of the token(s) used for E-authentication is not colluding with the Attacker who is attempting to falsely authenticate to the Verifier. With this assumption in mind, the threats to the token(s) used for E-authentication are listed below along with some examples.

Table 4 - Token Threats

| Token Threats/Attacks | Description | Examples |
|---|---|---|
| Theft | A token with a physical manifestation is stolen by an Attacker. | Hardware cryptographic device stolen. |
| | | One-time password device stolen. |
| | | Lookup token stolen. |
| | | Cell phone stolen. |
| Duplication | The Subscriber's token has been copied with or without his or her knowledge. | Passwords written on paper disclosed. |
| | | Passwords stored in electronic file copied. |
| | | Software PKI token (private key) copied. |
| | | Lookup token copied. |
| Eavesdropping | 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. |
| Offline cracking | The token is exposed using analytical methods outside the authentication mechanism. | Differential power analysis on stolen hardware cryptographic token. |
| | | Software PKI token is subjected to dictionary attack to identify correct PIN to use the private key within token. |
| 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. |
| | | Password revealed by bank Subscriber in response to an email inquiry from a Phisher pretending to represent the bank. |
| | | Password revealed by Subscriber at a bogus Verifier website reached through DNS re-routing. |
| 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 Subscriber to officemate asking for password on behalf of Boss. |
| | | Token revealed by Subscriber in telephone inquiry from masquerading system administrator. |
| Online guessing | The Attacker connects to the Verifier online and attempts to guess a valid token authenticator in the context of that Verifier. | Online dictionary attacks to guess passwords. |
| | | Online guessing of secret token registered to legitimate Claimant. |

### 7.2.1 Threat Mitigation Strategies

Token related mechanisms that assist in mitigating the threats identified above are summarized in the table below.

Table 5 - Mitigating Token Threats

| Token Threat/Attack | Threat Mitigation Mechanisms |
|---|---|
| Theft | - Use multi-factor tokens which need to be activated through a PIN or biometric. |
| Duplication | - Use tokens that are difficult to duplicate, such as hardware cryptographic tokens. |
| Eavesdropping | - Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator.<br>- Use tokens that generate authenticators based on a token input value. |
| Offline cracking | - Use a token with a high entropy token secret<br>- Use a token that locks up after a number of repeated failed activation attempts. |
| Phishing or pharming | - Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator.<br>- Use tokens that generate authenticators based on a token input value. |
| Social engineering | - Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator.<br>- Use tokens that generate authenticators based on a token input value. |
| Online guessing | - Use tokens that generate high entropy authenticators. |

There are several other strategies that may be applied to mitigate the threats described in the above table:

- *Multiple factors* raise the threshold for successful attacks. If an Attacker needs to steal a cryptographic token and guess a password, the work factor may be too high.
- *Physical security mechanisms* may be employed to protect a stolen token from duplication. Physical security mechanisms can provide tamper evidence, detection, and response.
- *Imposing password complexity rules* may reduce the likelihood of a successful guessing attack. Requiring the use of long passwords that don't appear in common dictionaries may force Attackers to try every possible password.
- *System and network security controls* may be employed to prevent an Attacker from gaining access to a system or installing malicious software.
- *Out of band techniques* may be employed to verify proof of possession of registered devices (e.g., cell phones) or identifiers (e.g., email IDs).

## 7.3 Token Assurance Levels

This section discusses the requirements for tokens used at various levels of assurance.

### 7.3.1 Requirements per Assurance Level

The following sections list token requirements for single and multi-factor authentication.

#### *7.3.1.1 Single Token Authentication*

The table below lists the assurance levels that may be achieved by each of the token types when used in a single-token authentication scheme. The requirements for each token are listed per assurance level. If token requirements are listed only at one assurance level, and checks are indicated at lower levels, the token may be used at lower levels but must satisfy the requirements of the highest level.

Table 6 - Token Requirements Per Assurance Level

| Token Type | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| Memorized Secret Token | The probability that an Attacker can guess a valid authenticator[5], over the lifetime of the token, must be less than $2^{-10}$ | The probability that an Attacker can guess a valid authenticator, over the lifetime of the token, must be less than $2^{-14}$<br><br>Authenticators must have greater than 10 bits of min-entropy[6] | | |
| Pre-Registered Knowledge Token | The probability that an Attacker can guess a valid authenticator, over the lifetime of the token, must be less than $2^{-10}$ | The probability that an Attacker can guess a valid authenticator, over the lifetime of the token, must be less than $2^{-14}$<br><br>Authenticators must have greater than 10 bits of min-entropy | | |

[5] It is assumed, for this purpose, that the Attacker is using an online guessing attack. The guessing entropy necessary to meet this requirement may vary depending on token lifetime and any throttling mechanisms which are employed by the verifier. See appendix A for more information.
[6] Appendix A provides an example of a password policy which may be used to ensure that passwords meet this requirement.

| Token Type | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| Look-up Secret Token | The probability that an Attacker can guess a valid authenticator, over the lifetime of the token, must be less than $2^{-10}$ | The probability that an Attacker can guess a valid authenticator, over the lifetime of the token, must be less than $2^{-14}$<br><br>Authenticators must have greater than 10 bits of min-entropy | | |
| Out of Band Token | The probability that an Attacker can guess a valid authenticator, over the lifetime of the token, must be less than $2^{-10}$ | The probability that an Attacker can guess a valid authenticator, over the lifetime of the token, must be less than $2^{-14}$<br><br>Authenticators must have greater than 10 bits of min-entropy<br><br>The authenticator must have a limited lifetime, on the order of minutes and can only be used once. | | |
| Single Factor One Time Password Device | √ | Must use approved block cipher or hash function to combine a symmetric key stored on device with a nonce to generate a one-time password. The cryptographic module performing this operation shall be validated at FIPS 140-2 Level 1 or higher.<br><br>The nonce may be a date and time, or a counter generated on the device<br><br>The one-time password must have a limited lifetime, on the order of minutes. | | |

| Token Type | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| Single Factor Cryptographic Device | √ | The cryptographic module shall be validated at FIPS 140-2 Level 1 or higher. | | |
| MF Software Cryptographic Token | √ | √ | The cryptographic module shall be validated at FIPS 140-2 Level 1 or higher. Each authentication shall require entry of the password or other activation data and the unencrypted copy of the authentication key shall be erased after each authentication. | |
| MF OTP Hardware Token | √ | √ | √ | Cryptographic module must be FIPS 140-2 validated, Level 2 or higher; with physical security at Level 3 or higher.<br><br>The one-time password shall be generated by using an Approved block cipher or hash function to combine a symmetric key stored on a personal hardware device with a nonce to generate a one-time password.<br><br>The nonce may be a date and time, a counter generated on the device. Each authentication shall require entry of a password or other activation data through an integrated input mechanism.<br><br>The one-time password must have a limited lifetime of less than 2 minutes. |

| Token Type | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| MF Hardware Cryptographic Token | √ | √ | √ | Cryptographic module must be FIPS 140-2 validated, Level 2 or higher; with physical security at Level 3 or higher. Must require the entry of a password or a biometric to activate the authentication key. Must not allow the export of authentication keys; |

#### *7.3.1.2 Multi-Token Authentication*

When two of the token types are combined for a *multi-token authentication scheme*, the table below shows the highest possible assurance level that can be achieved by the combination.[7]

Table 7 - Assurance Levels for Multi-Token E-Authentication Schemes[8]

| | **Memorized Secret Token** | **Pre-registered Knowledge Token** | **Look-up Secret Token** | **Out of Band Token** | **SF OTP Device** | **SF Crypto-graphic Device** | **MF Software Crypto-graphic Token** | **MF OTP Device** | **MF Crypto-graphic Device** |
|---|---|---|---|---|---|---|---|---|---|
| Memorized Secret Token | Level 2 | Level 2 | Level 3 | Level 3 | Level 3 | Level 3 | Level 3 | Level 4 | Level 4 |
| Pre-registered Knowledge Token | X | Level 2 | Level 3 | Level 3 | Level 3 | Level 3 | Level 3 | Level 4 | Level 4 |
| Look-up Secret Token | X | X | Level 2 | Level 2 | Level 2 | Level 2 | Level 3 | Level 4 | Level 4 |
| Out of Band Token | X | X | X | Level 2 | Level 2 | Level 2 | Level 3 | Level 4 | Level 4 |
| SF OTP Device | X | X | X | X | Level 2 | Level 2 | Level 3 | Level 4 | Level 4 |
| SF Cryptographic Device | X | X | X | X | X | Level 2 | Level 3 | Level 4 | Level 4 |
| MF Software Cryptographic Token | X | X | X | X | X | X | Level 3 | Level 4 | Level 4 |
| MF OTP Device | X | X | X | X | X | X | X | Level 4 | Level 4 |
| MF Cryptographic Device | X | X | X | X | X | X | X | X | Level 4 |

[7] Note that the table displays tokens that exhibit the properties of "something you have" and "something you know".
[8] The boxes marked with an "x" denote that the combination already appears in the table

The principles used in generating the table above are as follows. To achieve Level 3 and above, one of the two tokens used in a multi-token scheme shall be rated at Level 3, or, both tokens shall be rated at Level 2 and represent two different factors of authentication. For example, a Memorized Secret Token combined with a Lookup Secret Token can be used to achieve Level 3 authentication, since the lookup secret token is "something you have" and the Memorized Secret Token is "something you know". However, combining a MF software cryptographic token (which is rated at Level 3) and a Memorized Secret Token (which is rated at Level 2) achieves an overall level of 3, since the addition of the Memorized Secret Token does not increase the assurance of the combination. To achieve Level 4 with a single token or token combination, one of the tokens needs to be usable with an authentication mechanism that strongly resists man-in-the-middle attacks – this entails an electronic interface which may be placed under access control by the Claimant's operating system. For example, a MF OTP device combined with a TLS protocol such that the data exchange session is protected using a secret generated as a result of mutual authentication may achieve Level 4 assurance.

# 8 Token and Credential Management

## *8.1 Overview*

A credential is an object that authoritatively binds an identity (and optionally, additional attributes) to a token possessed and controlled by a person. Credentials are used to validate the token used by the Claimant and establish the identity of the Claimant as the Subscriber. The credential management activities used within a remote electronic authentication solution have an impact on the assurance level of the solution. This section discusses token and credential management activities performed by the CSP subsequent to the registration, identity proofing and issuance activities described in Section 6. This includes the lifecycle management activities for the token and credential.

Credentials can be categorized as described below from a confidentiality protection perspective:

- *Private Credentials* – This type of credential object links the user's identity with a representation of the token in a way that the exposure of the credential to unauthorized parties can lead to an exposure of the token secret. This type of credential may be used to derive, guess or crack the value of the token secret or spoof the possession of the token. Therefore, it is essential that the contents of the credential be kept confidential. For example, a password file that contains hashed values of passwords for users of a system needs to be confidentiality protected to ensure that the hashed password values do not get into the hands of an Attacker who can launch offline attacks against the hashed password values.
- *Public Credentials* – This type of credential object links the user identity to a representation of the token in a way that exposure of the credential does not lead to an exposure of the token secret. The token representation within the credential cannot be used to derive, guess or crack the value of the token secret or spoof the possession of the token. These types of credentials may be shared widely and have little or no confidentiality requirements. For example, a digital certificate binds the user's identity to the public key; it is effectively impossible to derive the value of the private key given the value of the public key.

Credentials can also be categorized based upon the strength of the credential binding technique from an integrity protection perspective, as described below:

- *Strongly Bound Credentials* – The association between the identity and the token within strongly bound credentials cannot be easily undone. For example, a digital signature binds the identity to the public key in a public key certificate; tampering of this signature can be easily detected through signature validation.

- *Weakly Bound Credentials* – The association between the identity and the token within a weakly bound credential can be readily undone and a new association can be readily created. For example, a password file is a weakly bound credential since anyone who has "write" access to the password file can potentially update the associations contained within the file.

Strongly bound credential mechanisms require little or no additional integrity protection; whereas weakly bound credentials require additional integrity protection or access controls to ensure that unauthorized parties cannot spoof or tamper with the binding of the identity to the token representation within the credential.

Unencrypted password files are private credentials that are weakly bound, and hence need to be afforded confidentiality as well as integrity protection. Signed password files are private credentials that are strongly bound. An unsigned pairing of a public key and the name of its owner is an example of a public credential that is weakly bound. Finally, a signed public key certificate represents a public credential that is strongly bound.

CSPs and Verifiers are trusted to obey the requirements in this section as well as Section 9.

### *8.1.1 Token and Credential Management Activities*

The Credential Service Provider (CSP) manages tokens and credentials. After the Registration Authority (RA) establishes the Applicant's identity, the CSP is responsible for generating credentials and supplying the Subscriber with a token or allowing the Subscriber to register his or her own token as described in Section 6. The CSP is responsible for some or all of the following token and credential management activities following issuance of the token and credential:

- *Credential storage* – After the credential has been created, the CSP may be responsible for maintaining the credentials in storage. In cases where the credentials are stored by the CSP, the level of security afforded to the credential will depend on the type of credential issued. For "private credentials", additional confidentiality mechanisms are required in storage, whereas for "public credentials", this is not necessary. Similarly, for "weakly bound credentials", additional integrity protection is needed in storage, unlike "strongly bound credentials. Finally, credentials need to be available to allow CSPs and Verifiers to determine the identity of the corresponding token owner.
- *Token and credential verification services* – In many E-authentication scenarios, the Verifier and the CSP are part of the same entity. In these cases, the CSP is responsible for providing the Verifier with the information needed to facilitate the token and credential verification process. The CSP may provide token and credential verification services to Verifiers. For example, the Verifier may request the CSP to validate the password submitted by the Claimant against the CSP's local password database.

- *Token and credential renewal /re-issuance* – Certain types of tokens and credentials may support the process of renewal or re-issuance. During renewal, the usage or validity period of the token and credential is extended without changing the Subscriber's identity or token. During re-issuance, a new credential is created for a Subscriber with a new identity and/or a new token.

  The CSP establishes suitable policies for renewal and re-issuance of tokens and credentials. The CSP may establish a time period prior to the expiration of the credential, when the Subscriber can request renewal or re-issuance following successful authentication using his or her existing, unexpired token and credential. For example, a digital certificate may be renewed for another year prior to the expiry of the current certificate by proving possession of the existing token (i.e., the private key).

