CMS

CENTERS FOR MEDICARE & MEDICAID SERVICES
OFFICE OF INFORMATION SERVICES

CENTERS for MEDICARE & MEDICAID SERVICES
Enterprise Information Security Group
7500 Security Boulevard
Baltimore, Maryland 21244-1850

ESG
Enterprise Information
Security Group

Risk Management, Oversight,
And Monitoring

Risk Management Handbook
Volume III
Standard 3.1
CMS Authentication Standards

FINAL
Version 1.3
April 17, 2014

Document Number: CMS-CISO-2014-vIII-std3.1

(This Page Intentionally Blank)

SUMMARY OF CHANGES IN *CMS AUTHENTICATION STANDARDS* VERSION 1.3, DATED APRIL 17, 2014

1. This version makes updates to address modifications to e-authentications requirements in National Institute of Standards and Technology (NIST) Special Publication (SP) 800-63-2, *Electronic Authentication Guideline*, dated August 2013. The substantive changes in the revised 800-63 are intended to facilitate the use of professional credentials in the identity proofing process, and to reduce the need to use postal mail to an address of record to issue credentials for Level 3 remote registrations.
2. Additional changes have been made to clarify the full transactional nature of the entire credential issuance, assertion, authentication, and authorization processes.

SUMMARY OF CHANGES IN *CMS AUTHENTICATION STANDARDS* VERSION 1.2, DATED JULY 31, 2012

1. Updated to address modifications to e-authentications requirements directed by Federal Information Processing Standards (FIPS) 200 and expanded upon in NIST SP 800-63-1, *Electronic Authentication Guideline,* dated December 2011.
2. Moved remaining CMS e-authentication guidance and direction from Appendix D of the *CMS Information Security Acceptable Risk Safeguards (ARS), CMS Minimum Security Requirements (CMSR)* manual into this standard, and deleted from the ARS Appendix D.

SUMMARY OF CHANGES IN *CMS AUTHENTICATION STANDARDS* VERSION 1.1, DATED AUGUST 31, 2010

1. Baseline Version.

(This Page Intentionally Blank)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

(This Page Intentionally Blank)

1 INTRODUCTION

This document provides the Centers for Medicare & Medicaid Services (CMS) position and standard on the use of authentication mechanisms in CMS systems. This standard is based on identity management and authentication standards published in: National Institute of Standards and Technology (NIST) Special Publication (SP) 800-63, *Electronic Authentication Guideline*, NIST SP 800-53, *Security and Privacy Controls for Federal Information Systems and Organizations*, and *Homeland Security Presidential Directive (HSPD) 12*[1], and various other NIST publications and Government Directives associated with minimum Federal authentication standards.

2 AUTHENTICATION COMPONENTS AND MECHANISMS

2.1 IDENTITY

The information technology world defines *Identity* as the individual characteristics by which a thing or person is recognized or known. A *digital identity* is the electronic representation of a real-world entity, and is usually taken to represent the online *equivalent* of a real individual. This online equivalent of an individual participates in electronic transactions on behalf of the individual it represents. Typically, digital identities are established and represented in the form of a unique identifier, such as a *User ID*, to represent an individual during a transaction. Note that a broader definition can also assign digital identities to organizations, companies, and even individual electronic devices.

A digital identity is often used jointly with one or more *credentials* that make credible assertions about an entity, and a *digital identity* claimed (asserted) by that entity. That is, these digital identities assert that they are, in fact, a valid representation of the individual whose real identity they represent.

> *In a non-digital application, a* ***credit card number*** *is a unique characteristic associated with an* ***identity*** *by which a shopper is identified at a store for conducting a credit-based sales transaction. The valid credit card number establishes that certain rights to purchase on credit, on behalf of a* ***specific individual,*** *have been established in the name of the real individual to whom the card was issued. The unique credit card number is used by the credit card issuer as the asserted identity (unique identifier) of the cardholder while the card itself acts as a credential token. Any* ***authorized*** *purchases made under the credit card number will ultimately be linked back to the individual to whom the card number was issued—and that individual will be held accountable for those purchases.*

[1] HSPD-12, *Policy for a Common Identification Standard for Federal Employees and Contractors*, is available at the US Department of Homeland security at http://www.dhs.gov/xabout/laws/gc_1217616624097.shtm.

2.2 CREDENTIAL

A *credential* is an attestation of a unique identity issued to an individual by a *trusted third party* with *authority* to do so. A credential *token* is an object that is verified when presented to the verifier in an *authentication* transaction. Credential issuers will associate a *credential* to a unique identity in several ways, though the use of the credential *token* and specialized attributes specifically associated that token. The token is normally associated with attributes that are one of three different types:

1. Something that the credential holders physically *possess* or ***have***, such as smart cards, hardware tokens, digital certificates, electronic tokens, or even a specific mobile phone,

2. Something that the credential holder ***knows***, such as passwords or secret (challenge) questions, or

3. Something that the credential holders physically ***are***, such as iris- or retina-patterns, fingerprints, facial characteristics, etc.

   In a non-digital example, a ***credential*** *token (such as a government-issued driver's license), issued by a* ***trusted issuer*** *(the State Department of Motor Vehicles [DMV]), is* ***presented*** *by a shopper to a store proprietor in order to establish that the asserted or claimed identity (in this case, the identity represented by the credit card number) belongs to the presenter of that identity (the shopper). This credential token is linked to the asserted identity through several attributes (such as the birthdate, address, and photograph on the ID.) The purpose of presenting the credential is so that the store proprietor can examine the token to verify that the shopper is (or represents) the actual asserted identity.*

In the example above, we can immediately see that the integrity of the entire transaction is dependent on how much the proprietor *trusts* the credentials (both the credit card and the driver's license). This trust is dependent on how much the store proprietor *trusts* that the *credential issuers* (both the credit card issuer and the DMV) have ***not*** *issued the credential to an* ***imposter***. In order to earn this trust, *credential issuers* must ensure that credentials are issued to individuals ***only after*** their *real* identity has been diligently *verified* by the credential issuer (to an appropriate level of assurance) and after a proper authority has *authorized* the issuance of the credential token. This process is known as *proofing*. (Note that credential issuers must also *convince* the *proprietor* that the *credential issuers* are *always diligent* concerning the proofing process in order to *earn* the proprietor's needed *trust* in the credential.)

The process of identity verification of the individual by the credential issuer is known as ***proofing***. The proofing process involves two steps:

1. Ensuring that the asserted identity is a *real* identity (avoiding the issuance of a credit card to *"Mickey Mouse"*), and

2. Ensuring that the person *asserting* to be that identity, *really is* that asserted identity.

The issuance of a credential, and the associated identity proofing process, has often been described as *binding* the *identity* to the *credential*. This refers to the issuing authority creating a trusted record that contains a unique reference to both the *individual* and *credential*. When we accept a presented credential, we are *trusting* that the credential issuer has completed the necessary proofing requirements before they issued the associated credential. This *trust relationship* is a significant factor when we contemplate accepting a credentials issued by an *outside* (or third-party) authority. This concept of issuing and accepting credentials from an outside authority (such as in the DMV, or the credit card company in the example above) is known as identity *federation* or *federated identity management*.

Note that if credential issuers are *not diligent* in their credential issuance and management, then "evil-doers" will abuse that lack of diligence by conducting fraudulent transactions until the trust relationship is destroyed, and all future transactions will be denied. Credit card companies go bankrupt, stores loose sales, and shoppers cannot buy anything on credit—everyone loses.

2.3 AUTHENTICATION

Authentication is the act of establishing or confirming someone (or something) as authentic. Much like the *proofing* process, this involves *confirming* the asserted identity of a person is real, tracing the origins of an artifact (the credential token), and ensuring that an entity (user, process, application, or machine) is who, or what, they assert to be. In a digital environment, authentication involves the verification of one or more presented trusted credential tokens.

In our continuing non-digital example, the store proprietor seeks to ensure that the presenter of the credit card is authorized to use it. The store proprietor evaluates the presented credential tokens—both the driver's license and the credit card—and verifies that the credentials are valid by ensuring the following:

- *Verify that the credential tokens (both the driver's license and the credit card) are* authentic, *that they are actually issued by the* trusted issuers, *and are* not forged. *So, the store proprietor checks for the expected watermarks, holograms, magnetic strips, etc. that the issuer places on the token specifically for that purpose. (Note the credit card number is usually also validated electronically by the credit card issuer during final stages of the transaction.)*
- *Verify that the credentials are still* valid. *The proprietor checks the expiration date on both the driver's license and the credit card.*
- *Verify that the license is* applicable *to the presenter—that the asserted* identity *matches the* credential. *In this case, the proprietor looks at the picture on the driver's license (an* attribute *of the credential token) to ensure it matches something that the presenter "is". In other words, the proprietor verifies that the picture matches the presenter's face.*
- *Verifies that the presenter is* authorized *to complete the transaction. For instance:*
    - *The proprietor verifies that the name on the driver's license (another attribute of the credential token) matches the name (corresponding attribute) on the presented credit card. So, if: face=driver's license picture; and driver's license name =credit card name, then the presenter is authorized to use the credit card.*
    - *For higher sensitivity transactions, such as if the presenter were purchasing alcohol, the proprietor would also verify that the asserted identity is* authorized *for this*

> *particular (higher-sensitivity) transaction by verifying* additional *(multiple) attributes—i.e., that the birthdate attribute on the presented driver's license indicates that the presenter is over 21 years old.*

When all of these verifications are successful, only then can the remainder of the transaction proceed with assurance.

2.3.1 MULTI-FACTOR AUTHENTICATION

Multi-factor authentication is generally required to access CMS *sensitive* data. Multi-factor authentication (required as specified in the *Identification and Authentication [IA]* family of the Acceptable Risk Safeguards [ARS]) uses a combination of two (or more) different token attributes (also known as factors), discussed above, to authenticate the asserted user.