  Once the Subscriber's credentials have expired, the Subscriber may be required to re-establish his or her identity with the CSP; this is typically the case with CSPs that issue digital certificates. Conversely, the CSP may establish a grace period for the renewal or re-issuance of an expired credential, such that the Subscriber can request renewal/re-issuance of his or her credential even after it has expired without the need to re-establish his or her identity with the CSP. For example, if a Claimant attempts to login to a username/password based system on which his or her password has already expired, and the system supports a grace period, the user may be prompted to create a new password and supply the last password for verification purposes. The use of expired tokens or credentials to invoke renewal/re-issuance is more practical when the Verifier and CSP are part of the same entity.

  The public key certificate for a Subscriber may be renewed with the same public key, or may be re-issued with a new public key. Passwords are seldom renewed so that the life of the existing password is extended for another period. Usually the account name/password credential for a Subscriber is renewed by having the Subscriber select a new password.

  Tokens can be renewed using out of band delivery mechanisms. If the Subscriber uses an out of band token delivery approach, re-registration of the delivery mechanism can be equated to token renewal or re-issuance. In such a case, the subscriber must use an alternate, yet already registered delivery mechanism to deliver the token and then gain access to the CSP such that the registration data can be updated by the Subscriber or, if no alternate out of band channel was registered with the original out of band channel the subscriber must re-establish their identity with the CSP in order to update their registration data.

- *Token and credential revocation and destruction* – The CSP is responsible for maintaining the revocation status of credentials and destroying the credential at the end of its life. Explicit and elaborate revocation mechanisms may be required for "public credentials" since these credentials are disseminated

widely, possibly with a preset validity period. For example, public key certificates are revoked using Certificate Revocation Lists (CRLs) after the certificates are distributed.

"Private credentials" are held closely by the CSP, and hence the revocation and destruction of these credentials is implemented easily through an update of the CSP's local credential stores. Credentials that bind usernames/passwords are instantaneously revoked and destroyed if the CSP deletes its mapping between the username and the password. Certain types of tokens may need to be explicitly deleted or zeroized at the end of the credential life in order to permanently disable the token and prevent its unauthorized reuse. For example, a Multi-factor Hardware Cryptographic Token may need to be zeroized to ensure that all of the information pertaining to the Subscriber is deleted from the token.

The CSP may be responsible for ensuring that hardware tokens are collected and cleared of any data when the Subscriber no longer has a need for its use. The CSP may establish policies for token collection to avoid the possibility of unauthorized use of the token after it is considered out of use. The CSP may destroy such collected tokens, or zeroize them to ensure that there are no remnants of information that can be used by an Attacker to derive the token value. For example, a Subscriber who is issued a hardware OTP token by a CSP may be required by policy to return the token to the CSP at the end of its life, or when the Subscriber's association with that CSP terminates.

- *Records retention* – The CSP or its representative is responsible for maintaining a record of the registration, history, and status of each token and credential, including revocation. CSPs operated by or on behalf of executive branch agencies shall also follow either the General Records Schedule established by the National Archives and Records Administration or an agency-specific schedule as applicable. All other entities shall comply with their respective records retention policies in accordance with whatever laws apply to those entities. A minimum record retention period is required at Level 2 and above.

### *8.2 Token and Credential Management Threats*

Tokens and credentials can only be as strong as the strength of the management mechanisms used to secure them. The CSP is responsible for mitigating threats to the management operations described in the last section. Token and credential management threats are described below; they are categorized in accordance with the management activity to which they apply.

These threats represent the potential to breach the confidentiality, integrity and availability of tokens and credentials during the CSP activities, and are listed below.

Table 8 - Token and Credential Management Threats

| Token and Credential Management Activity | Threat/Attack | Example |
|---|---|---|
| Credential storage | Disclosure | Usernames and passwords stored in a system file are revealed. |
| | Tampering | The file that maps usernames to passwords within the CSP is hacked so that the mappings are modified, and existing passwords are substituted with passwords known to the attacker. |
| | | |
| Token and credential verification services | Disclosure | An attacker is able to view requests and responses between the CSP and the Verifier. |
| | Tampering | An attacker is able to masquerade as the CSP and provide bogus responses to the Verifier's password verification requests. |
| | Unavailability | The password file or the CSP is unavailable to provide password and username mappings. |
| | | Public key certificates for Claimants are unavailable to the Verifier because the directory systems are down. |
| | | |
| Token and credential renewal/re-issuance | Disclosure | Password renewed by the CSP for a Subscriber is copied by an attacker as it is transported from the CSP to the Subscriber. |
| | Tampering | New password created by the Subscriber is modified by an attacker as it is being submitted to the CSP to replace an expired password. |
| | Unauthorized renewal/re-issuance | Attacker fools the CSP into re-issuing the credential for a current Subscriber – the new credential binds the current Subscriber's identity with a token provided by the attacker. |
| | | Attacker is able to take advantage of a weak credential renewal protocol to extend the credential validity period for a current Subscriber. |
| | | |

| Token and Credential Management Activity | Threat/Attack | Example |
|---|---|---|
| Token and credential revocation/destruction | Delayed revocation/destruction of credentials | Stale CRLs allow accounts (that should have been locked as a result of credential revocation) to be used by an attacker. |
| | | User accounts are not deleted when employees leave a company leading to a possible use of the old accounts by unauthorized persons. |
| | Token use after decommissioning | A hardware token is used after the corresponding credential was revoked or expired. |

### *8.2.1 Threat Mitigation Strategies*

Token and credential management related mechanisms that assist in mitigating the threats identified above are summarized in the table below.

Table 9 - Token and Credential Threat Mitigation Strategies

| Token and Credential Management Activity | Threat/Attack | Mitigation Strategy |
|---|---|---|
| Credential storage | Disclosure | Use access control mechanisms that protect against unauthorized disclosure of credentials held in storage. |
| | Tampering | Use access control mechanisms that protect against unauthorized tampering of credentials and tokens. |
| | | |
| Token and credential verification services | Disclosure | Use a communication protocol that offers confidentiality protection. |
| | Tampering | Ensure that Verifiers authenticate the CSP prior to accepting a verification response from that CSP. |
| | | Use a communication protocol that offers integrity protection. |
| | Unavailability | Ensure that the CSP has a well developed and tested Contingency Plan. |
| | | |
| Token and credential renewal/re-issuance | Disclosure | Use a communication protocol that provides confidentiality protection of session data. |
| | Tampering | Use a communication protocol that allows the Subscriber to authenticate the CSP prior to engaging in token re-issuance activities and protects the integrity of the data passed. |

| Token and Credential Management Activity | Threat/Attack | Mitigation Strategy |
|---|---|---|
| | Unauthorized renewal/re-issuance | Establish policy that Subscriber must prove possession of the old token to successfully negotiate the re-issuance process. Any attempt to negotiate the re-issuance process using an expired or revoked token should fail. |
| | | |
| Credential revocation/destruction | Delayed revocation/destruction of credentials | Revoke/Destroy credentials as soon as notification that the credentials should be revoked or destroyed. |
| | Token use after decommissioning | Destroy tokens after their corresponding credentials have been revoked. |

## 8.3 Token and Credential Management Assurance Levels

### 8.3.1 Requirements per Assurance Level

The stipulations for management of tokens and credentials by the CSP and Verifier are described below for each assurance level. The stipulations described at each level in this section are incremental in nature; requirements stipulated at lower levels are implicitly included at higher levels.

#### *8.3.1.1 Level 1*

At Level 1, the following shall be required:

- *Credential storage* – Files of shared secrets used by Verifiers at Level 1 authentication shall be protected by discretionary access controls that limit access to administrators and only to those applications that require access. Such shared secret files shall not contain the plaintext passwords; typically they contain a one-way hash or "inversion" of the password. In addition, any method allowed for the protection of long-term shared secrets at Levels 2, 3 or 4 may be used at Level 1.
- *Token and credential verification services* – Long term token secrets should not be shared with other parties unless absolutely necessary.
- *Token and credential renewal / re-issuance* – No stipulation
- *Token and credential revocation and destruction* – No stipulation
- *Records retention* – No Stipulation

#### *8.3.1.2 Level 2*

At Level 2, the following shall be required:

- *Credential storage* – Files of shared secrets used by CSPs at Level 2 shall be protected by discretionary access controls that limit access to administrators and only to those applications that require access. Such shared secret files shall not contain the plaintext passwords or secrets; two alternative methods may be used to protect the shared secret:

  1. Passwords may be concatenated to a variable salt (variable across a group of passwords that are stored together) and then hashed with an Approved algorithm so that the computations used to conduct a dictionary or exhaustion attack on a stolen password file are not useful to attack other similar password files. The hashed passwords are then stored in the password file. The variable salt may be composed using a global salt (common to a group of passwords) and the username (unique per password) or some other technique to ensure uniqueness of the salt within the group of passwords.

  2. Shared secrets may be stored in encrypted form using Approved encryption algorithms and modes, and the needed secret decrypted only when immediately required for authentication. In addition, any method allowed to protect shared secrets at Level 3 or 4 may be used at Level 2.

- *Token and credential verification services* – Long term shared authentication secrets, if used, shall never be revealed to any party except the Subscriber and CSP (including Verifiers operated as a part of the CSP); however, session (temporary) shared secrets may be provided by the CSP to independent Verifiers.

  Cryptographic protections are required for all messages between the CSP and Verifier which contain private credentials or assert the validity of weakly bound or potentially revoked credentials. Private credentials shall only be sent through a protected channel to an authenticated party to ensure confidentiality and tamper protection.

  The CSP may send the Verifier a message, which either asserts that a weakly bound credential is valid, or that a strongly bound credential has not been subsequently revoked. In this case, the message shall be logically bound to the credential, and the message, the logical binding, and the credential shall all be transmitted within a single integrity protected session between the Verifier and the authenticated CSP. If revocation is an issue, the integrity protected messages shall either be time stamped, or the session keys shall expire with an expiration time no longer than that of the revocation list. Alternatively, the time stamped message, binding, and credential may all be signed by the CSP, although, in this case, the three in combination would comprise a strongly bound credential with no need for revocation.

- *Token and credential renewal/re-issuance* – The CSP shall establish suitable policies for renewal and re-issuance of tokens and credentials. Proof-of-possession of the unexpired current token shall be demonstrated by the Claimant prior to the CSP allowing renewal and re-issuance. Passwords shall

not be renewed; they shall be re-issued. After expiry of current token, renewal and re-issuance shall not be allowed. All interactions shall occur over a protected channel such as SSL/TLS. Secondary credentials must never be renewed or re-issued.

- *Token and credential revocation and destruction* – CSPs shall revoke or destroy credentials and tokens within 72 hours after being notified that a credential is no longer valid or a token is compromised to ensure that a Claimant using the token cannot successfully be authenticated. If the CSP issues credentials that expire automatically within 72 hours (e.g. issues fresh certificates with a 24 hour validity period each day) then the CSP is not required to provide an explicit mechanism to revoke the credentials. CSPs that register passwords shall ensure that the revocation or de-registration of the password can be accomplished in no more than 72 hours. CAs cross-certified with the Federal Bridge CA at the Citizen and Commerce Class Basic, Medium and High or Common Certificate Policy levels are considered to meet credential status and revocation provisions of this level. Secondary credentials must have a lifetime less than 12 hours.
- *Records retention* – A record of the registration, history, and status of each token and credential (including revocation) shall be maintained by the CSP or its representative. The record retention period of data for Level 2 credentials is seven years and six months beyond the expiration or revocation (whichever is later) of the credential. CSPs operated by or on behalf of executive branch agencies shall also follow either the General Records Schedule established by the National Archives and Records Administration or an agency-specific schedule as applicable. All other entities shall comply with their respective records retention policies in accordance with whatever laws apply to those entities.

#### *8.3.1.3 Level 3*

At Level 3, the following is required:

- *Credential storage* – Files of long-term shared secrets used by CSPs or Verifiers at Level 3 shall be protected by discretionary access controls that limit access to administrators and only to those applications that require access. Such shared secret files shall be encrypted so that:
  1. The encryption key for the shared secret file is encrypted under a key held in a FIPS 140-2 Level 2 or higher validated hardware cryptographic module or any FIPS 140-2 Level 3 or 4 cryptographic module and decrypted only as immediately required for an authentication operation.
  2. Shared secrets are protected as a key within the boundary of a FIPS 140-2 Level 2 or higher validated hardware cryptographic module or any FIPS 140-2 Level 3 or 4 cryptographic module and is not exported in plaintext from the module.

Strongly bound credentials support tamper detection mechanisms such as digital signatures, but weakly bound credentials can be protected against tampering using access control mechanisms as described above.

- *Token and credential verification services* – CSPs shall provide a secure mechanism to allow Verifiers or Relying Parties to ensure that the credentials are valid. Such mechanisms may include on-line validation servers or the involvement of CSP servers that have access to status records in authentication transactions.

  Temporary session authentication keys may be generated from long-term shared secret keys by CSPs and distributed to third party Verifiers, as a part of the verification services offered by the CSP, but long-term shared secrets shall not be shared with any third parties, including third party Verifiers. This type of third-party (or delegated) verification is used in the realm of GSM (Global System for Mobile Communications) roaming; the locally available network authenticates the "roaming" Subscriber using a temporary session authentication key received from the Base Station. Such temporary session authentication keys are typically created by cryptographically combining the long term shared secret with a nonce challenge, to generate a session key. The challenge and session key are securely transmitted to the Verifier. The Verifier in turn sends only the challenge to the Claimant, and the Claimant applies the challenge to the long-term shared secret to generate the session key. Both Claimant and Verifier now share a session key, which can be used for authentication. Such verification schemes are permitted at this level provided that approved cryptographic algorithms are used for all operations.

  Token and credential verification services categorized as FIPS 199 "Moderate" or "High" for availability shall be protected in accordance with the Contingency Planning (CP) controls specified in NIST SP 800-53 to provide an adequate level of availability needed for the service.

- *Token and credential renewal /re-issuance* – Renewal and re-issuance shall only occur prior to expiration of the current credential. Claimants shall authenticate to the CSP using the existing token and credential in order to renew or re-issue the credential. All interactions shall occur over a protected channel such as SSL/TLS.

- *Credential revocation and destruction* – CSPs shall have a procedure to revoke credentials and tokens within 24 hours. The certificate status provisions of CAs cross-certified with the Federal Bridge CA at the Basic, Medium, High or Common Certificate Policy levels are considered to meet credential status and revocation provisions of this level. Verifiers shall ensure that the tokens they rely upon are either freshly issued (within 24 hours) or still valid. Shared secret based authentication systems may simply remove revoked Subscribers from the verification database. Secondary credentials must have a lifetime less than 2 hours.