- The first is what users ***know***. This is usually a password, but this can also include a user response to a secret *challenge question*. (This is generally known as *Knowledge Based Authentication,* and by itself, is insufficient for authentication to most CMS sensitive information.)
- The second is what users ***have***. This could be a physical object (*hard* token), for example, a smart card, or hardware token that generates one-time-only passwords. It might also be some encrypted software token (*soft* token) installed on an individual's system (usually with very limited functional parameters for use).
- The third is who users ***are***, as indicated by some biometric characteristic such as a fingerprint or an iris pattern.

Two-factor authentication means that instead of using only one single type of authentication token factor, such as only things a user *knows* (passwords, shared secrets, solicited personal information, etc.), a second token or factor, something the user *has* or something the user *is*, must *also* be supplied in order to complete the authentication process.

> *If you review the non-digital example in the previous section, you can see that even the simple example of the common credit card sales transaction is properly conducted as a* multi-factor *authentication transaction. The proprietor first validates physical possession of both the credit card and the driver's license (something they have), and then verifies a physical attribute of the presenter (something they are) with a presented credential attribute (the proprietor matches the picture on driver's license with the presenter's face.)*

Two-factor authentication is not a new or unique concept. Two-factor authentication is also used every time a bank customer visits the local Automated Teller Machine (ATM). One authentication factor is the physical ATM card the customer slides into the ATM (something they *have*). The second factor is the Personal Identification Number (PIN) they enter (something they *know*). If the bank customer is without either of these, user authentication cannot take place, and the ATM transaction is denied.

3 HUMAN USER AUTHENTICATION

Human user authentication is the process that provides a level of confidence that a *human* person (as opposed to *machine*), who is interacting with an electronic system, is who they assert to be. *Authorization* is the process of enforcing access control policies: determining what types or qualities of *activities*, *resources*, or *services* a user is permitted. The determination of *authorization* typically occurs within the context of whole authentication process, but *after* user identity is *verified* as the asserted identity. Only then is it determined if they are to be *authorized* for different types of access or activity.

There are several federal standards and practices associated with the *proofing*, *credentialing*, *authentication*, *authorization*, and even specifics technologies necessary to *manage* human users.

3.1 PERSONAL IDENTITY VERIFICATION (PIV) CARDS

3.1.1 HOMELAND SECURITY PRESIDENTIAL DIRECTIVE 12

Homeland Security Presidential Directive (HSPD) 12[2], dated August 27, 2004, entitled *Policy for a Common Identification Standard for Federal Employees and Contractors*, directed the promulgation of a Federal standard for secure and reliable forms of *identification* for Federal employees and contractors.

The purpose was to create standardized, interoperable *Personal Identity Verification* (PIV) cards, capable of being used as employee and contractor identification, and allowing for both *Physical access*[3] and *Logical access*[4] to federally controlled facilities and information systems.

It is the policy of the United States to enhance security, increase Government efficiency, reduce identity fraud, and protect personal privacy, by establishing a mandatory Government-wide standard for secure and reliable forms of identification issued by the Federal Government to its employees, contractors and subcontractors.

As directed in HSPD-12, the NIST *Computer Security Division* initiated a new program for improving the identification and authentication of Federal employees and contractors for access to Federal facilities and information systems. Federal Information Processing Standard (FIPS) 201[5], entitled *Personal Identity Verification of Federal Employees and Contractors*, was developed to satisfy the requirements of HSPD-12, approved by the Secretary of Commerce, and issued on February 25, 2005, and amended by Change Notice 1 on June 23, 2006.

[2] HSPD-12, *Policy for a Common Identification Standard for Federal Employees and Contractors*, is available at the US Department of Homeland security at http://www.dhs.gov/xabout/laws/gc_1217616624097.shtm.

[3] *Physical Access* means routine, unescorted, or unmonitored access to non-public areas of a federally-controlled facility.

[4] *Logical Access* means routine, unsupervised, non-public access to a CMS FISMA system.

[5] FIPS 201 (as amended by Change Notice 1) is available at http://csrc.nist.gov/publications/fips/fips201-1/FIPS-201-1-chng1.pdf. (Change Notice 1 provided changes to: 1] the graphics on the back of the PIV card, and 2] the ASN.1 encoding of NACI indicator.)

FIPS 201 incorporates three NIST Special Publications[6] specifying several aspects of the required administrative procedures and technical specifications that may change as the standard is implemented and used.

- NIST Special Publication 800-73, *Interfaces for Personal Identity Verification* specifies the interface and data elements of the PIV card;
- NIST Special Publication 800-76, *Biometric Data Specification for Personal Identity Verification* specifies the technical acquisition and formatting requirements for biometric data of the PIV system; and
- NIST Special Publication 800-78, *Cryptographic Algorithms and Key Sizes for Personal Identity Verification* specifies the acceptable cryptographic algorithms and key sizes to be implemented and used for the PIV system.

There is ***no provision for waivers*** to standards issued by the Secretary of Commerce under the Federal Information Security Management Act of 2002 (FISMA). Likewise, HSPD-12 also has no waiver provision.

On February 3, 2011, the Office of Management and Budget (OMB) issued Memorandum M-11-11[7], *Continued Implementation of Homeland Security Presidential Directive (HSPD) 12– Policy for a Common Identification Standard for Federal Employees and Contractors*, which stated the following:

- *Effective immediately*, all new systems under development must use HSPD-12 compliant PIV cards prior to being made operational.
- Starting in fiscal year 2012, existing physical and logical access control systems must be upgraded to use PIV cards prior to the agency using funding for further development or technology refresh.
- All procurements for products and services for facility and system access control must meet HSPD-12 standards and the Federal Acquisition Regulations to ensure interoperability.
- Agencies will accept and electronically verify secure Identification (ID) cards issued by other agencies.
- Solutions align with and implement the *Federal Identity, Credential and Access Roadmap and Implementation Guidance (FICAM).*[8]

3.1.2 TO WHOM DOES HSPD-12 APPLY?

As defined on OMB memorandum M-05-24[9], *Implementation of Homeland Security Presidential Directive (HSPD) 12 – Policy for a Common Identification Standard for Federal Employees and*

[6] All NIST Special Publications are available at http://csrc.nist.gov/publications/PubsSPs.html.
[7] OMB Memorandum M-11-11 is available at http://www.whitehouse.gov/sites/default/files/omb/memoranda/2011/m11-11.pdf.
[8] The *Federal Identity, Credential and Access Roadmap and Implementation Guidance (FICAM)* is available at http://www.idmanagement.gov/sites/default/files/documents/FICAM_Roadmap_and_Implementation_Guidance_v2%200_20111202_0.pdf.
[9] OMB Memorandum M-05-24 is available at http://www.whitehouse.gov/sites/default/files/omb/memoranda/fy2005/m05-24.pdf.

Contractors, CMS must conduct a background investigation, adjudicate the results, and issue identity credentials to their *employees* and *contractors* who require long-term (defined as greater than 6-months) *Routine Access*[10] to Federally controlled facilities and/or information systems.

3.1.2.1 CMS/FEDERAL EMPLOYEE

Which CMS employees need PIV cards?

- Any CMS (Federal) employee, as defined in Title 5 U.S.C. § 2105, *Employee*[11], within a department or agency.
- Other federally employed individuals employed by, detailed to, or assigned to CMS.

Does not apply to:

- Occasional visitors to CMS or contractor facilities to whom you would issue temporary identification.

3.1.2.2 CMS CONTRACTOR

Which contractors need PIV cards?

- Individual under contract or subcontract to CMS, requiring long-term (defined as greater than 6-months) routine access to federally controlled facilities and/or federally controlled information systems.
- Individuals under contract or subcontract to CMS requiring any amount of unsupervised logical access. (The PIV credentialing requirements apply whether the contractor accesses the information system from the premises of a CMS facility, from their own facility, through the Internet, or by any other networked means.)

Does not apply to:

- Contractors who do not need physical or logical access, but need temporary and/or intermittent (supervised) access to CMS facilities or information systems will be treated as visitors and issued alternate credentials. This group includes temporary and seasonal workers, and those needing intermittent physical access such as delivery services.

[10] *Routine Access* is defined as regularly scheduled access. For example, a contractor who accesses CMS assets on a regular basis in the performance of ongoing responsibilities has routine access and a personnel investigation must be conducted. A contractor who is summoned for an emergency service call is not required to have a personnel investigation and is treated as a visitor. Contractors who require regularly scheduled access to one or more CMS-controlled assets, even under multiple contracts, should be treated as having routine access.

[11] The definition of "employee" as defined by Title 5 U.S.C. § 2105 can be found at http://uscode.house.gov/uscode-cgi/fastweb.exe?getdoc+uscview+t05t08+172+0++%275%20USC%20Sec.%202105%27.

3.1.2.3 VISITOR OR TEMPORARY

Visitors

- Visitor passes are issued for physical access only.
- Visitor passes are issued on the day of use, solely for same-day use.
- Visitor passes expire at the end of the day.

Temporary Credentials

If an employee or long-term contractor forgets their PIV card on a particular day, or if the person is waiting for a replacement PIV card, they may be issued a temporary badge after their identity has been confirmed.

At a minimum, the Federal Bureau of Investigation (FBI) fingerprint check portion of a National Agency Check and Inquiries (NACI) must be completed prior to issuance of any PIV credential. However, temporary credentials may be issued to new employees and contractors pending the results of the FBI fingerprint check. The temporary credentials will allow limited physical access to CMS facilities and limited logical access to CMS information systems.

3.1.2.4 FEDERALLY CONTROLLED FACILITIES

Federally Controlled Facilities[12] are defined as:

- Federally-owned buildings or leased space, whether for single or multi-tenant occupancy, and its grounds and approaches, all or any portion of which is under the jurisdiction, custody or control of a department or agency;
- Federally-controlled commercial space shared with non-government tenants. For example, if a department or agency leased the 10th floor of a commercial building, the Directive applies to the 10th floor only;
- Government-owned, contractor-operated facilities, including laboratories engaged in national defense research and production activities; and
- Facilities under a management and operating contract, such as for the operation, maintenance, or support of a Government-owned or Government-controlled research, development, special production, or testing establishment.