- *Records retention* – All stipulations from Level 2 apply.

#### *8.3.1.4 Level 4*

At Level 4, the following is required:

- *Credential storage* – No additional stipulation.
- *Token and credential verification services* – No additional stipulation.
- *Token and credential renewal/re-issuance* – Sensitive data transfers shall be cryptographically authenticated using keys bound to the authentication process. All temporary or short-term keys derived during the original authentication operation shall expire and re-authentication shall be required after not more than 24 hours from the initial authentication.
- *Token and credential revocation and destruction* – CSPs shall have a procedure to revoke credentials within 24 hours. Verifiers or Relying Parties shall ensure that the credentials they rely upon are either freshly issued (within 24 hours) or still valid. The certificate status provisions of CAs cross-certified with the Federal Bridge CA at the High and Common Certificate Policies shall be considered to meet credential status provisions of Level 4. [FBCA1]

  It is generally good practice to destroy a token within 48 hours of the end of its life or the end of the Subscriber's association with the CSP. Destroying includes either the physical destruction of the token or cleansing it of all information related to the Subscriber.

  Secondary credentials must have a lifetime less than 2 minutes.
- *Records retention* – All stipulations from Levels 2 and 3 apply. The minimum record retention period for Level 4 credential data is ten years and six months beyond the expiration or revocation of the credential.

### *8.3.2 Relationship of PKI Policies to E-Authentication Assurance Levels*

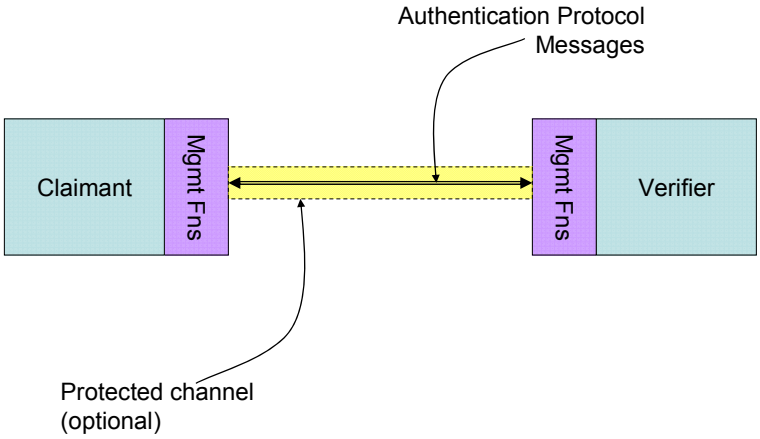
Appendix B specifies the mapping between the Federal PKI Certificate Policies and the requirements in Section 8.

# 9 Authentication Process

## *9.1 Overview*

The authentication process establishes the identity of the Claimant to the Verifier with a certain degree of assurance. It is implemented through an authentication protocol message exchange, as well as management mechanisms at each end that further constrain or secure the authentication activity. One or more of the messages of the authentication protocol may need to be carried on a protected channel. This is illustrated in the figure below.

Figure 4 - Authentication Process Model




*An authentication protocol is a defined sequence of messages between a Claimant and a Verifier that demonstrates that the Claimant has control of a valid token to establish his or her identity, and optionally, demonstrates to the Claimant that he or she is communicating with the intended Verifier*. An exchange of messages between a Claimant and a Verifier that results in authentication (or authentication failure) between the two parties is an authentication protocol run. During or after a successful authentication protocol run, a protected communication channel may be created between the two parties; this protected channel may be used to exchange the remaining messages of the authentication protocol run, or to exchange session data between the two parties.

Management mechanisms and functions may be implemented on the Claimant and the Verifier to further enhance the authentication process. For example, trust anchors may be established at the Claimant to enable the authentication of the Verifier using public key mechanisms such as TLS. Similarly, account lockout mechanisms may be implemented on the Verifier to prevent online guessing of passwords by an Attacker who is trying to authenticate as a legitimate Claimant.

### 9.2 Authentication Process Threats

In general, attacks that reveal long-term token secrets are worse than attacks that reveal short-term authentication secrets or session data, because in the former, the Attacker can then use the token secret to assume a Subscriber's identity and do greater harm.

Registration Authorities, CSPs, Verifiers and Relying Parties are ordinarily trustworthy (in the sense of being correctly implemented and not deliberately malicious). However, Claimants or their systems may not be trustworthy (or else their identity claims could simply be trusted). Moreover, while RAs, CSPs, Verifiers and Relying Parties are normally trustworthy, they are not invulnerable, and could become corrupted. Therefore, authentication protocols that expose long-term authentication secrets more than is absolutely required, even to trusted entities, should be avoided. The table below lists the types of threats posed to the authentication process.

Table 10 - Authentication Process Threats

| Type of Attack | Description | Example |
|---|---|---|
| Online guessing | An Attacker performs repeated logon trials by guessing possible values of the token authenticator. | An Attacker navigates to a web page and attempts to log in using a Subscriber's username and commonly used passwords, such as "password" and "secret". |
| Phishing | A Subscriber is lured to interact with a counterfeit Verifier, and tricked into revealing his or her token secret, sensitive personal data or authenticator values that can be used to masquerade as the Subscriber to the Verifier. | A Subscriber is sent an email that redirects him or her to a fraudulent website and is asked to log in using his or her username and password. |
| Pharming | A Subscriber who is attempting to connect to a legitimate Verifier, is routed to an Attacker's website through manipulation of the domain name service or routing tables. | A Subscriber is directed to a counterfeit website through DNS poisoning, and reveals or uses his or her token believing they are interacting with the legitimate Verifier. |
| Eavesdropping | An Attacker listens passively to the authentication protocol to capture information which can be used in a subsequent active attack to masquerade as the Claimant. | An Attacker captures the transmission of a password or password hash from a Claimant to a Verifier. |
| Replay | An Attacker is able to replay previously captured messages (between a legitimate Claimant and a Verifier) to authenticate as that Claimant to the Verifier. | An Attacker captures a Claimant's password or password hash from an actual authentication session, and replays it to the Verifier to gain access at a later time. |

| Type of Attack | Description | Example |
|---|---|---|
| Session hijack | An Attacker is able to insert himself or herself between a Subscriber and a Verifier subsequent to a successful authentication exchange between the latter two parties. The Attacker is able to pose as a Subscriber to the Verifier/Relying Party or vice versa to control session data exchange. | An Attacker is able to take over an already authenticated session by eavesdropping on or predicting the value of authentication cookies used to mark HTTP requests sent by the Subscriber. |
| Man-in-the-middle | The Attacker positions himself or herself in between the Claimant and Verifier so that he or she can intercept and alter the content of the authentication protocol messages. The Attacker typically impersonates the Verifier to the Claimant and simultaneously impersonates the Claimant to the Verifier. Conducting an active exchange with both parties simultaneously may allow the attacker to use authentication messages sent by one legitimate party to successfully authenticate to the other. | An Attacker breaks into a router that forwards messages between the Verifier and a Claimant. When forwarding messages, the attacker substitutes his or her own public key for that of the Verifier. The Claimant is tricked into encrypting his or her password so that the Attacker can decrypt it.<br>An Attacker sets up a fraudulent website impersonating the Verifier. When an unwary Claimant tries to log in using his or her one time password device, the Attacker's website simultaneously uses the Claimant's one time password to log in to the real Verifier. |

### *9.2.1 Other Threats*

Attacks are not limited to the authentication protocol itself. Other attacks include:

- Flooding attacks in which the Attacker overwhelms the Verifier by flooding it with a large amount of traffic over the authentication protocol;
- Malicious code attacks that may compromise authentication tokens;
- Attacks that fool Claimants into using an insecure protocol, when the Claimant thinks that he or she is using a secure protocol, or trick the Claimant into overriding security controls (for example, by accepting server certificates that cannot be validated).

The purpose of flooding attacks is to overwhelm the resources used to support an authentication protocol to the point where legitimate Claimants cannot reach the Verifier or to slow down the process to make it more difficult for the Claimant to reach the Verifier. For example, a Verifier that implements an authentication protocol that uses encryption/decryption is sent a large number of protocol messages causing the Verifier to be crippled due to the use of excessive system resources to encrypt/decrypt. Nearly all authentication protocols are susceptible to flooding attacks; possible ways to resist such attacks is through the use of distributed Verifier architectures, use of load balancing

techniques to distribute protocol requests to multiple mirrored Verifier systems, or other similar techniques.

Malicious code could be introduced into the Claimant's computer system for the purpose of compromising the Claimant's authentication token. The malicious code may be introduced by many means, including the threats detailed below. There are many countermeasures (e.g. virus checkers and firewalls) that can mitigate the risk of malicious code on Claimant systems. General good practice to mitigate malicious code threats is outside the scope of this document[9]. Hardware tokens prevent malicious software from extracting and copying the authentication secret token from the token. However, malicious code may still misuse the token, particularly if activation data is presented to the token via the computer. Similarly, the cryptographic tokens at least make it difficult to trick a user into verbally giving away his or her authentication secret, making social engineering more difficult, since many kinds of passwords can be readily expressed over the telephone.

### *9.2.2 Threat Mitigation Strategies*

The following are strategies to mitigate the attacks listed in the previous section:

- *Online guessing resistance* – An authentication process is resistant to online guessing attacks if it is impractical for the Attacker, with no a priori knowledge of the token authenticator, to authenticate successfully by repeated authentication attempts with guessed authenticators. The entropy of the authenticator, the nature of the authentication protocol messages, and other management mechanisms at the Verifier contribute to this property. For example, password authentication systems can make targeted password guessing impractical by requiring use of high-entropy passwords (see Appendix A) and limiting the number of unsuccessful authentication attempts, or by controlling the rate at which attempts can be carried out. Similarly, to resist untargeted password attacks, a Verifier may supplement these controls with network security controls.
- *Phishing and pharming resistance (verifier impersonation)* – An authentication process is resistant to phishing and pharming (also known as Verifier impersonation,) if the impersonator does not learn the value of token secret or a token authenticator that can be used to act as a Subscriber to the genuine Verifier. In the most general sense, this assurance can be provided by the same mechanisms that provide the strong man-in-the-middle resistance described later in this section; however, long term secrets can be protected against phishing and pharming simply by the use of a tamper resistant token, provided that the long term secret cannot be reconstructed from a Token Authenticator. To decrease the likelihood of phishing and pharming attacks, it is recommended that the Claimant authenticate the Verifier using cryptographic mechanisms prior to submitting the token authenticator to the

[9] See SP 800-53, *Recommended Security Controls For Federal Information Systems*

supposed Verifier. Additionally, management mechanisms can be implemented at the Verifier to send a Claimant personalized content after successful authentication of the Claimant or the Claimant's device (refer to Section 9.2.3 for further details on personalization). This allows the Claimant to achieve a higher degree of assurance of the authenticity of the Verifier before proceeding with the remainder of the session with the Verifier or Relying Party. It should be mentioned, however, that there is no foolproof way to prevent the Claimant from revealing any sensitive information to which he or she has access.

- *Eavesdropping resistance* – An authentication process is resistant to eavesdropping attacks if an eavesdropper who records all the messages passing between a Claimant and a Verifier finds it impractical to learn the Claimant's token secret or to otherwise obtain information that would allow the eavesdropper to impersonate the Subscriber in a future authentication session. Eavesdropping resistant protocols make it impractical[10] for an Attacker to carry out an off-line attack where he or she records an authentication protocol run and then analyses it on his or her own system for an extended period to determine the token secret or possible token authenticators. For example, an Attacker who captures the messages of a password-based authentication protocol run may try to crack the password by systematically trying every password in a large dictionary, and comparing it with the protocol run data.
- *Replay resistance* – An authentication process resists replay attacks if it is impractical to achieve a successful authentication by recording and replaying a previous authentication message. Protocols that use nonces or challenges to prove the "liveness" of the transaction are resistant to replay attacks since the Verifier will easily detect that the old protocol messages replayed do not contain the appropriate nonces or timeliness data related to the current authentication session.
- *Hijacking resistance* – An authentication process and data transfer protocol combination are resistant to hijacking if the authentication is bound to the data transfer in a manner that prevents an adversary from participating actively in the data transfer session between the Subscriber and the Verifier or Relying Party without being detected. This is a property of the relationship of the authentication protocol and the subsequent session protocol used to transfer data. This binding is usually accomplished by generating a per-session shared secret during the authentication process that is subsequently used by the Subscriber and the Verifier or Relying Party to authenticate the transfer of all session data.

[10] "Impractical" is used here in the cryptographic sense of nearly impossible, that is there is always a small chance of success, but even the Attacker with vast resources will nearly always fail. For off-line attacks, impractical means that the amount of work required to "break" the protocol is at least on the order of $2^{80}$ cryptographic operations. For on-line attacks impractical means that the number of possible on-line trials is very small compared to the number of possible key or password values.

It is important to note that web applications, even those protected by SSL/TLS, can still be vulnerable to a type of session hijacking attack called Cross Site Request Forgery (CSRF.) In this type of attack, a malicious website contains a link to the URL of the legitimate Relying Party. The malicious website is generally constructed so that a web browser will automatically send an HTTP request to the Relying Party whenever the browser visits the malicious website. If the Subscriber visits the malicious website while he or she has an open SSL/TLS session with the Relying Party, the request will generally be sent in the same session and with any authentication cookies intact. While the attacker never gains access to the session secret, the request may be constructed to have side effects, such as sending an email message or authorizing a large transfer of money.

CSRF attacks may be prevented by making sure that neither an attacker nor a script running on the attacker's website has sufficient information to construct a valid request authorizing an action (with significant consequences) by the Relying Party. This can be done by inserting random data, supplied by the Relying Party, into any linked URL with side effects and into a hidden field within any form on the Relying Party's website. This mechanism, however, is not effective if the attacker can run scripts on the Relying Party's website (Cross Site Scripting or XSS.) To prevent XSS vulnerabilities, the Relying Party should sanitize inputs from Claimants or Subscribers to make sure they are not executable, or at the very least not malicious, before displaying them as content to the Subscriber's browser.

- *Man-in-the-middle resistance* – Authentication protocols are resistant to a man-in-the-middle attack when both parties (e.g., Claimant and Verifier) are authenticated to the other in a manner that prevents the undetected participation of a third party. There are two levels of resistance:

  1. *Weak man-in-the-middle resistance* – A protocol is said to be weakly resistant to man-in-the-middle attacks if it provides a mechanism for the Claimant to determine whether he or she is interacting with the real Verifier, but still leaves the opportunity for the non-vigilant Claimant to reveal a token authenticator (to an unauthorized party) that can be used to masquerade as the Claimant to the real Verifier. For example, sending a password over server authenticated TLS is weakly resistant to man-in the middle attacks. The browser allows the Claimant to verify the identity of the Verifier; however, if the Claimant is not sufficiently vigilant, the password will be revealed to an unauthorized party who can abuse the information. Weak man-in-the-middle resistance can also be provided by a zero-knowledge password protocol, such as Encrypted Key Exchange (EKE), Simple Password Exponential Key Exchange (SPEKE), or Secure Remote Password Protocol (SRP), which enables the Claimant to authenticate to a Verifier without disclosing the token secret. However, it is possible for the attacker to trick the Claimant into passing his or her

password into a less secure protocol, thereby revealing the password to the attacker. Furthermore, if it is unreasonably difficult for the Claimant to verify that the proper protocol is being used, then the overall authentication process does not even provide weak MitM resistance (for example, if a zero-knowledge password protocol is implemented by an unsigned java applet displayed on a plaintext HTTP page).

2. *Strong man-in-the-middle resistance*: A protocol is said to be strongly resistant to man-in-the-middle attack if it does not allow the Claimant to reveal, to an attacker masquerading as the Verifier, information (token secrets, authenticators) that can be used by the latter to masquerade as the true Claimant to the real Verifier. An example of such a protocol is client authenticated TLS, where the browser and the web server authenticate one another using PKI credentials. Even an unwary Claimant cannot easily reveal to an attacker masquerading as the Verifier any information that can be used by the attacker to authenticate to the real Verifier. Specialized protocols where the Claimant's token device will only release an authenticator to a preset list of valid Verifiers may also be strongly resistant to man-in-the-middle attacks.

### *9.2.3 Phishing and Pharming (Verifier Impersonation)*

It is important to note that phishing and pharming are attacks that use different techniques to achieve the same goal. Effectively, the Claimant is tricked into believing that he or she is interacting with the Verifier when in actuality, the Verifier is being impersonated by an Attacker attempting to collect token information or other sensitive information.

In a successful phishing attack, the Attacker sends an official looking email to a Subscriber claiming to be a Verifier. The email usually contains a link to a counterfeit Verifier and will ask the Subscriber to click on the link and authenticate to the Verifier[11]. The Subscriber proceeds to authenticate to the counterfeit Verifier and the login information and token authenticator is captured. At this point, the Subscriber is unaware that he or she has been phished, and proceeds with the actions requested by the original email. Once the Subscriber logs off, he or she is unaware that his or her login information has been captured, and that potentially, sensitive data has been captured.

In a successful pharming attack, the Attacker corrupts either the domain name service (using a technique called DNS poisoning) or the local routing tables (by modifying the host files on a Claimant's computer to point to bogus DNS server). When the Subscriber attempts to connect to a legitimate Verifier on the Internet, the corrupted DNS tables or routing tables take the Subscriber to a counterfeit Verifier on the Internet. The Subscriber unknowingly reveals token authenticators and other sensitive information to the counterfeit Verifier.

[11] Some phishing attacks may request the Subscriber to provide personally sensitive information so that the Attacker may impersonate the Subscriber outside the scope of E-authentication.

The strongest mechanism for preventing phishing and pharming of authentication secrets, such as token authenticators, is to make sure that some authentication secrets are not directly accessible to the Claimant (as described in Section 9.2.2). However, to help mitigate a wider variety of phishing and pharming attacks, the following techniques may be used:

- *Out of band confirmation of transaction details* – Details (e.g. account number, amount etc.) of sensitive transactions authorized by the Subscriber may be sent by the Relying Party to the Subscriber's out of band token and displayed along with a confirmation code. The confirmation code may either be cryptographically derived from the Subscriber's token secret and the transaction details, or it may be a random value that is sent to the Subscriber's out of band token along with the transaction details. Alternatively, transaction details may be typed in by the Subscriber as manual inputs to a one time password device. In order to complete the transaction, the Subscriber must send the correct one time password or confirmation code to the Verifier or Relying Party.
- *Adding a "Last Login" feature by the Verifier to inform the Subscriber of his or her last login* – If the Subscriber logged in at 8:00am and then logs in at 4:00pm but the Last Login feature states that the last login was at 2:00pm, the Subscriber may suspect that he or she has been phished and take appropriate action.

Personalization is the process of customizing a webpage or email for a user to enhance the user experience. For the purpose of this document, personalization schemes can assist the user to determine if he or she is interacting with the correct entity. It is important to note that personalization is at best a low assurance mechanism for mitigating Phishing and Pharming threats, especially when delivered over a communication protocol that is not strongly resistant to man-in-the-middle attacks. However, personalization may provide additional assurance when combined with other techniques.

There are three types of personalization in the context of this guideline:

- *Pre-authentication personalization* – The Verifier displays to the Claimant some personalized indicator prior to the latter submitting the token authenticator to the former. This indicator may be established by the Subscriber at the time of registration. When the Claimant views the personalized indicator, the Claimant has an increased sense of assurance that he or she is interacting with the correct Verifier. For example, a Verifier may require the Claimant to submit the username first; in response, the Verifier provides the personalized indicator for the claimed username. If the Claimant recognizes the personalized indicator as his or her own, the Claimant submits his or her token authenticator to the Verifier. Pre-authentication personalization does not eliminate Phishing attacks, but requires the Attacker to use a more complex technique to succeed in a Phishing attack.

- *Post authentication personalization* – The Verifier displays a personalized indicator to the Subscriber after successful authentication of the latter. The personalized indicator provides assurance to the Subscriber that he or she has in fact logged in to the correct site. This indicator may be established by the Subscriber at the time of registration. For example, after a Subscriber authenticates to the Verifier, the Verifier provides a personalized indicator (such as a picture, a phrase, or a greeting) that the Subscriber can readily recognize as his or her own. If the personalized indicator is not shown, or is not recognized by the Subscriber, the Subscriber suspects that he or she has been phished and takes appropriate action. The advantage to using post authentication personalization is that it assists the Subscriber in recognizing that he or she is interacting with a bogus Verifier; the Subscriber can then refrain from revealing any sensitive information. It should be noted that the most straightforward implementation of personalization is susceptible to a fairly easy man-in-the-middle attack where the phishing website is set up to authenticate to the real Verifier simultaneously as it captures information from the real Subscriber.
- *Personalization of email sent to the Subscriber by a valid Verifier* – This type of personalization is employed to help the Subscriber differentiate between email from a valid Verifier, and email from a Phisher. For example, an email from a Verifier may contain a picture which the Subscriber selected in the registration process. This type of personalization forces the Phisher to use a fairly difficult attack and in effect forces the Phisher to either use a targeted attack against each Subscriber or hope that the Subscriber will not notice the incorrect or missing personalization identifier.

  It is important to note that using a Subscriber's name (first or last) as the only method of personalization is a relatively weak method to thwart a phishing attack since it is fairly easy for an Attacker to gain this type of information and display it in an email or display it after logging into a site. Information of a non-public nature is a better candidate for use during personalization.

## 9.3 Authentication Process Assurance Levels

The stipulations for authentication process assurance levels are described in the following sections.

### 9.3.1 Threat Resistance per Assurance Level

Authentication process assurance levels can be defined in terms of threat resistance. The table below lists the threat resistance requirements per assurance level:

Table 11 – Authentication Protocol Threat Resistance per Assurance Level

| Authentication Process Attacks/Threats | Threat Resistance Requirements | | | |
|---|---|---|---|---|
| | Level 1 | Level 2 | Level 3 | Level 4 |
| Online guessing | Yes | Yes | Yes | Yes |
| Replay | Yes | Yes | Yes | Yes |

| Authentication Process Attacks/Threats | Threat Resistance Requirements | | | |
|---|---|---|---|---|
| | Level 1 | Level 2 | Level 3 | Level 4 |
| Session hijacking | No | Yes | Yes | Yes |
| Eavesdropping | No | Yes | Yes | Yes |
| Phishing/pharming(verifier impersonation) | No | No | Yes[12] | Yes |
| Man in the middle | No | Weak | Weak | Strong |
| Denial of service/flooding[13] | No | No | No | No |

### 9.3.2 Additional Requirements per Assurance Level

This section covers some additional requirements levied on the authentication process at each assurance level. At Levels 2 and above, the authentication process shall provide sufficient information to the Verifier to uniquely identify the appropriate registration information that was (i) provided by the Subscriber at the time of registration, and (ii) verified by the RA in the issuance of the token and credential. It is important to note that the requirements listed below will not protect the authentication process if malicious code is introduced on the Claimant's machine or at the Verifier.

#### *9.3.2.1 Level 1*

Although there is no identity proofing requirement at this level, the authentication mechanism provides some assurance that the same Claimant is accessing the protected transaction or data. It allows a wide range of available authentication technologies to be employed and permits the use of any token methods of Levels 2, 3 or 4. Successful authentication requires that the Claimant shall prove, through a secure authentication protocol, that he or she controls the token.

Plaintext passwords or secrets shall not be transmitted across a network at Level 1. However this level does not require cryptographic methods that block offline analysis by eavesdroppers. For example, password challenge-response protocols that combine a password with a challenge to generate an authentication reply satisfy this requirement although an eavesdropper who intercepts the challenge and reply may be able to conduct a successful off-line dictionary or password exhaustion attack and recover the password. Common protocols that meet Level 1 requirements include APOP [RFC 1939], S/KEY [SKEY], and Kerberos [KERB]. Since an eavesdropper who intercepts such a protocol exchange will often be able to find the password with a straightforward dictionary attack, and this vulnerability is independent of the strength of the operations, there is no requirement at this level to use Approved cryptographic techniques. Level 1, long-term shared authentication secrets may be revealed to Verifiers.

A wide variety of technologies should be able to meet the requirements of Level 1. For example, a Verifier might obtain a Subscriber password from a CSP and authenticate the

[12] Long term authentication secrets shall be protected at this level. Short term secrets may or may not be protected.
[13] Although there are techniques used to resist flood attacks, no protocol has comprehensive resistance to stop flooding.

Claimant by use of a challenge-response protocol. A password sent through a TLS protocol session is another example.

#### *9.3.2.2 Level 2*

Level 2 allows a wide range of available authentication technologies to be employed and permits the use of any of the token methods of Levels 2, 3 and 4. Successful authentication requires that the Claimant shall prove, through a secure authentication protocol, that he or she controls the token. Session hijacking (when required based on the FIPS 199 security category of the systems as described below), replay, and on-line guessing attacks shall be resisted. Approved cryptography is required to resist eavesdropping to capture authentication data. Protocols used at Level 2 and above shall be at least weakly MitM resistant, as described in the threat mitigation strategies subsection.

Session data transmitted between the Claimant and the Relying Party following a successful Level 2 authentication must be protected as described in the NIST FISMA guidelines. Specifically, all session data exchanged between information systems that are categorized as FIPS 199 "Moderate" or "High" for confidentiality and integrity, shall be protected in accordance with NIST SP 800-53 Control SC-8 (which requires transmission confidentiality) and SC-9 (which requires transmission integrity).

A wide variety of technologies can meet the requirements of Level 2. For example, a Verifier might authenticate a Claimant who provides a password through a secure (encrypted) TLS protocol session (tunneling).

#### *9.3.2.3 Level 3*

Refer to Section 7 for single tokens and token combinations that are allowed to be used to achieve Level 3 authentication assurance. Additionally, At Level 3, strong cryptographic mechanisms shall be used to protect token secret(s) and authenticator(s). Long-term shared authentication secrets, if used, shall never be revealed to any party except the Claimant and CSP; however, session (temporary) shared secrets may be provided to Verifiers by the CSP. Approved cryptographic techniques shall be used for all operations including the transfer of session data.

Level 3 assurance may be satisfied by client authenticated TLS (implemented in all modern browsers), with Claimants who have public key certificates. Other protocols with similar properties may also be used.

Level 3 authentication assurance may also be met by tunneling the output of a MF OTP Token, or the output of a SF OTP Token in combination with a Level 2 personal password, through a TLS session.

#### *9.3.2.4 Level 4*

Level 4 is intended to provide the highest practical remote network authentication assurance. Refer to Section 7 for single tokens and token combinations that are allowed to be used to achieve Level 4 authentication assurance.

Level 4 requires strong cryptographic authentication of all parties, and all sensitive data transfers between the parties. Either public key or symmetric key technology may be used. The token secret shall be protected from compromise through the malicious code threat as described in Section 9.2.1 above. Long-term shared authentication secrets, if used, shall never be revealed to any party except the Claimant and CSP; however session (temporary) shared secrets may be provided to Verifiers or Relying Parties by the CSP. Strong, Approved cryptographic techniques shall be used for all operations including the transfer of session data. All sensitive data transfers shall be cryptographically authenticated using keys that are derived from the authentication process in such a way that MitM attacks are strongly resisted.

Level 4 assurance may be satisfied by client authenticated TLS (implemented in all modern browsers), with Claimants who have public key MF Hardware Cryptographic Tokens. Other protocols with similar properties can also be used.

# 10 Assertions

## *10.1 Overview*

Assertions are statements from a Verifier to a Relying Party that contain information about a Subscriber. Assertions are used when the Relying Party and the Verifier are not collocated (e.g., they are connected through a shared network). The Relying Party uses the information in the assertion to identify the Claimant and make authorization decisions about his or her access to resources controlled by the Relying Party. An assertion may include identification and authentication statements regarding the Subscriber, and may additionally include attribute statements that further characterize the Subscriber and support the authorization decision at the Relying Party.

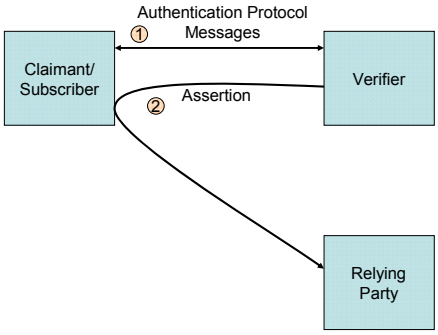
Assertion based authentication of the Claimant serves several important goals. It supports the process of Single-Sign-On for Claimants, allowing them to authenticate once to a Verifier and subsequently obtain services from multiple Relying Parties without further authentication. Assertion mechanisms also support the implementation of a federated identity for a Subscriber, allowing the linkage of multiple identities/accounts held by the Subscriber with different Relying Parties through the use of a common "federated" identifier. In this context, a federation is a group of entities (Relying Parties, Verifiers and CSPs) that are bound together through common, agreed upon, business practices, policies, trust mechanisms, profiles and protocols. Finally, assertion mechanisms can also facilitate authentication schemes that are based on the attributes or characteristics of the Claimant in lieu of (or in addition to) the identity of the Claimant.