The following are *not* Federally Controlled Facilities:

- Contractor owned/contractor operated facilities that provide goods and/or services to CMS under contract.

[12] Pursuant to 48 CFR 2.101 (Title 48, Federal Acquisition Regulations System; Chapter 1, Federal Acquisition Regulation; Subchapter A, General; Part 2, Definitions of Words and Terms; Subpart 2.1, Definitions), available at https://www.acquisition.gov/far/current/html/Subpart%202_1.html#wp1145507.

3.1.2.5 FEDERALLY CONTROLLED INFORMATION SYSTEMS

Federally Controlled Information Systems are defined[13] as information technology systems (or information systems[14]) used or operated by CMS or by a CMS contractor or other organization on behalf of CMS.

HSPD-12 does not apply to identification associated with *National Security Systems* as defined by FISMA (44 U.S.C. § 3542(2)(A)). *As of the date of this publication, CMS does not have systems that qualify under this definition.* Contact the Enterprise Information Security Group (EISG) at mailto:ciso@cms.hhs.gov for questions concerning CMS systems suspected of meeting this definition.

3.2 E-AUTHENTICATION

In accordance with OMB Memorandum 04-04, dated December 16, 2003, *E-authentication Guidelines for Federal Agencies*[15], e-authentication is the process of establishing confidence in human user identities electronically presented to an information system. E-authentication requirements apply to *remote* authentication of human users of Federal agency IT systems for the purposes of electronically conducting government business transactions[16] (or e-government) over *"untrusted"* networks (i.e., the Internet). While that authentication typically involves a computer or other electronic device, this guidance does *not* apply to the authentication of servers, or other machines and network components. Non-human (machine) authentication is addressed in Section 4, *Machine-to-Machine Authentication*.

The e-authentication levels (1 through 4) are *only* applicable to those users that are i) accessing *"remotely"*, and ii) accessing over an *"untrusted"* network (i.e., the Internet.) The required authentication level (1 through 4) is contingent on the type of information being accessed and the risk associated with a breach or disclosure of such data. NIST SP 800-63 provides a detailed description of the assessment process required to assign the appropriate e-authentication levels for various data types. The CMS Chief Information Security Officer (CISO) has completed that assessment against each of the known CMS data types as defined by FIPS 199, and the results detailed in Table 5, *CMS Information Types & E-authentication Level Determination*. ***However***, PIV is still required for *any* logical access (even *"remote"*; *"trusted connection or not"*) for *all* PIV holders access where an identity assertion is required. As such, OMB M-04-04 e-authentication requirements only apply for i) *"remote"* human-user authentication over

[13] Pursuant to 48 CFR 2.101 (Title 48, Federal Acquisition Regulations System; Chapter 1, Federal Acquisition Regulation; Subchapter A, General; Part 2, Definitions of Words and Terms; Subpart 2.1, Definitions), available at https://www.acquisition.gov/far/current/html/Subpart%202_1.html#wp1145507.

[14] In FISMA (44 U.S.C. § 3502(8)) the term *information system* means a discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information.

[15] OMB Memorandum M-04-04 is available at http://www.whitehouse.gov/sites/default/files/omb/memoranda/fy04/m04-04.pdf.

[16] For the purposes of this document, a *transaction* is defined as: a discrete event between user and systems that supports a business or programmatic purpose.

"untrusted" networks (i.e., the Internet) and ii) those personnel (non-Federal employees and non-contractors) that are *not* covered under HSPD-12 PIV requirements.

There are a variety of terms and definitions used in this document to describe the authentication process. Refer to Table 1 to establish how these terms are used and defined in this document.

Table 1 E-authentication Terms, Abbreviations, and Definitions

Term or Abbreviation	Definitions
Active Attack	An attack on the authentication protocol where the Attacker transmits data to the Claimant, Credential Service Provider (CSP), Verifier, or Relying Party (RP). 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 process of registration and identity proofing.
Assertion	A statement from a Verifier to a RP 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 by an unauthorized individual to fool a Verifier or a RP into believing that the unauthorized individual in question is the Subscriber.
Attacker	A party who acts with malicious intent to compromise an information system.
Attribute	A claim of a named quality or characteristic inherent in or ascribed to someone or something. (See term in [FICAM] for more information.)
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 possession and 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.

Term or Abbreviation	Definitions
Authentication Protocol Run	An exchange of messages between a Claimant and a Verifier that results in authentication (or authentication failure) between the two parties.
Authentication Secret	A generic term for any secret value that could be used by an Attacker to impersonate the Subscriber in an authentication protocol. These are further divided into short-term authentication secrets, which are only useful to an Attacker for a limited period of time, and long-term authentication secrets, which allow an Attacker to impersonate the Subscriber until they are manually reset. The token secret is the canonical example of a long-term authentication secret, while the token authenticator, if it is different from the token secret, is usually a short-term authentication secret.
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 RP has to assume that the assertion was issued to the Subscriber who presents the assertion or the corresponding assertion reference to the RP.
Bit	A binary digit: 0 or 1.
Biometrics	Automated recognition of individuals based on their behavioral and biological characteristics (e.g., fingerprint). 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 [Internet Engineering Task Force (IETF) Standard Track Request for Comments (RFC) 5280[17]].
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 asserted identity is to be verified using an authentication protocol.
Claimed Address	The physical location asserted by an individual (e.g. an Applicant) where he/she can be reached. It includes the residential street address of an individual and may also include the mailing address of the individual. For example, a person with a foreign passport, living in the U.S., will need to give an address when going through the identity proofing process. This address would not be an "address of record" but a "claimed address."

[17] RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,* is available at http://tools.ietf.org/html/rfc5280.

Term or Abbreviation	Definitions
Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA)	An interactive feature added to web-forms to distinguish use of the form by humans as opposed to automated agents. Typically, it requires entering text corresponding to a distorted image or from a sound stream.
Cookie	A character string, placed in a web browser's memory, which is 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.
Credential	An object or data structure that authoritatively binds an identity (and optionally, additional attributes) to a token possessed and controlled by a Subscriber. While common usage often assumes that the credential is maintained by the Subscriber, this document also uses the term to refer to electronic records maintained by the CSP, which establish a binding between the Subscriber's token and identity.
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 (RAs) 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 RP and connected through a secure session, browses to an Attacker's website which causes the Subscriber to unknowingly invoke unwanted actions at the RP. For example, if a bank website is vulnerable to a CSRF attack, it may be possible for a Subscriber to unintentionally authorize a large money transfer, merely by viewing a malicious link in a webmail message while a connection to the bank is open in another browser window.
Cross Site Scripting (XSS)	A vulnerability that allows attackers to inject malicious code into an otherwise benign website. These scripts acquire the permissions of scripts generated by the target website and can therefore compromise the confidentiality and integrity of data transfers between the website and client. Websites are vulnerable if they display user-supplied data from requests or forms without sanitizing the data so that it is not executable.
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, *Recommendation for Key Management – Part 1: General*. 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.
Derived Credential	A credential issued based on proof of possession and control of a token associated with a previously issued credential, so as not to duplicate the identity proofing process.
Digital Signature	An asymmetric key operation where the private key is used to digitally-sign data and the public key is used to verify the signature. Digital signatures provide authenticity protection, integrity protection, and non-repudiation.

Term or Abbreviation	Definitions
Eavesdropping Attack	An attack in which an Attacker listens passively to the authentication protocol to capture information that 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.
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.
Extensible Mark-up Language (XML)	Extensible Markup Language (XML) describes a class of data objects called XML documents and partially describes the behavior of computer programs that process them.
Federal Bridge Certification Authority (FBCA)	The entity operated by the Federal Public Key Infrastructure (FPKI) Management Authority that is authorized by the Federal Public Key Infrastructure (PKI) Policy Authority to create, sign, and issue public key certificates to Principal Certification Authorities.
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.
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: 1. (One-way) It is computationally infeasible to find any input that maps to any pre-specified output, and 2. (Collision resistant) It is computationally infeasible to find any two distinct inputs that map to the same output. (NOTE: SHA-1 has been deprecated because it has proven to not be collision resistant.)
Holder-of-Key Assertion	An assertion that contains a reference to a symmetric key or a public key (corresponding to a private key) held by the Subscriber. The RP may authenticate the Subscriber by verifying that he or she can indeed prove possession and control of the referenced key.
Identity	A set of attributes that uniquely describe a person within a given context.
Identity Proofing	The process by which a Credential Service Provider and a Registration Authority collect and verify information about a person for the purpose of issuing credentials to that person.
Kerberos	A widely used authentication protocol developed at Massachusetts Institute of Technology (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. Longer password length and complexity provide some mitigation to this vulnerability, although sufficiently long passwords tend to be cumbersome for users.

Term or Abbreviation	Definitions
Knowledge Based Authentication	Authentication of an individual based on knowledge of information associated with his or her asserted identity in public databases. Knowledge of such information is considered to be private rather than secret, because it may be used in contexts other than authentication to a Verifier, thereby reducing the overall assurance associated with the authentication process.
Man-in-the-Middle Attack (MitM)	An attack on the authentication protocol run, in which the Attacker positions himself or herself in between the Claimant and Verifier so that he/she 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. MACs provide authenticity and integrity protection, but not non-repudiation protection.
Min-Entropy	A measure of the difficulty that an Attacker has to guess the most commonly chosen password used in a system. 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).
Multi-Factor	A characteristic of an authentication system or a token that uses more than one authentication factor. The three types of authentication factors are something you know, something you have, and something you are.
Network	An open communications medium, typically the Internet, which 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 (i.e., impersonation, man-in-the-middle, session hijacking) and passive (i.e., eavesdropping) attack at any point between the parties (e.g., Claimant, Verifier, CSP or RP).
Nonce	A value used in security protocols that is never repeated with the same key. For example, nonces used as challenges in challenge-response authentication protocols must not be repeated until authentication keys are changed. Otherwise, there is a possibility of a replay attack. Using a nonce as a challenge is a different requirement from 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.
Online 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.
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.

Term or Abbreviation	Definitions
Personal Identification Number (PIN)	A password consisting only of decimal digits.
Personal Identity Verification (PIV) Card	Defined by FIPS 201, *Personal Identity Verification (PIV) of Federal Employees and Contractors*, as a physical artifact (e.g., identity card, smart card) issued to federal employees and contractors that contains stored credentials (e.g., photograph, cryptographic keys, digitized fingerprint representation) so that the asserted 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).
Personally Identifiable Information (PII)	Defined by Government Accountability Office (GAO) Report 08-536, *Alternatives Exist for Enhancing Protection of Personally Identifiable Information*, as *"Any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual's identity, such as name, social security number, date and place of birth, mother's maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information."*
Pharming	An attack in which an Attacker corrupts an infrastructure service such as Domain Name Service (DNS) causing the Subscriber to be misdirected to a forged Verifier/RP, which could cause the Subscriber to reveal sensitive information, download harmful software or contribute to a fraudulent act.
Phishing	An attack in which the Subscriber is lured (usually through an email) to interact with a counterfeit Verifier/RP and tricked into revealing information that can be used to masquerade as that Subscriber to the real Verifier/RP.
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 the parties to an authentication process (i.e., RA, CSP, or Verifier). It usually describes the policies and practices of the parties and can become legally binding.
Private Credentials	Credentials that cannot be disclosed by the CSP because the contents can be used to compromise the token.
Private Key	The secret part of an asymmetric key pair that is used to digitally sign or decrypt data.
Protected Session	A session wherein messages between two participants are encrypted and integrity is protected using a set of shared secrets called session keys. A participant is said to be *authenticated* if, during the session, he, she, or it proves possession of a long-term token in addition to the session keys, and if the other party can verify the identity associated with that token. If both participants are authenticated, the protected session is said to be *mutually authenticated*.
Pseudonym	A false or fictitious name. In this document, all *unverified* names are assumed to be pseudonyms.
Public Credentials	Credentials that describe the binding in a way that does not compromise the token.
Public Key	The public part of an asymmetric key pair that is used to verify signatures or encrypt data.

Term or Abbreviation	Definitions
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 5280].
Public Key Infrastructure (PKI)	A set of policies, processes, server platforms, software and workstations used for the purpose of administering certificates and public-private key pairs, including the ability to issue, maintain, and revoke public key certificates.
Registration	The process through which an Applicant applies to become a Subscriber of a CSP and a Registration Authority (RA) validates the identity of the Applicant on behalf of the CSP.
Registration Authority (RA)	A trusted entity that establishes and vouches for the identity or attributes 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 (RP)	An entity that relies upon the Subscriber's token and credentials or a Verifier's assertion of a Claimant's identity, typically to process a transaction or grant access to information or a system.
Remote	(As in remote authentication or remote transaction) An information exchange between network-connected devices where the information cannot be reliably protected end-to-end by a single organization's security controls. Note: Any information exchange across the Internet is considered remote.
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.
Risk Assessment	The process of identifying the risks to system security and determining the probability of occurrence, the resulting impact, and additional safeguards that would mitigate this impact. Part of Risk Management and synonymous with Risk Analysis.
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.
Secondary Authenticator	A temporary secret, issued by the Verifier to a successfully authenticated Subscriber as part of an assertion protocol. This secret is subsequently used, by the Subscriber, to authenticate to the RP. Examples of secondary authenticators include bearer assertions, assertion references, and Kerberos session keys.
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 Markup Language (SAML)	An XML-based security specification developed by the Organization for the Advancement of Structured Information Standards (OASIS) for exchanging authentication (and authorization) information between trusted entities over the Internet.
SAML Authentication Assertion	A SAML assertion that conveys information from a Verifier to an RP about a successful act of authentication that took place between the Verifier and a Subscriber.

Term or Abbreviation	Definitions
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. Sessions between the Claimant and the RP can also be similarly compromised.
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.
Strongly Bound Credentials	Credentials that describe the binding between a user and token in a tamper-evident fashion.
Subscriber	A party who receives 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 cryptographic module or password) that is used to authenticate the Claimant's identity.
Token Authenticator	The output value generated by a token. The ability to generate valid token authenticators on demand proves that the Claimant possesses and controls the token. Protocol messages sent to the Verifier are dependent upon the token authenticator, but they may or may not explicitly contain it.
Token Secret	The secret value, contained within a token, which is used to derive token authenticators.
Transport Layer Security (TLS)	An authentication and security protocol widely implemented in browsers and web servers. TLS is defined by [RFC 2246][18], [RFC 3546][19], and [RFC 5246][20]. TLS is similar to the older Secure Socket 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.
Trust Anchor	A public or symmetric key that is trusted because it is directly built into hardware or software, or securely provisioned via out-of-band means, rather than because it is vouched for by another trusted entity (e.g. in a public key certificate).
Unverified Name	A Subscriber name that is not verified as meaningful by identity proofing.
User	May be referred to as the Applicant, Subscriber, or Claimant, depending on the stage in the lifecycle of the credential.
Valid	In reference to an ID, the quality of not being expired or revoked.
Verified Name	A Subscriber name that has been verified by identity proofing.

[18] RFC 2246, *The TLS Protocol Version 1.0,* is available at http://tools.ietf.org/html/rfc2246.
[19] RFC 3546, *Transport Layer Security (TLS) Extensions*, is available at http://tools.ietf.org/html/rfc3546.
[20] RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2,* is available at http://tools.ietf.org/html/rfc5246.

Term or Abbreviation	Definitions
Verifier	An entity that verifies the Claimant's identity by verifying the Claimant's possession and control 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 a Claimant to the real Verifier.
Weakly Bound Credentials	Credentials that describe the binding between a user and token in a manner than can be modified without invalidating the credential.
Zeroize	Overwrite a memory location with data consisting entirely of bits with the value zero so that the data is destroyed and not recoverable. This is often contrasted with deletion methods that merely destroy reference to data within a file system rather than the data itself.
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 Encrypted Key Exchange (EKE), Simple Password Exponential Key Exchange (SPEKE), and Secure Remote Password (SRP).

3.2.1 E-AUTHENTICATION ASSURANCE LEVELS

E-authentication presents a technical challenge when the process involves remote authentication of individual people over a network for the purpose of electronic government and commerce. The NIST SP 800-63, *Electronic Authentication Guideline,* provides technical guidance to agencies (as directed by OMB M-04-04) to allow an individual person to remotely authenticate his/her identity to a Federal IT system. NIST SP 800-63 addresses only traditional, widely implemented methods for remote authentication based on secrets. With these methods, the individual to be authenticated proves that they *"know"*, *"are"*, or *"have"* some verifiable attribute.

NIST SP 800-63 and OMB M-04-04 define four (4) assurance levels of authentication (i.e., assurance levels 1−4) required by all Federal "agencies" for electronic government transactions.

The OMB and NIST define the required level of authentication assurance (i.e., e-authentication level) in terms of the likely consequences of an authentication error. Each assurance level describes the degree of certainty that the user has presented an identifier (i.e., a credential[21]) that refers to his/her identity. In this context, assurance is defined as: i) the degree of confidence in the vetting process (proofing) used to establish the identity of the individual to whom the credential was issued, and ii) the degree of confidence that the individual who uses the credential is the individual to whom the credential was issued.

Table 2 lists the four (4) OMB e-authentication assurance levels and describes their degree of authentication confidence. Note that Table 2 is not to be interpreted as "criteria" for which level is appropriate, but describes "definitions" of what each level provides.

[21] A credential is defined as: an object that is verified when presented to the verifier in an authentication transaction.

Table 2 E-authentication Assurance Level Definitions

E-authentication Assurance Level	Definition
Level 1	Little or no confidence in the asserted identity's validity.
Level 2	Some confidence in the asserted identity's validity.
Level 3	High confidence in the asserted identity's validity.
Level 4	Very high confidence in the asserted identity's validity.

3.2.1.1 E-AUTHENTICATION LEVEL 1

For Level 1 e-authentication transactions, we expect that little or no confidence exists in the asserted identity. In fact, for Level 1 e-authentication, names in credentials *and* assertions are ***assumed to be pseudonyms***. Level 1 credentials might be utilized to allow people to personalize items on a web page for future reference, or "subscribe" to public information via a public newsletter. In these instances, the submission of forms by individuals in an electronic transaction will be a Level 1 transaction: i) when all information is flowing *to* the Federal organization *from* the individual, ii) there is *no release of information in return*, and iii) the criteria for higher assurance levels are not triggered. For example, if an individual applies to CMS for receipt of a monthly public newsletter about specific Medicare services, the transaction, with CMS *and* the individual, would present minimal risks and could be treated as Level 1. There is no need to collect PII (or other sensitive information), and little risk if the provided information (email address or account ID) is accessed inappropriately. So, for e-authentication Level 1, we require the following at each stage of the e-authentication process:

1. **Initial *proofing* prior to issuing the required token**: No proofing is required. Names (and individuals) in credentials *and* assertions are ***assumed to be pseudonyms***.
2. **Subsequent session *authentications* (i.e., log-on sessions)**: Simply verify that the credential token matches what was issued to the entity. Now, we *still* assume that the names (and individuals) in credentials *and* assertions are *pseudonyms*—but at least we are dealing with the same individual (*whoever* they are.)

Although there is no identity-proofing requirement at level 1, the authentication mechanism does provide some assurance that the *same* claimant who participated in *previous* transactions is accessing the Level 1 protected data in *subsequent* transactions. It allows a wide range of available authentication technologies to be employed and permits the use 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 eavesdroppers. 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. Assertions and assertion references require protection from manufacture/modification and reuse attacks.