It is important to note that assertion schemes are fairly complex multiparty protocols, and therefore have fairly subtle security assumptions which must be satisfied. When evaluating a particular assertion scheme, it may be instructive to break it down into its component interactions. Generally speaking, interactions between the Claimant/Subscriber and the Verifier and between the Claimant/Subscriber and Relying Party are similar to the authentication mechanisms presented in Section 9, while interactions between the Verifier and Relying Party are similar to the token and credential verification services presented in Section 8. Many of the requirements presented in this section will, therefore, be similar to corresponding requirements in those two sections.

There are two basic assertion models. After successful authentication with the Verifier, the Subscriber is issued an assertion or an assertion reference, which the Subscriber uses to authenticate to the Relying Party.

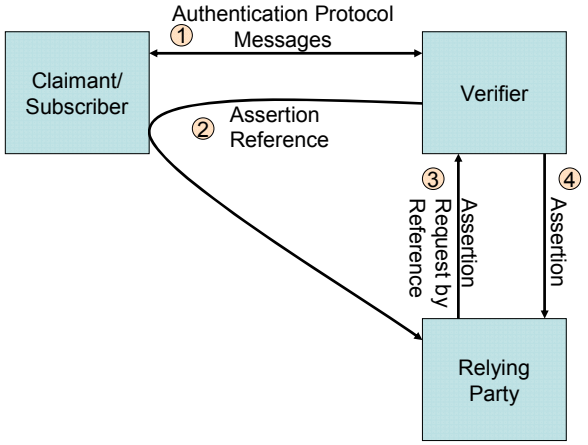
- *The Direct Model* – In the direct model, the Claimant uses his or her E-authentication token to authenticate to the Verifier. Following successful authentication of the Claimant, the Verifier creates an assertion, and sends it to the Subscriber to be forwarded to the Relying Party. The assertion is used by the Claimant/Subscriber to authenticate to the Relying Party. The figure below illustrates this model.

Figure 5 - Direct Assertion Model




- *The Indirect Model* – In the indirect model, the Claimant uses his or her token to authenticate to the Verifier. Following successful authentication, the Verifier creates an assertion as well as an assertion reference (which identifies the Verifier and includes a pointer to the full assertion held by the Verifier). The assertion reference is sent to the Subscriber to be forwarded to the Relying Party. In this model, the assertion reference is used by the Claimant/Subscriber to authenticate to the Relying Party. The Relying Party then uses the assertion reference to explicitly request the assertion from the Verifier. The figure below illustrates this model.

Figure 6 - Indirect Assertion Model




As mentioned earlier, an assertion contains a set of claims or statements about an authenticated Subscriber. Based on the statements contained within it, an assertion can be of one of two types:

- *Holder-of-Key Assertion* – A holder-of-key assertion contains a reference to a symmetric key or a public key (corresponding to a private key) possessed by the Subscriber. The Relying Party may require the Subscriber to prove possession of the secret that is referenced in the assertion. In proving possession of the Subscriber's secret, the Subscriber also proves that he or she

is the rightful owner of the assertion. It is therefore difficult for an Attacker to use a holder-of-key assertion issued to another Subscriber, since the former cannot prove possession of the secret referenced within the assertion.

- *Bearer Assertions* – A bearer assertion does not provide a mechanism for the Subscriber to prove that he or she is the rightful owner of the assertion. The Relying Party has to assume that the assertion was issued to the Subscriber who presents the assertion or the corresponding assertion reference to the Relying Party. If a bearer assertion (in the direct model) or assertion reference (in the indirect model) belonging to a Subscriber is captured, copied, or manufactured by an Attacker, the latter can impersonate the rightful Subscriber to obtain services from the Relying Party. Bearer assertions can be made secure only if some part of the assertion or assertion reference, sent to the Subscriber by the Verifier, is unpredictable to an Attacker and can reliably be kept secret.

There are cases in which the Relying Party should be anonymous to the Verifier for the purpose of privacy. The direct model is more suitable for the "anonymous Relying Party" scenario since there is no requirement for the Relying Party to authenticate to the Verifier as in the indirect model. However, it is possible to devise authentication schemes (e.g. using key hierarchies within a group or federation) that allow the use of the indirect model to support the "anonymous Relying Party" scenario.

There are other cases where privacy concerns require that the Claimant's identity/account at the Verifier and Relying Party not be linked through use of a common identifier/account name. In such scenarios, pseudonymous identifiers are used within the assertions generated by the Verifier for the Relying Party.

It should be noted that the two models described above are abstractions. There may be other interactions between the three players preceding or interspersed with the interactions described in the model. For example, the Claimant may initiate a connection with a Relying Party of his or her choice, at which point, the latter would redirect the Claimant to an appropriate Verifier to be authenticated using the direct model, resulting in an assertion being sent to the Relying Party. Alternately, the Claimant may first authenticate to a Verifier of his or her choice and then select one or more Relying Parties to obtain further services. The direct model is used to generate assertions for each of these Relying Parties. Parallel scenarios may be constructed for the indirect model as well.

Three types of assertion technologies will be discussed within this section: Web browser cookies, SAML (Security Assertions Markup Language) assertions, and Kerberos tickets. Other assertion technologies may be used in an E-authentication environment as long as they meet the requirements set forth in Section 10.3 below for the targeted assurance level.

### *10.1.1 Cookies*

One type of assertion widely in use is Internet cookie technology. Cookies are text files used by a browser to store information provided by a particular web site. The contents of the cookie are sent back to the web site each time the browser requests a page from the same web site. The web site uses the contents of the cookie to identify the user and prepare customized Web pages for that user, or to authorize the user for certain transactions.

Cookies have two mandatory parameters:

- *Name* – This parameter states the name of the cookie.
- *Value* – This parameter holds information that a cookie is storing. For example, the value parameter could hold a user ID or session ID.

Cookies also have four optional parameters:

- *Expiration date* – This parameter determines how long the cookie stays valid.
- *Path* – This parameter sets the path over which the cookie is valid.
- *Domain* – This parameter determines the domain in which cookie is valid.
- *Secure* – This parameter indicates the cookie requires that a secure connection exist for the cookie to be used.

There are two types of cookies:

- *Session cookies* – A cookie that is erased when the user closes the web browser. The session cookie is stored in temporary memory and is not retained after the browser is closed.
- *Persistent cookies* – A cookie that is stored on a user's hard drive until it expires (persistent cookies are set with expiration dates) or until the user deletes the cookie.[14]

Cookies are effective as assertions for Internet single sign on where the Relying Party and Verifier are part of the same Internet domain, and when the cookie contains authentication status for that domain. They are not usable in scenarios where the Relying Party and the Verifier are part of disparate domains.

Cookies are also often used by the Claimant to re-authenticate to a server after the communication channel between them has been closed. This may be considered to be a use of assertion technology. In this case, the server acts as a Verifier when it sets the cookie in the Subscriber's browser, and as a Relying Party when it requests the cookie from a Claimant who wishes to re-authenticate to it. Note that, if the cookie is used as an assertion reference in this way, no assertion needs to be sent on an open network, and

[14] Per OMB M-03-22 OMB Guidance for Implementing the Privacy Provisions of the E-Government Act of 2002, Federal agencies are prohibited from using persistent cookie technology for their websites.

therefore, confidentiality and integrity requirements for assertion data at Level 2 and below may be satisfied by discretionary access controls rather than by cryptographic methods. This is in line with the credential storage requirement presented in Section 8.

### *10.1.2 Security Assertions Markup Language (SAML)*

SAML is an XML-based framework for creating and exchanging authentication and attribute information between trusted entities over the Internet. As of this writing, the latest specification for SAML is SAML v2.0, issued 15 March 2005.

The building blocks of SAML include the Assertions XML schema which define the structure of the assertion; the SAML Protocols which are used to request assertions and artifacts (that is, the assertion reference mentioned in Section 10.1); and the Bindings that define the underlying communication protocols (such as HTTP or SOAP) and that can be used to transport the SAML assertions. The three components above define a SAML profile that corresponds to a particular use case such as "Web Browser SSO".

SAML Assertions are encoded in an XML schema and can carry up to three types of statements:

- *Authentication statements* – Include information about the assertion issuer, the authenticated subject, validity period, and other authentication information. For example, an Authentication Assertion would state the subject "John" was authenticated using a password at 10:32pm on 06-06-2004
- *Attribute statements* – Contain specific additional characteristics related to the Subscriber. For example, subject "John" is associated with attribute "Role" with value "Manager".
- *Authorization statements* – Identify the resources the Subscriber has permission to access. These resources may include specific devices, files, information on specific web servers, etc. For example, subject "John" for action "Read" on "Webserver1002" given evidence "Role".

Authorization statements will not be discussed in this document as this topic is beyond the scope of this document.

### *10.1.3 Kerberos Tickets*

The Kerberos Network Authentication Service [RFC4120] was designed to provide strong authentication for client/server applications using symmetric-key cryptography. Extensions to Kerberos can support the use of public key cryptography for selected steps of the protocol. Kerberos also supports confidentiality and integrity protection of session data between the Subscriber and the Relying Party.

Kerberos supports authentication of a Claimant over an untrusted, shared network using two or more Verifiers. The Claimant implicitly authenticates to the Verifier by demonstrating the ability to decrypt a random session key encrypted for the Subscriber by the Verifier. (Some Kerberos variants also require the Subscriber to explicitly

authenticate to the Verifier, but this is not universal.) In addition to the encrypted session key, the Verifier also generates another encrypted object called a Kerberos ticket. The ticket contains the same session key, the identity of the Subscriber to whom the session key was issued, and an expiration time after which the session key is no longer valid. The ticket is confidentiality and integrity protected by a pre-established key shared between the Verifier and the Relying Party.

To authenticate using the session key, the Claimant sends the ticket to the Relying Party along with encrypted data that proves that the Claimant possesses the session key embedded within the Kerberos ticket. Session keys are either used to generate new tickets, or to encrypt and authenticate communications between the Subscriber and the Relying Party.

To begin the process, the Claimant sends an authentication request to the Authentication Server (AS.) The AS encrypts a session key for the Subscriber using the Subscriber's long term credential. The long term credential may either be a secret key shared between the AS and the Subscriber, or in the PKINIT variant of Kerberos, a public key certificate. The AS also issues a ticket using a key it shares with the Ticket Granting Server (TGS). This ticket is referred to as a Ticket Granting Ticket (TGT), since the verifier uses the session key in the TGT to issue tickets rather than to explicitly authenticate the Claimant. The TGS uses the session key in the TGT to encrypt a new session key for the Subscriber and uses a key it shares with the Relying Party to generate a ticket corresponding to the new session key. The subscriber decrypts the session key and uses the ticket and the new session key together to authenticate to the Relying Party.

### *10.2 Assertion Threats*

In this section, it is assumed that the two endpoints of the assertion transmission (namely, the Verifier and the Relying Party) are uncompromised. However, the Claimant is not assumed to be entirely trustworthy as the Claimant may have an interest in modifying or replacing an assertion to obtain a greater level of access to a resource/service provided by the Relying Party. Other Attackers are assumed to lurk within the shared transmission medium (e.g., Internet) that may be interested in obtaining or modifying assertions and assertion references to impersonate a Subscriber or access unauthorized data or services. Furthermore, it is possible that two or more entities may be colluding to attack another party. An Attacker may attempt to subvert assertion protocols by directly compromising the integrity or confidentiality of the assertion data. For the purpose of this type of threat, authorized parties who attempt to exceed their privileges may be considered Attackers.

- *Assertion manufacture/modification* – An Attacker may generate a bogus assertion or modify the assertion content (such as the authentication or attribute statements) of an existing assertion, causing the Relying Party to grant inappropriate access to the Subscriber. For example, an Attacker may modify the assertion to extend the validity period; a Subscriber may modify the assertion to have access to information that they should not be able to view.

- *Assertion disclosure* – Assertions may contain authentication and attribute statements that include sensitive Subscriber information. Disclosure of the assertion contents can make the Subscriber vulnerable to other types of attacks.
- *Assertion repudiation* – An assertion may be repudiated by a Verifier if the proper mechanisms are not in place. For example, if a Verifier does not digitally sign an assertion, the Verifier can claim that it was not generated through the services of the Verifier.
- *Assertion redirect*: An Attacker uses the assertion generated for one Relying Party to obtain access to a second Relying Party.
- *Assertion reuse* – An Attacker attempts to use an assertion that has already been used once with the intended Relying Party.

In addition to reliable and confidential transmission of assertion data from the Verifier to the RP, assertion protocols have a further goal: The Subscriber, once authenticated by the Verifier, must be issued some secret information, the knowledge of which distinguishes the Subscriber from attackers who wish to impersonate the Subscriber, and allows the Subscriber to authenticate to the Relying party. In the case of holder-of-key assertions, this secret is generally the Subscriber's long term token secret, but in other cases the Verifier will generate a temporary secret for this purpose. Since, when this secret is used to authenticate to the Relying Party, it generally replaces the token authenticator in the type of protocols described in Section 9, this temporary secret will be referred to here as a secondary authenticator. Secondary authenticators include assertions in the direct model, session keys in Kerberos, and assertion references in the indirect model. The threats to the secondary authenticator are as follows:

- *Secondary authenticator manufacture* – An Attacker may attempt to generate a valid secondary authenticator and use it to impersonate a Subscriber.
- *Secondary authenticator capture* – The Attacker may use a session hijacking attack to capture the secondary authenticator when the Verifier transmits it to the Subscriber after the primary authentication step, or the Attacker may use a man-in-the-middle attack to obtain the secondary authenticator as it is being used by the Subscriber to authenticate to the Relying Party. If, as in the indirect model, the RP needs to send the secondary authenticator back to the Verifier in order to check its validity or obtain the corresponding assertion data, an attacker may similarly subvert the communication protocol between the Verifier and the Relying Party to capture a secondary authenticator. In any of the above scenarios, the secondary authenticator can be used to impersonate the Subscriber.

Finally, in order for the Subscriber's authentication to the Relying Party to be useful, the binding between the secret used to authenticate to the RP and the assertion data referring to the Subscriber must be strong.