3.2.1.2 E-AUTHENTICATION LEVEL 2

For Level 2 e-authentication transactions, confidence exists that the asserted identity is *accurate*. Level 2 credentials are appropriate for a wide range of business with the public, where CMS requires an initial identity assertion (the details of which are *verified* prior to any CMS action). For example, a beneficiary changes their address of record through the CMS website. The site needs authentication to ensure that the Security and Privacy Controls for Federal Information Systems and Organizations person's address is changed. This transaction involves a *low* risk of inconvenience. However, since official notices regarding payment amounts, account status, and records of changes are sent to the beneficiary's address of record, it entails *moderate* risk of unauthorized release of personally sensitive data. CMS (using OMB and NIST standards) determines that the risk of unauthorized release merits e-authentication Level 2 authentication because the risk to the individual being affective negatively is significant. While the mechanisms for *subsequent authentication* into an account (User ID and Password) are similar to Level 1, the initial *proofing* requirements are *more rigorous* in order to establish the *identity* of the beneficiary, and their rights to be *initially* issued the token. So, while Level 2 provides single-factor remote network authentication similar to Level 1, at Level 2, identity-*proofing* requirements are introduced, requiring presentation of identifying materials or information. That is, we can *no longer assume* that names in the assertions are *pseudonyms* (e.g., Level 1). So, for e-authentication Level 2, we require the following at each stage of the e-authentication process:

1. **Initial *proofing* prior to issuing the required token**: Must verify that the *asserted* identity corresponds to a *real* individual, ***and*** that the asserted identity i*s on the other end of the proofing transaction*—then (an only then) we can issue the appropriate credential token.
2. **Subsequent session *authentications* (i.e., log-on sessions)**: Verify that the appropriate credential token matches what was issued to the entity. *Now,* and *only* now, we can *also* assume that the individual who is represented by the asserted identity *is actually* performing the transaction.

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 permits 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 resisted. Protocols are also required to be at least weakly-resistant to man-in-the middle attacks.

Long-term shared authentication secrets, if used, are never revealed to any party except verifiers operated by the *Credentials Service Provider (CSP)*[22]; however, session (temporary) shared

[22] NIST SP 800-63-1 defines a *Credentials Service Provider (CSP)* as 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.

secrets may be provided to independent verifiers by the CSP. In addition to Level 1 requirements, assertions are resistant to disclosure, redirection, capture, and substitution attacks. Approved cryptographic techniques (i.e., NIST-compliant) are required for all assertion protocols used at Level 2 and above.

3.2.1.3 E-AUTHENTICATION LEVEL 3

Level 3 e-authentication is appropriate for transactions needing *high* confidence in the asserted identity's accuracy. People may use Level 3 credentials to access restricted web services without the need for additional identity assertion controls. For example, a Medicare *provider* accesses a CMS payment website to access one or more beneficiaries' claims data, and make necessary adjustments to beneficiary claims data. Since the provider may be able to access and update claims data for many beneficiaries, and the risk of a compromise of the account could be high, Level 3 e-authentication requires a more rigorous level of *initial proofing* and *subsequent authentication* for each transaction. So, for e-authentication Level 3, we require the following at each stage of the e-authentication process:

1. **Initial *proofing* prior to issuing the required token**: Must verify that the *asserted* identity corresponds to a *real* individual, *and* that the asserted identity i*s on the other end of the proofing transaction*, verified to a *higher level* than Level 2—then (an only then) we can issue the appropriate credential token(s).
2. **Subsequent session *authentications* (i.e., log-on sessions)**: Verify that *multiple* credential tokens (i.e., multi-factor) match what was issued to the entity. *Now*, and *only* now, we can *also* assume that the individual who is represented by the asserted identity *is actually* performing the transaction.

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 e-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 permits any of the token methods of Level 4. Level 3 e-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.

Authentication requires that the claimant prove, through a secure authentication protocol, that he or she controls the token. The claimant unlocks the token with a password or biometric, or uses 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 CSP; however, session (temporary) shared secrets may be provided to independent verifiers by the CSP. In addition to Level 2 requirements, assertions are protected against repudiation by the verifier.

3.2.1.4 E-AUTHENTICATION LEVEL 4

Level 4 e-authentication is appropriate for transactions needing *very high* confidence in the asserted identity's accuracy. Users may present Level 4 credentials to assert identity and gain access to highly restricted web resources, without the need for further identity assertion controls. For example, a CMS equipment vendor uses a remote system giving access to apply vendor updates to CMS equipment firmware. Using their corporate equipment, they access CMS infrastructure hardware over the Internet via various connections. The sensitive *information* they may access creates only a *moderate* potential impact for unauthorized access, but the inherent risk in Internet access raises the overall risk to *high*. So, for e-authentication Level 4, we require the following at each stage of the e-authentication process:

1. **Initial *proofing* prior to issuing the required token**: Must verify that the *asserted* identity corresponds to a *real* individual, *and* that the asserted identity i*s on the other end of the proofing transaction,* verified to a *higher level* than Level 3—then (an only then) we can issue the appropriate credential tokens—with at least one token being a cryptographic-based (FIPS 140-2) hardware token.

2. **Subsequent session *authentications* (i.e., log-on sessions)**: Verify that the appropriate multiple credential tokens (including the cryptographic hardware token) match what was issued to the entity. *Now,* and *only* now, we can *also* assume that the individual who is represented by the asserted identity *is actually* performing the transaction.

Level 4 is intended to provide the highest practical remote network authentication assurance. Level 4 e-authentication is based on proof of possession of a key through a cryptographic protocol. At this level, in-person identity proofing is required. Level 4 is similar to Level 3 (*multi-factor*) except that only "hard" *cryptographic* tokens are allowed. The token is required to 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. Level 4 token requirements may be met by using the PIV authentication key of a FIPS 201 compliant PIV Card.

Level 4 requires strong cryptographic authentication of all communicating 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 are required to 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 CSP; however, session (temporary) shared secrets may be provided to independent verifiers by the CSP. Approved cryptographic techniques are used for all operations. All sensitive data transfers are cryptographically authenticated using keys bound to the authentication process.

3.2.2 CMS ENTERPRISE E-AUTHENTICATION RISK ASSESSMENT FOR DETERMINING E-AUTHENTICATION LEVELS

The e-authentication assurance level is determined by assessing the potential risks to CMS and by identifying measures to minimize their impact. The risks from an authentication error are a function of two (2) factors: i) potential harm or impact, and ii) the likelihood of such harm or impact, as they apply to six (6) OMB-defined potential impact categories. The potential impact for each of the potential impact categories is assessed using the potential impact values described in FIPS 199 (i.e., High, Moderate, or Low).

Table 3 presents the six (6) OMB potential impact categories for authentication errors and their respective potential impact values.

Table 3 Potential Impact Categories and Potential Impact Values

Level	Potential impact of *"inconvenience, distress or damage to standing or reputation"*
Low	At worst, limited, short-term inconvenience, distress or embarrassment to any party.
Moderate	At worst, serious short term or limited long-term inconvenience, distress or damage to the standing or reputation of any party.
High	Severe or serious long-term inconvenience, distress, or damage to the standing or reputation of any party (ordinarily reserved for situations with particularly severe effects or which affect many individuals).

Level	Potential impact of *"financial loss"*
Low	At worst, an insignificant or inconsequential unrecoverable financial loss to any party, or at worst, an insignificant or inconsequential agency liability.
Moderate	At worst, a serious unrecoverable financial loss to any party, or a serious agency liability.
High	Severe or catastrophic unrecoverable financial loss to any party; or severe or catastrophic agency liability.

Level	Potential impact of *"harm to agency programs or public interests"*
Low	At worst, a limited adverse effect on organizational operations or assets, or public interests. Examples of limited adverse effects are: i) mission capability degradation to the extent and duration that the organization is able to perform its primary functions with noticeably reduced effectiveness, or ii) minor damage to organizational assets or public interests.
Moderate	At worst, a serious adverse effect on organizational operations or assets, or public interests. Examples of serious adverse effects are: i) significant mission capability degradation to the extent and duration that the organization is able to perform its primary functions with significantly reduced effectiveness; or ii) significant damage to organizational assets or public interests.
High	A severe or catastrophic adverse effect on organizational operations or assets, or public interests. Examples of severe or catastrophic effects are: i) severe mission capability degradation or loss of to the extent and duration that the organization is unable to perform one or more of its primary functions; or ii) major damage to organizational assets or public interests.

Level	Potential impact of "unauthorized release of sensitive information"
Low	At worst, a limited release of personal, U.S. government-sensitive, or commercially sensitive information to unauthorized parties resulting in a loss of confidentiality with a low impact as defined in FIPS 199.
Moderate	At worst, a release of personal, U.S. government-sensitive, or commercially sensitive information to unauthorized parties resulting in loss of confidentiality with a moderate impact as defined in FIPS 199.
High	A release of personal, U.S. government-sensitive or commercially-sensitive information to unauthorized parties resulting in loss of confidentiality with a high impact as defined in FIPS 199.

Level	Potential impact of *"personal safety"*
Low	At worst, minor injury not requiring medical treatment.
Moderate	At worst, moderate risk of minor injury or limited risk of injury requiring medical treatment.
High	A risk of serious injury or death.

Level	Potential impact of "civil or criminal violations"
Low	At worst, a risk of civil or criminal violations of a nature that would not ordinarily be subject to enforcement efforts.
Moderate	At worst, a risk of civil or criminal violations that may be subject to enforcement efforts.
High	A risk of civil or criminal violations that are of special importance to enforcement programs.

The assurance level is determined by comparing the potential impact category to the potential impact value associated with each assurance level, as shown in Table 4. The required assurance level is determined by locating the highest level whose impact profile meets or exceeds the potential impact for every impact category.