- *Assertion substitution* – A subscriber may attempt to impersonate a more privileged subscriber by subverting the communication channel between the

Verifier and Relying Party, for example by reordering the messages, to convince the Relying Party that his or her secondary authenticator corresponds to assertion data sent on behalf of the more privileged subscriber. This is primarily a threat to the indirect model, since in the direct model, assertion data is directly encoded in the secondary authenticator.

### *10.2.1 Threat Mitigation Strategies*

Mitigation techniques are described below for each of the threats described in the last subsection.

Logically speaking, an assertion is issued by a Verifier and consumed by a Relying Party – these are the two end points of the channel that needs to be secured to protect the assertion. In the direct model, the channel over which the assertion is passed traverses the Subscriber. Furthermore, in the current web environment, the assertion may pass through two separate secure channels (one between the Verifier and the Subscriber, and the other between the Subscriber and the Relying Party), with a break in channel security on the Subscriber's browser. This is reflected in the mitigation strategies described below. In the indirect model, the assertion flows directly from the Verifier to the Relying Party; this protocol channel needs to be protected. All of the threat mitigation strategies in Section 9 apply to the protocols used to request, retrieve and submit assertions and assertion references.

- *Assertion manufacture/modification*: To mitigate this threat, one of the following mechanisms may be used:

  1. The assertion may be digitally signed by the Verifier. The Relying Party should check the digital signature to verify that it was issued by a legitimate Verifier.

  2. The assertion may be sent over a protected channel such as TLS/SSL. In order to protect the integrity of assertions from malicious attack, the Verifier must be authenticated.

- *Assertion disclosure* – To mitigate this threat, one of the following mechanisms may be implemented:

  1. The assertion may be sent over a protected channel to an authenticated Relying Party. Note that, in order to protect assertions against both disclosure and manufacture/modification using a protected channel, both the Relying Party and the Verifier need to be authenticated.

  2. If assertions are signed by the Verifier, they may be encrypted for a specific Relying Party with no additional integrity protection. It should be noted that any protocol that requires a series of messages between two parties to be signed by their source and encrypted for their recipient provides all the same guarantees as a mutually authenticated protected channel, and may therefore be considered equivalent. The general requirement for protecting against both assertion disclosure and assertion

manufacture/modification may therefore be described as a mutually authenticated protected channel or equivalent between Verifier and Relying party.

- *Assertion repudiation* – To mitigate this threat, the assertion may be digitally signed by the Verifier using a key that supports non-repudiation. The Relying Party should check the digital signature to verify that it was issued by a legitimate Verifier.
- *Assertion redirect* – To mitigate this threat, the assertion may include the identity of the Relying Party for whom it was generated. The Relying Party verifies that incoming assertions include its identity as the recipient of the assertion.
- *Assertion reuse* – To mitigate this threat, the following mechanisms may be used:
  1. The assertion includes a timestamp and a short lifetime of validity. The Relying Party checks the timestamp and lifetime values to ensure that the assertion is currently valid.
  2. The Relying Party keeps track of assertions that were consumed within a (configurable) time window to ensure that an assertion cannot be used more than once within that time window.
- *Secondary authenticator manufacture* – To mitigate this threat, one of the following mechanisms may be implemented:
  1. The secondary authenticator may contain sufficient entropy that an Attacker without direct access to the Verifier's random number generator cannot guess the value of a valid secondary authenticator.
  2. The secondary authenticator may contain timely assertion data that is signed by the Verifier or integrity protected using a key shared between the Verifier and the Relying Party.
  3. The Subscriber may authenticate to the Relying Party directly using his or her long term token and avoid the need for a secondary authenticator altogether.
- *Secondary authenticator capture* – To mitigate this threat, adequate protections must be in place throughout the lifetime of any secondary authenticators used in the assertion protocol.
  1. In order to protect the secondary authenticator while it is in transit between the Verifier and the Subscriber, the secondary authenticator must be sent via a protected channel established during the primary authentication of the Subscriber using his or her token. This requirement is the same as the requirement, in the Authentication Process section, to

protect sensitive data (in this case the secondary authenticator) from session hijacking attacks.

2. In order to protect the secondary authenticator from capture as it is submitted to the Relying Party, the secondary authenticator must be used in an authentication protocol which protects against eavesdropping and man-in-the-middle attacks as described in Section 9.

3. In order to protect the secondary authenticator after it has been used, it must never be transmitted on an unprotected channel or to an unauthenticated party while it is still valid. The secondary authenticator may be sent in the clear only if the sending party has strong assurances that the secondary authenticator will not subsequently be accepted by any Relying Party. This is possible if the secondary authenticator is specific to a single Relying Party, and if that Relying Party will not accept secondary authenticators with the same value until the maximum lifespan of the corresponding assertion has passed.

- *Assertion substitution* – To mitigate this threat, one of the following mechanisms may be implemented:

  1. Responses to assertion requests, signed or integrity protected by the Verifier, may contain the value of the assertion reference used in the request or some other nonce that was cryptographically bound the request by the RP.

  2. Responses to assertion requests may be bound to the corresponding requests by message order, as in HTTP, provided that assertions and requests are protected by a protocol such as TLS that can detect and disallow malicious reordering of packets.

## 10.3 Assertion Assurance Levels

The stipulations for assertion assurance levels are described in the next sections.

### 10.3.1 Threat Resistance per Assurance Level

The table below lists the requirements for assertions (both in the direct and indirect models) and assertion references (in the indirect model) at each assurance level in terms of resistance to the threats listed above.

Table 12 – Threat Resistance per Assurance Level

| Threat | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| Assertion manufacture/modification | Yes | Yes | Yes | Yes |
| Assertion disclosure | No | Yes | Yes | Yes |
| Assertion repudiation | No | No | Yes | Yes |

| Threat | Level 1 | Level 2 | Level 3 | Level 4 |
|---|---|---|---|---|
| Assertion redirect | No | Yes | Yes | Yes |
| Assertion reuse | Yes | Yes | Yes | Yes |
| Secondary authenticator manufacture | Yes | Yes | Yes | Yes |
| Secondary authenticator capture | No | Yes | Yes | Yes |
| Assertion substitution | No | Yes | Yes | Yes |

### *10.3.2 Requirements per Assurance Level*

The following sections summarize the requirements for assertions at each assurance level.

#### *10.3.2.1 Level 1*

All assertions recognized within this guideline must indicate the assurance level of the initial authentication of the Claimant to the Verifier using the former's long-term token(s). The assurance level indication within the assertion may be implicit (e.g., through the identity of the Verifier implicitly indicating the resulting assurance level) or explicit (e.g., through an explicit field within the assertion.)

At Level 1, it must be impractical for an Attacker to manufacture an assertion or assertion reference that can be used to impersonate the Subscriber. If the direct model is used, the assertion which is used must be signed by the Verifier or integrity protected using a secret key shared by the Verifier and Relying Party, and if the indirect model is used, the assertion reference which is used must have a minimum of 64 bits of entropy. Assertions shall be specific to a single transaction, and, if assertion references are used, they shall be freshly generated whenever a new assertion is created by the Verifier. In other words, assertions and assertion references are generated for one time use.

Furthermore, in order to protect assertions against modification in the indirect model, all assertions sent from the Verifier to the Relying Party must either be signed by the Verifier, or transmitted from an authenticated Verifier via a protected channel. In either case, a strong mechanism must be in place which allows the Relying Party to establish a binding between the assertion reference and its corresponding assertion, based on integrity protected (or signed) communications with the authenticated Verifier.

To lessen the impact of captured assertions, assertions that are consumed by a Relying Party which is not part of the same Internet domain as the Verifier must expire within 5 minutes of their creation. Assertions used within a single Internet domain, including assertions contained in or referenced by cookies, however, may last as long as 12 hours.

#### *10.3.2.2 Level 2*

At Level 2, assertions must specify whether the name of the Subscriber is a verified name or a pseudonym. Level 2 assertions shall be protected against manufacture/modification, capture, redirect and reuse., Assertion references shall be protected against manufacture,

capture and reuse. Assertions must include the identity of the intended recipient (Relying Party) and recipients must check recipient identity on incoming assertions.

All stipulations from Level 1 apply. Additionally, assertions, assertion references and any session cookies used by the Verifier or Relying party for authentication purposes, shall be transmitted to the Subscriber through a protected channel which is linked to the primary authentication process in such a way that session hijacking attacks are resisted (see Section 9.2.2 for methods which may be used to protect against session hijacking attacks.) Assertions, assertion references and session cookies shall not be subsequently transmitted over an unprotected channel or to an unauthenticated party while they remain valid. (To this end, any session cookies used for authentication purposes shall be flagged as secure, and redirects used to forward secondary authenticators from the Subscriber to the Relying Party shall specify a secure protocol such as HTTPS.)

To protect assertions against manufacture, modification, and disclosure, assertions which are sent from the Verifier to the Relying Party, whether directly or through the Subscriber's device, shall either be sent via a mutually authenticated channel between the Verifier and Relying Party, or equivalently shall be signed by the Verifier and encrypted for the Relying Party.

All assertion protocols used at Level 2 and above require the use of Approved cryptographic techniques. As such, the use of Kerberos keys derived from user generated passwords is not permitted at Level 2 or above.

#### *10.3.2.3 Level 3*

At Level 3, in addition to Level 2 requirements, assertions shall be protected against repudiation by the Verifier; all assertions used at Level 3 shall be signed. Level 3 assertions shall specify verified names and not pseudonyms.

Kerberos uses symmetric key mechanisms to protect key management and session data, and it does not protect against assertion repudiation. However, based on the high degree of vetting conducted on the Kerberos protocol and its wide deployment, Kerberos tickets are acceptable for use as assertions at Level 3 as long as:

- All Verifiers (Kerberos Authentication Servers and Ticket Granting Servers) are under the control of a single management authority that ensures the correct operation of the Kerberos protocol;
- The Subscriber authenticates to the Verifier using a Level 3 token;
- All Level 3 requirements unrelated to non-repudiation are satisfied.

Also, at Level 3, assertions shall expire within 30 minutes when used within a single domain (e.g. Web browser cookies). Cross-domain assertions shall expire within 5 minutes.

However, in order to deliver the effect of single sign on, the Verifier may re-authenticate the Subscriber prior to delivering assertions to new Relying Parties, using a combination of long term and short term single domain assertions provided that the following assurances are met:

- The Subscriber has successfully authenticated to the Verifier within the last 12 hours.
- The Subscriber can demonstrate that he or she was the party that authenticated to the Verifier. This could be demonstrated, for example, by the presence of a cookie set by the Verifier in the Subscriber's browser.
- The Verifier can reliably determine the "liveness" of the Subscriber with a Relying Party since the last assertion was delivered by the Verifier. This means that the Verifier must have evidence that the Subscriber is actively using the services of the Relying Party and has not been idle for more than 30 minutes.

#### *10.3.2.4 Level 4*

At Level 4, bearer assertions shall not be used to establish the identity of the Claimant to the Relying Party. Assertions made by the Verifier may however be used to bind keys or other attributes to an identity. Holder-of-key assertions may be used, provided that all three requirements below are met:

- The Claimant authenticates to the Verifier using a Level 4 token (as described in Section 7) in a Level 4 authentication protocol (as described in Section 9.)
- The Verifier generates a holder-of-key assertion that references a key that is part of the Level 4 token (used to authenticate to the Verifier) or linked to it through a chain of trust, and;
- The Relying Party verifies that the Subscriber possesses the key that is referenced in the holder-of-key assertion using a Level 4 protocol (where the RP plays the role attributed to the Verifier by Section 9.)

The RP shall maintain records of the assertions it receives, so that if a suspicious transaction occurs at the RP, the key asserted by the Verifier may be compared to the value registered with the CSP. This record keeping requirement allows the RP to detect any attempt by the Verifier to impersonate the Subscriber using fraudulent assertions and may also be useful for preventing the Subscriber from repudiating various aspects of the authentication process.

Kerberos uses symmetric key mechanisms to protect key management and session data, and it does not protect against assertion repudiation by the Subscriber or the Verifier. However, based on the high degree of vetting conducted on the Kerberos protocol and its

wide deployment, Kerberos tickets are acceptable for use as assertions at Level 4 as long as:

- All Verifiers (Kerberos Authentication Servers and Ticket Granting Servers) are under the control of a single management authority that ensures the correct operation of the Kerberos protocol;
- The Subscriber authenticates to the Verifier using a Level 4 token;
- All Level 4 requirements unrelated to non-repudiation are satisfied.

All Level 1-3 requirements for the protection of assertion data remain in force at Level 4.

# 11 References

## *11.1 General References*


[DOJ 2000] *Guide to Federal Agencies on Implementing Electronic Processes* (November 2000), available at: http://www.usdoj.gov/criminal/cybercrime/ecommerce.html

[FIPS 199] *Standards for Security Categorization of Federal Information and Information Systems* (February 2004), available at: http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf

[FIPS 201] *Personal Identity Verification (PIV) of Federal Employees and Contractors*, available at: http://csrc.nist.gov/publications/fips/fips201-1/FIPS-201-1-chng1.pdf

[OMB 04-04] OMB Memorandum M-04-04, *E-Authentication Guidance for Federal agencies*, December 16, 2003, available at: http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf

[OMB 03-22] OMB Memorandum M-03-22, *OMB Guidance for Implementing the Privacy Provisions of the E-Government Act of 2002*, September 26, 2003 available at: http://www.whitehouse.gov/omb/memoranda/m03-22.html.

[KERB] Neuman, C., and T. Ts'o, *Kerberos: An Authentication Service for Computer Networks*, IEEE Communications, vol. 32, no.9, 1994.

[RFC 4120] IETF, RFC 4120, The Kerberos Network Authentication Service (V5), July 2005, available at http://www.ietf.org/rfc/rfc4120.txt

[RFC 1939] IETF, RFC 1939, *Post Office Protocol,* Version 3, May 1996, available at: http://www.ietf.org/rfc/rfc1939.txt

[RFC 2246] IETF, RFC 2246, *The TLS Protocol,* Version 1.0. January 1999, available at: http://www.ietf.org/rfc/rfc2246.txt

[RFC2560] IETF, RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*, available at: http://www.ietf.org/rfc/rfc2560.txt

[RFC 3280] IETF, RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, available at: http://www.ietf.org/rfc/rfc3280.txt

[RFC 3546] IETF, RFC 3546, *Transport Layer Security (TLS) Extensions*, June 2003, available at: http://www.ietf.org/rfc/rfc3546.txt

[SKEY] IETF, RFC 1760, *The S/KEY One Time Password System*, February 1995, available at: http://www.ietf.org/rfc/rfc1760.txt

### 11.2 NIST ITL Bulletins

NIST ITL Bulletins are available at: http://csrc.nist.gov/publications/nistbul/index.html. The following bulletins may be of particular interest to those implementing systems of applications requiring E-authentication.