Table 4 Maximum Assurance Level for each Potential Impact Category

	Assurance Level Impact Profiles			
Potential Impact Categories	**1**	**2**	**3**	**4**
Inconvenience, distress or damage to standing or reputation	Low	Mod	Mod	High
Financial loss or agency liability	Low	Mod	Mod	High
Harm to agency programs or public interests	N/A	Low	Mod	High
Unauthorized release of sensitive information	N/A	Low	Mod	High
Personal Safety	N/A	N/A	Low	Mod/High
Civil or criminal violations	N/A	Low	Mod	High

Using the CMS-defined information types and the OMB four (4) e-authentication assurance levels, the CMS CISO performed an enterprise risk assessment determination (in accordance with NIST SP 800-63) to determine which assurance level impact profile applies to each potential security impact category (based on the CMS information type). The results of these determinations are included in Table 5. The basis for determining the overall e-authentication assurance level for each information type is based on selecting the *highest* applicable impact level for each information type (refer to the bolded, highlighted levels in Table 5). Note that, for the purposes of e-authentication, the authentication requirements apply to users *accessing* the applicable data described. If the system does not (or cannot) present the described information to the user, then that category does not apply, even though the data may exist within the system.

If a Business Owner does not agree that the information type processed by their information system requires the same e-authentication authorization level stated in Table 5, they must use the information provided in Table 5 to demonstrate and explain why the assurance level should be different. To demonstrate this, the Business Owner must conduct an e-authentication Risk Assessment on their system to determine the e-authentication assurance level and submit the completed assessment results to the EISG. The explanation, in accordance with Table 5, and the reasons for modifying the e-authentication assurance level must also be included in the applicable system risk assessment.

Using the e-authentication assurance level published in Table 5 or the appropriate assurance level approved by the CMS CISO, the Business Owner uses the information provided in Sections 3.2.3 through 3.2.13 to apply the necessary control requirements to their information system.

Table 5 CMS Information Types & E-authentication Level Determination

Information Type	**Explanation and Examples**	**E-authentication Level**			
Investigation, intelligence-related, and security information (14 CFR PART 191.5(D))	Information related to investigations for law enforcement purposes; intelligence-related information that cannot be classified, but is subject to confidentiality and extra security controls. Includes security plans, contingency plans, emergency operations plans, incident reports, reports of investigations, risk or vulnerability assessments certification reports; does not include general plans, policies, or requirements.	**Level 4**			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	Mod	**<Mod>**	High
	Financial loss or agency liability	Low	Mod	**<Mod>**	High
	Harm to agency programs or public interests	N/A	Low	**<Mod>**	High
	Unauthorized release of sensitive information	N/A	Low	Mod	**<High>**
	Personal safety	N/A	**<N/A>**	Low	Mod/High
	Civil or criminal violations	N/A	Low	Mod	**<High>**

Information Type	Explanation and Examples	E-authentication Level			
Mission-critical information	Information and associated infrastructure directly involved in making payments for Medicare Fee-for-Service (FFS), Medicaid and State Children's Health Insurance Program (SCHIP).	**Level 4**			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	Mod	Mod	**<High>**
	Financial loss or agency liability	Low	Mod	**<Mod>**	High
	Harm to agency programs or public interests	N/A	Low	**<Mod>**	High
	Unauthorized release of sensitive information	N/A	Low	Mod	**<High>**
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	N/A	Low	**<Mod>**	High

Information Type	Explanation and Examples	E-authentication Level			
Information about persons	Information related to personnel, medical, and similar data. Includes all information covered by the Privacy Act of 1974 (e.g., salary data, social security information, passwords, user identifiers (IDs), Equal Employment Opportunity (EEO), personnel profile (including home address and phone number), medical history, employment history (general and security clearance information), and arrest/criminal investigation history as well as personally identifiable information (PII), individually identifiable information (IIF), or personal health information (PHI) covered by the Health Insurance Portability and Accountability Act of 1996 (HIPAA).	**Case 1: A user can ONLY access or update information about themselves:** **Level 2** **Case 2: A user can ONLY submit, review, or update information about persons that THEY have provided DURING THE CURRENT SESSION:** **Level 2** **Case 3: A user, not covered in Cases 1 or 2, can access or update information about persons OTHER THAN themselves:** **Level 3**			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	**<Case 1 or 2: Mod>**	**<Case 3: Mod>**	High
	Financial loss or agency liability	Low	**<Case 1 or 2: Mod>**	**<Case 3: Mod>**	High
	Harm to agency programs or public interests	N/A	**<Case 1 or 2: Low>**	**<Case 3: Mod>**	High
	Unauthorized release of sensitive information	N/A	**<Case 1 or 2: Low>**	**<Case 3: Mod>**	High
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	N/A	**<Case 1 or 2: Low>**	**<Case 3: Mod>**	High

Information Type	Explanation and Examples	E-authentication Level			
Financial, budgetary, commercial, proprietary and trade secret information	Information related to financial information and applications, commercial information received in confidence, or trade secrets (i.e., proprietary, contract bidding information, sensitive information about patents, and information protected by the Cooperative Research and Development Agreement). Also included are information about payments, payroll, automated decision making, procurement, market-sensitive, inventory, other financially-related systems, and site operating and security expenditures.	**Level 3**			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	Mod	**<Mod>**	High
	Financial loss or agency liability	Low	Mod	**<Mod>**	High
	Harm to agency programs or public interests	N/A	Low	**<Mod>**	High
	Unauthorized release of sensitive information	N/A	Low	**<Mod>**	High
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	N/A	Low	**<Mod>**	High

Information Type	Explanation and Examples	E-authentication Level			
Internal administration	Information related to the internal administration of an agency. Includes personnel rules, bargaining positions, advance information concerning procurement actions, management reporting, etc.	**Level 3**			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	**<Mod>**	Mod	High
	Financial loss or agency liability	Low	**<Mod>**	Mod	High
	Harm to agency programs or public interests	N/A	**<Low>**	Mod	High
	Unauthorized release of sensitive information	N/A	Low	**<Mod>**	High
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	N/A	**<Low>**	Mod	High

Information Type	Explanation and Examples	E-authentication Level			
Other Federal agency information	Information, the protection of which is required by statute, or which has come from another Federal agency and requires release approval by the originating agency.	Level 3			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	**<Mod>**	Mod	High
	Financial loss or agency liability	Low	**<Mod>**	Mod	High
	Harm to agency programs or public interests	N/A	**<Low>**	Mod	High
	Unauthorized release of sensitive information	N/A	Low	**<Mod>**	High
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	N/A	**<Low>**	Mod	High

Information Type	Explanation and Examples	E-authentication Level			
New technology or controlled scientific information	Information related to new technology; scientific information that is prohibited from disclosure or that may require an export license from the Department of State and/or the Department of Commerce.	Level 3			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	**<Mod>**	Mod	High
	Financial loss or agency liability	Low	**<Mod>**	Mod	High
	Harm to agency programs or public interests	N/A	**<Low>**	Mod	High
	Unauthorized release of sensitive information	N/A	Low	**<Mod>**	High
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	N/A	**<Low>**	Mod	High

Information Type	Explanation and Examples	E-authentication Level			
Operational information	Information that requires protection during operations; usually time-critical information.	Level 3			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	**<Mod>**	Mod	High
	Financial loss or agency liability	Low	**<Mod>**	Mod	High
	Harm to agency programs or public interests	N/A	Low	**<Mod>**	High
	Unauthorized release of sensitive information	N/A	Low	**<Mod>**	High
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	N/A	**<Low>**	Mod	High

Information Type	Explanation and Examples	E-authentication Level			
System configuration management information	Any information pertaining to the internal operations of a network or computer system, including but not limited to network and device addresses; system and protocol addressing schemes implemented at an agency; network management information protocols, community strings, network information packets, etc.; device and system passwords; device and system configuration information.	**Level 3**			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	Mod	**<Mod>**	High
	Financial loss or agency liability	Low	Mod	**<Mod>**	High
	Harm to agency programs or public interests	N/A	Low	**<Mod>**	High
	Unauthorized release of sensitive information	N/A	Low	**<Mod>**	High
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	N/A	Low	**<Mod>**	High

Information Type	Explanation and Examples	E-authentication Level			
Other sensitive information	Any information for which there is a management concern about its adequate protection, but which does not logically fall into any of the above categories. Use of this category should be rare.	**Level 2**			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	Low	**<Mod>**	Mod	High
	Financial loss or agency liability	Low	**<Mod>**	Mod	High
	Harm to agency programs or public interests	N/A	**<Low>**	Mod	High
	Unauthorized release of sensitive information	N/A	**<Low>**	Mod	High
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	N/A	**<Low>**	Mod	High

Information Type	Explanation and Examples	E-authentication Level			
Public information	Any information that is declared for public consumption by official authorities and has no identified requirement for integrity or availability. This includes information contained in press releases approved by the Office of Public Affairs or other official sources.	**Case 1: No tracking or control on a user-level basis is desired.** **Level 0** **(No authentication required)** **Case 2: Tracking or control on a user-level basis is desired for business purposes.** **Level 1**			
	Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
	Inconvenience, distress or damage to standing or reputation	**<Case 1: N/A> <Case 2: Low>**	Mod	Mod	High
	Financial loss or agency liability	**<Case 1: N/A> <Case 2: Low>**	Mod	Mod	High
	Harm to agency programs or public interests	**<N/A>**	Low	Mod	High
	Unauthorized release of sensitive information	**<N/A>**	Low	Mod	High
	Personal safety	**<N/A>**	N/A	Low	Mod/High
	Civil or criminal violations	**<N/A>**	Low	Mod	High

3.2.3 E-AUTHENTICATION MODEL

In accordance with OMB guidance (OMB M-04-04), e-authentication is the process of establishing confidence in user identities presented electronically to an information system. Systems can use the authenticated identity to determine whether that individual is authorized to perform an electronic transaction. In most cases, the authentications and transactions 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 CSP. If approved, the Subscriber is issued a credential by the CSP, which binds a token to an identifier (and possibly one or more attributes that the RA has verified). The token may be issued by the CSP, generated directly by the Subscriber, or provided by a third party. The CSP registers the token by creating a credential that binds the token to an identifier and possibly other attributes that the RA has verified. The token and credential may be used in subsequent authentication events.

In a common case, the RA and CSP are separate functions of the same system. However, a RA might be part of an 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 Registration Authorities, and a RA may have relationships with different CSPs as well.

The name specified in a credential may be either a *verified* name or an *unverified* name. If the RA has determined that the name is officially associated with a *real person* and the Subscriber is the person who is entitled to use that identity, the name is considered a *verified* name. If the RA has not verified the Subscriber's name, or the name is known to differ from the official name, the name is considered a *pseudonym*. The process used to verify a Subscriber's association with a name is called *identity proofing*, and is performed by an RA that registers Subscribers with the CSP.

At Level 1, identity proofing is not required, so names in credentials and assertions are *assumed* to be *pseudonyms*. At Level 2, identity proofing is required, but the credential may assert the verified name or a pseudonym. In the case of a pseudonym, the CSP shall retain the name verified during registration. Level 2 credentials and assertions shall specify whether the name is a verified name or a pseudonym. This information assists Relying Parties (RPs) in making access control or authorization decisions. In most cases, only verified names may be specified in credentials and assertions at Levels 3 and 4. (The required use of a *verified* name at higher levels of assurance is derived from OMB M-04-04 and is specific to Federal IT systems, rather than a general e-authentication requirement.)

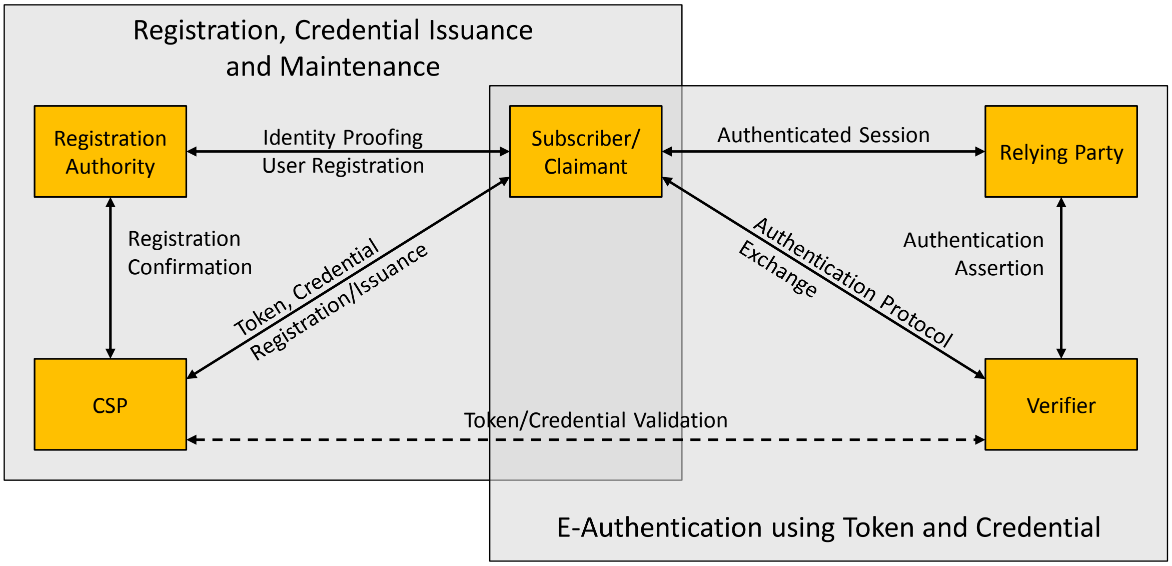
The party to be authenticated is called a *Claimant* and the party verifying that identity is called a *Verifier*. When a Claimant successfully demonstrates possession and control of a token to a Verifier through an authentication protocol, the Verifier can verify that the Claimant is the Subscriber named in the corresponding credential. 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 RP, the assertion may be implicit. The RP can use the authenticated information provided by the Verifier to make access control or authorization decisions.

Authentication establishes confidence in the Claimant's *identity*, and in some cases in the Claimant's *personal attributes* (for example the Subscriber is a U.S. Citizen, is a student at a particular university, or is assigned a particular number or code by an agency or organization). Authentication does not determine the Claimant's authorizations or access privileges; this is a separate decision. RPs (e.g., government agencies) will use a Subscriber's authenticated identity and attributes with other factors to make access control or authorization decisions.

As part of authentication, mechanisms such as device identity or geo-location could be used to identify or prevent possible authentication false positives. While these mechanisms do not directly increase the assurance level for authentication, they can enforce security policies and mitigate risks. In many cases, the authentication process and services will be shared by many applications and agencies. However, it is the individual agency or application acting as the RP that shall make the decision to grant access or process a transaction based on the specific application requirements.

The various entities and interactions that comprise the e-authentication model used here are illustrated below in Figure 1. The shaded box on the *left* shows the *registration*, *credential issuance, maintenance activities*, and the *interactions between the Subscriber/Claimant, the RA, and the CSP*. The usual sequence of interactions is as follows:

- An individual Applicant applies to an RA through a registration process.


Registration, Credential Issuance and Maintenance
Registration Authority
Identity Proofing
User Registration
Subscriber/ Claimant
Authenticated Session
Relying Party
Registration Confirmation
Token, Credential Registration/Issuance
Authentication Protocol Exchange
Authentication Assertion
CSP
Token/Credential Validation
Verifier
E-Authentication using Token and Credential

The shaded 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:

- The Claimant proves to the Verifier that he or she *possesses and controls* the token through an authentication protocol.
- The Verifier interacts with the CSP to *validate* the credential that binds the Subscriber's identity to his or her token.
- If the Verifier is separate from the RP (application), the Verifier provides an assertion about the Subscriber to the RP, which uses the information in the assertion to make an access control or authorization decision.
- An authenticated session is established between the Subscriber and the RP.

In some cases, the Verifier does not need to directly communicate with the CSP to complete the authentication activity (e.g., some uses of digital certificates). Therefore, the dashed line between the Verifier and the CSP represents a logical link between the two entities rather than a physical link. In some implementations, the Verifier, RP, and the CSP functions may be distributed and separated as shown in Figure 1; however, if these functions reside 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.

As noted above, CSPs maintain status information about credentials they issue. CSPs generally assign a finite lifetime when issuing credentials to limit the maintenance period. When the status changes, or when the credentials near expiration, credentials 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 may be required to repeat the registration process to obtain a new token and credential. The CSP may choose to accept a request during a grace period after expiration.

3.2.4 REGISTRATION PROCESS

In the registration process, an Applicant undergoes *identity proofing* by a trusted 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.

3.2.4.1 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 document concentrates on addressing impersonation threats. Infrastructure threats are addressed by normal computer security controls (e.g., separation of duties, record keeping, independent audits) and are outside the scope of this document.

The threats to the issuance process include impersonation attacks and threats to the transport mechanism for the token and credential issuance. Table 6 lists the threats related to registration and issuance.

Table 6 Registration and Identity Threats

Activity	Threat/Attack	Example
Registration	**Impersonation of claimed identity**	An Applicant asserts an incorrect identity by using a forged driver's license.
	Repudiation of registration	A Subscriber denies registration, claiming that he or she did not register that token.
Issuance	**Disclosure**	A key created by the CSP for a Subscriber is copied by an Attacker as it is transported from the CSP to the Subscriber during token issuance.
	Tampering	A new password created by the Subscriber is modified by an Attacker as it is being submitted to the CSP during the credential issuance phase.
	Unauthorized issuance	A person asserting to be the Subscriber (but in reality is not the Subscriber) is issued credentials for that Subscriber.

3.2.4.2 THREAT MITIGATION STRATEGIES

Registration threats can be deterred by making impersonation more difficult to accomplish or increasing the likelihood of detection. This document 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 asserted identity exists, that the Applicant is the person who is entitled to the asserted identity, and that 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. Table 7 lists strategies for mitigating threats to the registration and issuance processes.

Table 7 Registration and Issuance Threat Mitigation Strategies

Activity	Threat/Attack	Mitigation Strategy
Registration	**Impersonation of claimed identity**	RAs request documentation that provides a specified level of confidence (or assurance) in 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, and passports presented by the Applicant are often used to assert the identity of the Applicant.
		Have 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 the token in person, physically mail it in a sealed envelope to a secure location, or use a protected session to send the token electronically.
	Tampering	Issue credentials in person, 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 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.
		Implement a dual-control issuance process that ensures two independent individuals shall cooperate in order to issue a token and/or credential.

3.2.4.3 REGISTRATION AND ISSUANCE ASSURANCE LEVELS

The registration and identity proofing processes are designed based on the required 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; and
- 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. Remote registration requirements are designed to permit fully automated solutions. However, implementations may also leverage call centers, or online assistance (including using *in-person* proofing solutions) as a substitute or complement for fully automated solutions.

In some context, Business Owners 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 CMS that could help confirm the Applicant's identity. Table 8 summarizes the registration and identity proofing process.

Registration, identity proofing, token creation/issuance, and credential issuance are separate processes that can be broken up into a number of separate physical encounters or 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 processes:

- At Level 1: There is no specific requirement, however some effort should be made to uniquely identify and track applications.
- At Level 2: For electronic transactions, 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.

  For physical transactions, the Applicant shall identify himself/herself in person by either using a secret as described above, or by biometric verification (comparing a captured biometric sample to a reference biometric sample that was enrolled during a prior encounter).
- At Level 3: For electronic transactions, the Applicant shall identify himself/herself in any new electronic transaction (beyond the first transaction or encounter) by presenting a temporary secret that was established during a prior transaction or encounter, or sent to the Applicant's phone number, email address, or physical address of record. Permanent secrets shall only be issued to the applicant within a protected session.