[ITL Dec02] ITL Bulletin, *Security of Public Webservers*, Dec. 2002

[ITL July02] ITL Bulletin, *Overview: The Government Smartcard Interoperability Specification*, July 2002

[ITL Jan02] ITL Bulletin, *Guideline on Firewalls and Firewall Policy*, January 2002

[ITL Feb00] ITL Bulletin, *Guideline for Implementing Cryptography in the Federal Government*, February 2000

[ITL Dec99] ITL Bulletin, *Operating System Security: Adding to the Arsenal of Security Techniques*, December 1999

[ITL Nov99] ITL Bulleting, *Acquiring and Deploying Intrusion Detection Systems*, November 1999

[ITL Sep99] ITL Bulletin, *Securing Web Servers*, September 1999

[ITL May99] ITL Bulletin, *Computer Attacks: What They Are and How to Defend Against Them*, May 1999

### 11.3 NIST Special Publications

NIST 800 Series Special Publications are available at: http://csrc.nist.gov/publications/nistpubs/index.html. The following publications may be of particular interest to those implementing systems of applications requiring E-authentication.

[SP 800-30] NIST Special Publication 800-30, *Risk Management Guide for Information Technology Systems*, July 2002

[SP 800-31] NIST Special Publication, 800-31, *Intrusion Detection Systems (IDS)*, November 2001

[SP 800-32] NIST Special Publication, 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure*, February 2001

[SP 800-33] NIST Special Publication 800-33, *Underlying Technical Models for Information Technology Security*, December 2001

[SP 800-37] NIST Special Publication 800-37, *Guide for the Security Certification and Accreditation of Federal Information Systems*, May 2004

[SP 800-40] NIST Special Publication 800-40, *Procedures for Handling Security Patches*, September 2002

[SP 800-41] NIST Special Publication 800-41, *Guidelines on Firewalls and Firewall Policy*, January 2002

[SP 800-42] NIST Special Publication 800-42, *Guideline on Network Security Testing*, October 2003

[SP 800-43] NIST Special Publication 800-43, *Guide to Securing Windows 2000 Professional*, November 2002

[SP 800-44] NIST Special Publication 800-44, *Guidelines on Securing Public Web Servers*, September 2002

[SP 800-47] NIST Special Publication 800-47, *Security Guide for Interconnecting Information Technology Systems*, September 2002

[SP 800-52] NIST Special Publication 800-52, *Guidelines for the Selection and Use of Transport Layer Security Implementations*, June 2005

[SP 800-53] NIST Special Publication 800-53 *Recommended Security Controls for Federal Information Systems*, February 2005

[SP 800-53A] NIST Special Publication 800-53A, *Guide for Assessing the Security Controls in Federal Information Systems*, draft

[SP 800-57] NIST Special Publication 800-57, *Recommendation for Key Management – Part 1: General*, May 2006

### 11.4 Federal Information Processing Standards

FIPS can be found at: http://csrc.nist.gov/publications/fips/

[FIPS 46-3] Federal Information Processing Standard Publication 46-3, *Data Encryption Standard (DES)*, NIST, October 25, 1999

[FIPS 140-2] Federal Information Processing Standard Publication 140-2, *Security Requirements for Cryptographic Modules*, NIST, May 25, 2001

[FIPS 180-2] Federal Information Processing Standard Publication 180-2, *Secure Hash Standard (SHS)*, NIST, August 2002

[FIPS186-2] Federal Information Processing Standard Publication 186-2, *Digital Signature Standard (DSS)*, NIST, June 2000

[FIPS 197] Federal Information Processing Standard Publication 197, *Advanced Encryption Standard (AES)*, NIST, November 2001

### 11.5 Certificate Policies

These certificate policies can be found at: http://www.cio.gov/fpkipa/

[FBCA1] *X.509 Certificate Policy For The Federal Bridge Certification Authority (FBCA),* version 2.1 January 12, 2006. Available at: http://www.cio.gov/fpkipa/documents/FBCA_CP_RFC3647.pdf

[FBCA2] *Citizen & Commerce Certificate Policy,* Version 1.0 December 3, 2002. Available at: http://www.cio.gov/fpkipa/documents/citizen_commerce_cpv1.pdf

[FBCA3] *X.509 Certificate Policy for the Common Policy Framework*, Version 2.4 February 15, 2006. Available at: http://www.cio.gov/fpkipa/documents/CommonPolicy.pdf

# Appendix A: Estimating Entropy and Strength

## *Password Entropy*

Passwords represent a very popular implementation of memorized secret tokens. In this case impersonation of an identity requires only that the impersonator obtain the password. Moreover, the ability of humans to remember long, arbitrary passwords is limited, so passwords are often vulnerable to a variety of attacks including guessing, use of dictionaries of common passwords, and brute force attacks of all possible password combinations. There are a wide variety of password authentication protocols that differ significantly in their vulnerabilities, and many password mechanisms are vulnerable to passive and active network attacks. While some cryptographic password protocols resist nearly all direct network attacks, these techniques are not at present widely used and all password authentication mechanisms are vulnerable to keyboard loggers and observation of the password when it is entered. Experience also shows that users are vulnerable to "social engineering" attacks where they are persuaded to reveal their passwords to unknown parties, who are basically "confidence men."

Claude Shannon coined the use of the term "entropy[15]" in information theory. The concept has many applications to information theory and communications and Shannon also applied it to express the amount of actual information in English text. Shannon says, "The entropy is a statistical parameter which measures in a certain sense, how much information is produced on the average for each letter of a text in the language. If the language is translated into binary digits (0 or 1) in the most efficient way, the entropy H is the average number of binary digits required per letter of the original language."[16]

Entropy in this sense is at most only loosely related to the use of the term in thermodynamics. A mathematical definition of entropy in terms of the probability distribution function is:

$$H(X) := -\sum_{x} P(X = x) \log_2 P(X = x)$$

where $P(X=x)$ is the probability that the variable $X$ has the value $x$.

Shannon was interested in strings of ordinary English text and how many bits it would take to code them in the most efficient way possible. Since Shannon coined the term, "entropy" has been used in cryptography as a measure of the difficulty in guessing or determining a password or a key. Clearly the strongest key or password of a particular size is a truly random selection, and clearly, on average such a selection cannot be compressed. However it is far from clear that compression is the best measure for the strength of keys and passwords, and cryptographers have derived a number of alternative forms or definitions of entropy, including "guessing entropy" and "min-entropy." As

[15] C. E. Shannon, "A mathematical Theory of Communication," *Bell System Technical Journal*, v. 27, pp. 379-423, 623-656, July, October 1948, see http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html

[16] C. E. Shannon, "Prediction and Entropy of Printed English", *Bell System Technical Journal*, v.30, n. 1, 1951, pp. 50-64.

applied to a distribution of passwords the guessing entropy is, roughly speaking, an estimate of the average amount of work required to guess the password of a selected user, and the min-entropy is a measure of the difficulty of guessing the easiest single password to guess in the population.

If we had a good knowledge of the frequency distribution of passwords chosen under a particular set of rules, then it would be straightforward to determine either the guessing entropy or the min-entropy of any password. An Attacker who knew the password distribution would find the password of a chosen user by first trying the most probable password for that chosen username, then the second most probable password for that username and so on in decreasing order of probability until the Attacker found the password that worked with the chosen username. The average for all passwords would be the guessing entropy. The Attacker who is content to find the password of any user would follow a somewhat different strategy, he would try the most probable password with every username, then the second most probable password with every username, until he found the first "hit." This corresponds to the min-entropy.

Unfortunately, we do not have much data on the passwords users choose under particular rules, and much of what we do know is found empirically by "cracking" passwords, that is by system administrators applying massive dictionary attacks to the files of hashed passwords (in most systems no plaintext copy of the password is kept) on their systems. NIST would like to obtain more data on the passwords users actually choose, but, where they have the data, system administrators are understandably reluctant to reveal password data to others. Empirical and anecdotal data suggest that many users choose very easily guessed passwords, where the system will allow them to do so.

## *A.1 Randomly Selected Passwords*

We use the term here, "entropy" denotes the uncertainty in the value of a password. Entropy of passwords is conventionally expressed in bits. If a password of k bits is chosen at random there are 2k possible values and the password is said to have k bits of entropy. If a password of length l characters is chosen at random from an alphabet of b characters (for example the 94 printable ISO characters on a typical keyboard) then the entropy of the password is $b^l$ (for example if a password composed of 8 characters from the alphabet of 94 printable ISO characters the entropy is $948 \approx 6.09 \times 1015$ – this is about 252, so such a password is said to have about 52 bits of entropy). For randomly chosen passwords, guessing entropy, min-entropy, and Shannon entropy are all the same value. The general formula for entropy, H is given by:

$$H = \log_2 (b^l)$$

Table A.1 gives the entropy versus length for a randomly generated password chosen from the standard 94 keyboard characters (not including the space). Calculation of randomly selected passwords from other alphabets is straightforward.

## *A.2 User Selected Passwords*

It is much more difficult to estimate the entropy in passwords that users choose for themselves, because they are not chosen at random and they will not have a uniform random distribution. Passwords chosen by users probably roughly reflect the patterns and character frequency distributions of ordinary English text, and are chosen by users so that they can remember them. Experience teaches us that many users, left to choose their own passwords will choose passwords that are easily guessed and even fairly short dictionaries of a few thousand commonly chosen passwords, when they are compared to actual user chosen passwords, succeed in "cracking" a large share of those passwords.

### A.2.1 Guessing Entropy Estimate

Guessing entropy is arguably the most critical measure of the strength of a password system, since it largely determines the resistance to targeted, in band password guessing attacks.

In these guidelines, we have chosen to use Shannon's estimate of the entropy in ordinary English text as the starting point to estimate the entropy of user-selected passwords. It is a big assumption that passwords are quite similar to other English text, and it would be better if we had a large body of actual user selected passwords, selected under different composition rules, to work from, but we have no such resource, and it is at least plausible to use Shannon's work for a "ballpark" estimate. Readers are cautioned against interpreting the following rules as anything more than a very rough rule of thumb method to be used for the purposes of E-authentication.

Shannon conducted experiments where he gave people strings of English text and asked them to guess the next character in the string. From this he estimated the entropy of each successive character. He used a 27-character alphabet, the ordinary English lower case letters plus the space.

In the following discussion we assume that passwords are user selected from the normal keyboard alphabet of 94 printable characters, and are at least 6-characters long. Since Shannon used a 27 character alphabet it may seem that the entropy of user selected passwords would be much larger, however the assumption here is that users will choose passwords that are almost entirely lower case letters, unless forced to do otherwise, and that rules that force them to include capital letters or non-alphabetic characters will generally be satisfied in the simplest and most predictable manner, often by putting a capital letter at the start (as we do in ordinary English) and punctuation or special characters at the end, or by some simple substitution, such as $ for the letter "s." Moreover rules that force passwords to appear to be highly random will be counterproductive because they will make the passwords hard to remember. Users will then write the passwords down and keep them in a convenient (that is insecure) place, such as pasted on their monitor. Therefore it is reasonable to start from estimates of the entropy of simple English text, assuming only a 27-symbol alphabet.

Shannon observed that, although there is a non-uniform probability distribution of letters, it is comparatively hard to predict the first letter of an English text string, but, given the first letter, it is much easier to guess the second and given the first two the third is easier still, and so on. He estimated the entropy of the first symbol at 4.6 to 4.7 bits, declining to on the order of about 1.5 bits after 8 characters. Very long English strings (for example the collected works of Shakespeare) have been estimated to have as little as .4 bits of entropy per character.[17] Similarly, in a string of words, it is harder to predict the first letter of a word than the following letters, and the first letter carries about 6 times more information than the 5th or later letters[18].

An Attacker attempting to find a password will try the most likely chosen passwords first. Very extensive dictionaries of passwords have been created for this purpose. Because users often choose common words or very simple passwords systems commonly impose rules on password selection in an attempt to prevent the choice of "bad" passwords and improve the resistance of user chosen passwords to such dictionary or rule driven password guessing attacks. For the purposes of these guidelines, we break those rules into two categories:

1. Dictionary tests that test prospective passwords against an "extensive dictionary test" of common words and commonly used passwords, then disallow passwords found in the dictionary. We do not precisely define a dictionary test, since it must be tailored to the password length and rules, but it should prevent selection of passwords that are simple transformations of any one word found in an unabridged English dictionary, and should include at least 50,000 words. There is no intention to prevent selection of long passwords (16 characters or more based on phrases) and no need to impose a dictionary test on such long passwords of 16 characters or more.
2. Composition rules that typically require users to select passwords that include lower case letters, upper case letters, and non-alphabetic symbols (e.g.;: "~!@#$%^&*()_-+={}[]|\:;'<,>.?/1234567890").

Either dictionary tests or composition rules eliminate some passwords and reduce the space that an adversary must test to find a password in a guessing or exhaustion attack. However they can eliminate many obvious choices and therefore we believe that they generally improve the "practical entropy" of passwords, although they reduce the work required for a truly exhaustive attack. The dictionary check requires a dictionary of at least 50,000 legal passwords chosen to exclude commonly selected passwords. Upper case letters in candidate passwords converted to lower case before comparison.

Table A.1 provides a rough estimate of the average entropy of user chosen passwords as a function of password length. Estimates are given for user selected passwords drawn from the normal keyboard alphabet that are not subject to further rules, passwords subject to a dictionary check to prevent the use of common words or commonly chosen passwords

[17] Thomas Schurmann and Peter Grassberger, "Entropy estimation of symbol sequences," http://arxiv.org/ftp/cond-mat/papers/0203/0203436.pdf
[18] *ibid.*

and passwords subject to both composition rules and a dictionary test. In addition an estimate is provided for passwords or PINs with a ten-digit alphabet. The table also shows the calculated entropy of randomly selected passwords and PINs. The values of Table A.1 should not be taken as accurate estimates of absolute entropy, but they do provide a rough relative estimate of the likely entropy of user chosen passwords, and some basis for setting a standard for password strength.

The logic of the Table A.1 is as follows for user-selected passwords drawn from the full keyboard alphabet:

- The entropy of the first character is taken to be 4 bits;
- The entropy of the next 7 characters are 2 bits per character; this is roughly consistent with Shannon's estimate that "when statistical effects extending over not more than 8 letters are considered the entropy is roughly 2.3 bits per character;"
- For the 9th through the 20th character the entropy is taken to be 1.5 bits per character;
- For characters 21 and above the entropy is taken to be 1 bit per character;
- A "bonus" of 6 bits of entropy is assigned for a composition rule that requires both upper case and non-alphabetic characters. This forces the use of these characters, but in many cases thee characters will occur only at the beginning or the end of the password, and it reduces the total search space somewhat, so the benefit is probably modest and nearly independent of the length of the password;
- A bonus of up to 6 bits of entropy is added for an extensive dictionary check. If the Attacker knows the dictionary, he can avoid testing those passwords, and will in any event, be able to guess much of the dictionary, which will, however, be the most likely selected passwords in the absence of a dictionary rule. The assumption is that most of the guessing entropy benefits for a dictionary test accrue to relatively short passwords, because any long password that can be remembered must necessarily be a "pass-phrase" composed of dictionary words, so the bonus declines to zero at 20 characters.

For user selected PINs the assumption of Table A.1 is that such pins are subjected at least to a rule that prevents selection of all the same digit, or runs of digits (e.g., "1234" or "76543"). This column of Table A.1 is at best a very crude estimate, and experience with password crackers suggests, for example, that users will often preferentially select simple number patterns and recent dates, for example their year of birth.

### A.2.2 Min Entropy Estimates

Experience suggests that a significant share of users will choose passwords that are very easily guessed ("password" may be the most commonly selected password, where it is allowed). Suppose, for example, that one user in 1,000 chooses one of the 2 most common passwords, in a system that allows a user 3 tries before locking a password. An

Attacker with a list of user names, who knows the two most commonly chosen passwords can use an automated attack to try those 2 passwords with each user name, and can expect to find at least one password about half the time by trying 700 usernames with those two passwords. Clearly this is a practical attack if the only goal is to get access to the system, rather than to impersonate a single selected user. This is usually too dangerous a possibility to ignore.

We know of no accurate general way to estimate the actual min-entropy of user chosen passwords, without examining in detail the passwords that users actually select under the rules of the password system, however it is reasonable to argue that testing user chosen passwords against a sizable dictionary of otherwise commonly chosen legal passwords, and disallowing matches, will raise the min entropy of a password. A dictionary test is specified here that is intended to ensure at least 10-bits of min entropy. That test is:

- Upper case letters in passwords are converted to entirely lower case and compared to a dictionary of at least 50,000 commonly selected otherwise legal passwords and rejected if they match any dictionary entry, and
- Passwords that are detectable permutations of the username are not allowed.

This is estimated to ensure at least 10-bits of min entropy. Other means may be substituted to ensure at least 10 bits of min-entropy. User chosen passwords of at least 15 characters are assumed to have at least 10-bits of min-entropy. For example a user might be given a short randomly chosen string (two randomly chosen characters from a 94-bit alphabet have about 13 bits of entropy). A password, for example, might combine short system selected random elements, to ensure 10-bits of min-entropy, with a longer user-chosen password.

## *A.2 Other Types of Passwords*

Some password systems require a user to memorize a number of images, such as faces. Users are then typically presented with successive fields of several images (typically 9 at a time), each of which contains one of the memorized images. Each selection represents approximately 3.17 bits of entropy. If such a system used five rounds of memorized images, then the entropy of system would be approximately 16 bits. Since this is randomly selected password the guessing entropy and min-entropy are both the same value.

It is possible to combine randomly chosen and user chosen elements into a single composite password. For example a user might be given a short randomly selected value to ensure min-entropy to use in combination with a user chosen password string. The random component might be images or a character string.

## *A.3 Examples*

The intent of these guidelines is to allow designers and implementers flexibility in designing password authentication systems. System designers can trade off password

length, rules and measures imposed to limit the number of guesses an adversary can attempt.

The approach of this recommendation to password strength is that it is a measure of the probability that an Attacker, who knows nothing but a user's name, can discover the user's password by means of "in-band" password guessing attack. That is the Attacker attempts to try different passwords until he/she authenticates successfully. At each level given below, the maximum probability that, over the life of the password, an Attacker with no *a priori* knowledge of the password will succeed in an in-band password guessing attack is:

1. Level 1- $2^{-10}$ (1 in 1024)
2. Level 2 - $2^{-14}$ (1 in 16,384)

Consider a system that assigns Subscribers 6 character passwords, randomly selected from an alphabet of 94 printable keyboard characters. From Table A.1 we see that such a password is considered to have 39.5 bits of entropy. If the authentication system limits the number of possible unsuccessful authentication trials to $2^{39.5}/2^{14} = 2^{25.5}$ trials, the password strength requirements of Level 2 are satisfied. The authentication system could, for example, simply maintain a counter that locked the password after $2^{25.5}$ (about forty-five million) total unsuccessful trials. An alternative scheme would be to lock out the Claimant for a minute after three successive failed authentication attempts. Such a lock out would suffice to limit automated attacks to 3 trials a minute and it would take about 90 years to carryout $2^{25.5}$ trials. If the system required that password authentication attempts be locked for one minute after three unsuccessful trials and that passwords be changed every ten years, then the targeted password guessing attack requirements of Level 2 would be comfortably satisfied. Because the min-entropy of a randomly chosen password is the same as the guessing entropy, the min-entropy requirements of Level 2 are met.

Consider a system that used:

- A minimum of 8 character passwords, selected by Subscribers from an alphabet of 94 printable characters,
- Required Subscribers to include at least one upper case letter, one lower case letter, one number and one special character, and;
- Used a dictionary to prevent Subscribers from including common words and prevented permutations of the username as a password.

Such a password would meet the composition and dictionary rules for user-selected passwords in Appendix A, and from Table A.1 we estimate guessing entropy at 30 bits. Any system that limited a Subscriber to less than $2^{16}$ (about 65,000) failed authentication attempts over the life of the password would satisfy the targeted guessing attack requirements of Level 2. For example, consider a system that required passwords to be changed every two years and limited trials by locking an account for 24 hours after 6 successive failed authentication attempts. An Attacker could get $2 \times 365 \times 6 = 4{,}380$

attempts during the life of the password and this would easily meet the targeted attack requirements of Level 2. Because of the dictionary test, this would also meet the min-entropy rules for Level 2.

There are many strategies for limiting the number of in-band password trials available to an attacker, including a simple cap on the maximum number of bad password attempts, before locking the account. This guideline does not require any particular strategy. However it is fairly common to implement a strategy that allows three successive unsuccessful password attempts then locks the account for a "back-off" period of time, before allowing three more attempts, and so on, until the maximum password lifetime is exceeded. The password must then be changed. For convenience, Table A.2 gives the minimum password entropy needed for various back-off periods and password lifetimes in order to satisfy the Level 2 requirement that the probability of a successful in-band password guessing attack be less than $2^{-14}$ (1 in 16,384).

It will be very hard to impose dictionary rules on longer passwords, and many people may prefer to memorize a relatively long "pass-phrases" of words, rather than a shorter, more arbitrary password. An example might be: "IamtheCapitanofthePina4".

As an alternative to imposing some arbitrary specific set of rules, an authentication system might grade user passwords, using the rules stated above, and accept any that meet some minimum entropy standard. For example, suppose passwords with at least 24-bits of entropy were required. We can calculate the entropy estimate of "IamtheCapitanofthePina4" by observing that the string has 23 characters and would satisfy a composition rule requiring upper case and non-alphabetic characters. Table A.1 estimates 45 bits of guessing entropy for this password.

Table A.1 – Estimated Password Guessing Entropy in bits vs. Password Length

| | User Chosen | | | | Randomly Chosen | |
|---|---|---|---|---|---|---|
| | 94 Character Alphabet | | | 10 char. alphabet | | 94 char alphabet |
| **Length Char.** | **No Checks** | **Dictionary Rule** | **Dict. & Comp. Rule** | | | |
| **1** | 4 | - | - | 3 | 3.3 | 6.6 |
| **2** | 6 | - | - | 5 | 6.7 | 13.2 |
| **3** | 8 | - | - | 7 | 10.0 | 19.8 |
| **4** | 10 | 14 | 16 | 9 | 13.3 | 26.3 |
| **5** | 12 | 17 | 20 | 10 | 16.7 | 32.9 |
| **6** | 14 | 20 | 23 | 11 | 20.0 | 39.5 |
| **7** | 16 | 22 | 27 | 12 | 23.3 | 46.1 |
| **8** | 18 | 24 | 30 | 13 | 26.6 | 52.7 |
| **10** | 21 | 26 | 32 | 15 | 33.3 | 65.9 |
| **12** | 24 | 28 | 34 | 17 | 40.0 | 79.0 |
| **14** | 27 | 30 | 36 | 19 | 46.6 | 92.2 |
| **16** | 30 | 32 | 38 | 21 | 53.3 | 105.4 |
| **18** | 33 | 34 | 40 | 23 | 59.9 | 118.5 |
| **20** | 36 | 36 | 42 | 25 | 66.6 | 131.7 |
| **22** | 38 | 38 | 44 | 27 | 73.3 | 144.7 |
| **24** | 40 | 40 | 46 | 29 | 79.9 | 158.0 |
| **30** | 46 | 46 | 52 | 35 | 99.9 | 197.2 |
| **40** | 56 | 56 | 62 | 45 | 133.2 | 263.4 |

Figure A.1 - Estimated User Selected Password Entropy vs. Length

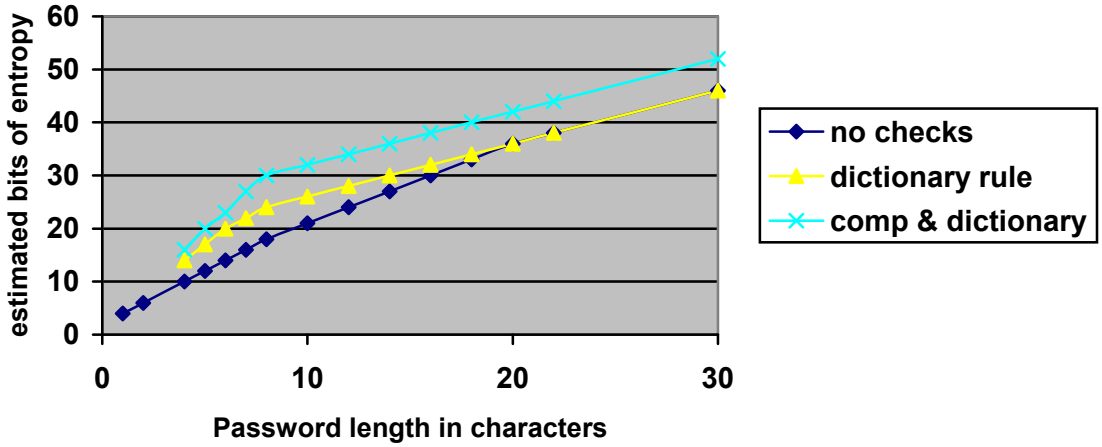


Table A.2 – Required Password Entropy for Level 2 and various Password Lifetimes and Back-off Periods

| | **Bits of Password Entropy Needed for Level 2** | | | | |
|---|---|---|---|---|---|
| **Back-off after 3 failures** | **90 day password lifetime** | **180 day password lifetime** | **1 year password lifetime** | **2 year password lifetime** | **5 year password lifetime** |
| **1 min.** | 33 | 34 | 35 | 36 | 37 |
| **10 min.** | 29 | 30 | 31 | 32 | 34 |
| **1 hour** | 27 | 28 | 29 | 30 | 31 |
| **1 day** | 22 | 23 | 24 | 25 | 26 |

# Appendix B: Mapping of Federal PKI Certificate Policies to E-authentication Assurance Levels

Agencies are, in general, issuing certificates under the policies specified in the Common Policy Framework [FBCA3] to satisfy FIPS 201. Agencies that were early adopters of PKI technology, and organizations outside the Federal government, issue PKI certificates under organization specific policies instead of the Common Policy Framework. The primary mechanism for evaluating the assurance provided by public key certificates issued under organization specific policies is the policy mapping of the Federal Policy Authority to the Federal Bridge CA policies. These policies include the Rudimentary, Basic, Medium, Medium-HW, and High assurance policies specified in [FBCA1] and the Citizen and Commerce class policy specified in [FBCA2].

The table below summarizes how certificates issued under the Common Policy Framework correspond to the E-authentication assurance levels. Note that the Card Authentication and Common Device policies are not listed; these policies support authentication of a system or a cryptographic module rather than a person. In addition, table B.1 summarizes how organization specific certificate policies correspond to E-authentication assurance levels. At Level 2 agencies may use certificates issued under policies that have not been mapped by the Federal policy authority, but are determined to meet the Level 2 identify proofing, token and status reporting requirements. For Levels 3 and 4, agencies shall depend upon the mappings provided by the Federal PKI. Organizations outside the Federal Government may select their own trust anchors.

Table B.1 – Certificate Policies and the E-authentication Assurance Levels

| Certificate Policy | Selected Policy Components | | | Overall Equivalence |
|---|---|---|---|---|
| | Identity Proofing | Token | Token and Credential Management[19] | |
| Common-Auth | Meets Level 4 | Meets Level 4 | Meets Level 4 | Meets Level 4 |
| Common-SW | Meets Level 4 | Meets Level 3 | Meets Level 4 | Meets Level 3 |
| Common-HW | Meets Level 4 | Meets Level 4 | Meets Level 4 | Meets Level 4 |
| Common-High | Meets Level 4 | Meets Level 4 | Meets Level 4 | Meets Level 4 |
| Citizen and Commerce Class[20] | Meets Level 2 | Meets Level 2 | Meets Level 2 | Meets Level 2 |
| Basic[20] | Meets Level 3 | Meets Level 3 | Meets Level 3 | Meets Level 3 |
| Medium[20] | Meets Level 4 | Meets Level 3 | Meets Level 4 | Meets Level 3 |
| Medium-HW[20] | Meets Level 4 | Meets Level 4 | Meets Level 4 | Meets Level 4 |
| High[20] | Meets Level 4 | Meets Level 4 | Meets Level 4 | Meets Level 4 |

The Federal PKI has also added two policies, Medium Commercial Best Practices (Medium-CBP) and Medium Hardware Commercial Best Practices (MediumHW-CBP)

19 The key component in token and credential management is the credential status mechanism.
20 These policies are not asserted in the user certificates, but equivalence is established through policy mapping at the Federal Bridge CA

to support recognition of non-Federal PKIs. In terms of E-authentication levels, the Medium CBP and MediumHW-CBP are equivalent to Medium and Medium-HW, respectively.