  For physical transactions, 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. If the CSP issues permanent secrets during a physical transaction, then they shall be loaded locally onto a physical device that is issued in person to the applicant or delivered in a manner that confirms the address of record.
- At Level 4: Only physical transactions apply. The Applicant shall identify himself/herself in person in each new physical transaction through the use of a biometric that was recorded during a prior encounter.[23] If the CSP issues permanent secrets, then they shall be loaded locally onto a physical device that is issued in person or delivered in a manner that confirms the address of record.

[23] Special arrangements can be made for Applicants who are unable to provide the required biometrics.

A common reason for breaking up the registration process as described above is to allow the subscriber to register or obtain tokens for use in two or more 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.

If a valid credential has already been issued, then the CSP may issue another credential of equivalent or lower assurance. In this case, proof of possession and control of the original token may be substituted for repeating the identity proofing steps. (This is a special case of a derived credential. See NIST SP 800-63 Section 5.3.5, *Requirements for Derived Credentials*, for procedures when the derived credential is issued by a different CSP.) Any requirements for credential delivery at the appropriate Level shall still be satisfied.

3.2.4.4 REQUIREMENTS FOR ONE-TIME USE

For infrequently used applications, issuance and maintenance of credentials would be prohibitively expensive. Claimants can be authenticated for immediate one-time access to an application for Levels 1 thru 3. At Level 1, there is no requirement for identity proofing before one-time use. At Levels 2 and 3, application owners act as the RA/CSP in the remote registration processes described in Table 8, using processes that do not require confirmation of the address of record and omitting credential issuance.

For immediate one-time access at Level 2, application owners can use the registration processes specified in Table 7 that:

- Confirm "*the ability of the Applicant to receive telephone communications or text message at phone number or e-mail address associated with the Applicant in records*"; or
- Subsequently send a *"notice to an address of record confirmed in the records check."*

For immediate one-time access at Level 3, application owners can use the registration process specified in Table 8 that:

- Confirms *"the ability of the Applicant to receive telephone communications at a phone number associated with the Applicant in records, while recording the Applicant's voice or using alternative means that establish an equivalent level of non-repudiation."*

Table 8 Registration and Identity Proofing

Control	Levels of Assurance			
	Level 1	Level 2	Level 3	Level 4
1. Registration Requirements	There are no level-specific requirements at Level 1.	Both in-person and remote registration are permitted. The Applicant supplies his or her full legal name, an address of record, and date of birth (DoB), and may also supply other individual identifying information subject to CMS requirements.	Both in-person and remote registration are permitted. The Applicant supplies his or her full legal name, an address of record, and date of birth (DoB), and may also supply other individual identifying information subject to CMS requirements.	Only in-person registration is permitted. The Applicant supplies his or her full legal name, an address of record, and date of birth (DoB), and may also supply other individual identifying information subject to CMS requirements.
2. In-Person Identity Proofing Requirements				
2.1. Basis for Issuing Credentials	There are no level-specific requirements at Level 1.	Possession of a valid current primary government picture ID[24] that contains Applicant's picture, and either address of record or nationality of record (e.g., driver's license or passport).	Possession of verified current primary government picture ID that contains Applicant's picture and either address of record or nationality of record ID (e.g., driver's license or passport).	In-person appearance and verification of: 1. A current primary government picture ID that contains Applicant's picture, and either address of record or nationality of record (e.g., driver's license or passport), and; 2. Either a second, independent government ID document that contains current corroborating information (e.g., either address of record **or** nationality of record), or verification of a financial account number (e.g., checking account, savings account, loan or credit card) confirmed via records.
2.2. RA and	There are no	RA inspects photo-ID, compares	RA inspects photo-ID and verifies via	**Primary Photo ID:**

[24] The following resources offer examples of what some agencies consider to be primary or secondary ID:
U.S. Citizenship and Immigration Services (USCIS) Form I-9, *"Lists of Acceptable Documents"*, http://www.uscis.gov/files/form/i-9.pdf
Instructions for First Time Passport Applicants, http://travel.state.gov/passport/get/first/first_830.html#step4first
Secondary Evidence of Identification, http://travel.state.gov/passport/get/secondary_evidence/secondary_evidence_4314.html.

Control	Levels of Assurance			
	Level 1	Level 2	Level 3	Level 4
CSP Actions	level-specific requirements at Level 1.	picture to Applicant, and records the ID number, address, and DoB. (RA optionally reviews personal information in records to support issuance process "1" below.) If ID appears valid and photo matches Applicant, then: 1. If personal information in records includes a telephone number or e-mail address, the CSP issues credentials in a manner that confirms the ability of the Applicant to receive telephone communications or text message at phone number or e-mail address associated with the Applicant in records. Any secret sent over an unprotected session shall be reset upon first use; or 2. If ID confirms address of record, RA authorizes or CSP issues credentials. Notice is sent to address of record, or; 3. If ID does not confirm address of record, CSP issues credentials in a manner that confirms the claimed address.	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, and records the ID number. If ID is valid and photo matches Applicant then: 1. If personal information in records includes a telephone number, the CSP 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 alternative means that establish an equivalent level of nonrepudiation; or 2. If ID confirms address of record, RA authorizes or CSP issues credentials. Notice is sent to address of record, or; 3. If ID does not confirm address of record, CSP issues credentials in a manner that confirms address.	RA inspects photo-ID and verifies 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, and records ID number. **Secondary Government ID or financial account:** 1. RA inspects secondary government ID and if apparently valid, confirms that the identifying information is consistent with the primary photo-ID, or; 2. RA verifies financial account number supplied by Applicant through record checks or through credit bureaus or similar databases, and confirms that: name, DoB, address, and other personal information in records are on balance consistent with the application and sufficient to identify a unique individual. **Note: Address of record shall be confirmed through validation of either primary or secondary ID.** **Current Biometric:** RA records a current biometric (e.g., photograph or fingerprints) to ensure that Applicant cannot repudiate application. **Credential Issuance:**

Control	Levels of Assurance			
	Level 1	Level 2	Level 3	Level 4
				CSP issues credentials in a manner that confirms address of record.
3. Remote Identity Proofing Requirements				
3.1 Basis for Issuing Credentials	There are no level-specific requirements at Level 1.	Possession of a valid current government ID[25] (e.g., a driver's license or passport) number and a financial or utility account number (e.g., checking account, savings account, utility account, loan, or credit card, to tax ID) confirmed via records of **either** the government ID or account number. Note that confirmation of the financial or utility account may require supplemental information from the Applicant.	Possession of a valid current government ID (e.g., a driver's license or passport) number and a financial or utility account number (e.g., checking account, savings account, utility account, loan, or credit card) confirmed via records of **both** numbers. Note that confirmation of the financial or utility account may require supplemental information from the Applicant.	**Not Applicable**
3.2. RA and CSP Actions	There are no level-specific requirements at Level 1.	RA 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, and other personal information in records are on balance consistent with the application and sufficient to identify a unique individual. For utility account numbers, confirmation shall be performed by verifying knowledge of recent account activity. (This technique may also be applied to	RA 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. At a minimum, the records check for both the ID number and the account number should confirm the name and address of the Applicant. For utility account numbers, confirmation shall be performed by verifying knowledge of recent account activity. (This technique may also be applied to	**Not Applicable**

[25] Agencies issuing credentials to foreign nationals residing in foreign countries determine what constitutes a valid Government issued ID as required.

Control	Levels of Assurance			
	Level 1	Level 2	Level 3	Level 4
		some financial accounts.) Address/phone number confirmation and notification: 1. CSP issues credentials in a manner that confirms the ability of the Applicant to receive mail at a physical address associated with the Applicant in records; or 2. If personal information in records includes a telephone number or e-mail address, the CSP issues credentials in a manner that confirms the ability of the Applicant to receive telephone communications or text message at phone number or e-mail address associated with the Applicant in records. Any secret sent over an unprotected session shall be reset upon first use and shall have a maximum lifetime of seven days; or 3. The CSP issues credentials. RA or CSP sends notice to an address of record confirmed in the records check.[26]	some financial accounts.) Address confirmation: 1. CSP issues credentials in a manner that confirms the ability of the Applicant to receive mail at a physical address associated with the Applicant in records; or 2. If personal information in records includes both an electronic address and a physical address that are linked together with the Applicant's name, and are consistent with the information provided by the applicant, then the CSP may issue credentials in a manner that confirms ability of the Applicant to receive messages (Simple Message Service (SMS), voice or e-mail) sent to the electronic address. Any secret sent over an unprotected session shall be reset upon first use and shall have a maximum lifetime of seven days.	
4. Records Retention Requirements	There are no level-specific requirements at Level 1.	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 minimum record retention period	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 minimum record retention period	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 minimum record retention period

[26] Agencies are encouraged to use methods "1" and "2" where possible to achieve better security. Method "3" is especially weak when not used in combination with knowledge of account activity.

Control	Levels of Assurance			
	Level 1	Level 2	Level 3	Level 4
		for data is **seven (7) years and six (6) months beyond the expiration or revocation** (whichever is later) of the credential.	for data is **seven (7) years and six (6) months beyond the expiration or revocation** (whichever is later) of the credential.	for data **is ten (10) years and six (6) months beyond the expiration or revocation** of the credential.
5. Remote One-Time Use Credential Requirements				
5.1 Identity Proofing	There are no level-specific requirements at Level 1.	Application owners act as the RA/ CSP in the remote registration processes described in this table (Section 3.1 above), using processes that do not require confirmation of the address of record and omitting credential issuance.	Application owners act as the RA/ CSP in the remote registration processes described in this table (Section 3.1 above), using processes that do not require confirmation of the address of record and omitting credential issuance.	**Not Applicable**
5.2 Registration Process	There are no level-specific requirements at Level 1.	For immediate one-time access, application owners can use the registration processes specified in this table (Section 3.2 above) that: 1. Confirm “the ability of the Applicant to receive telephone communications or text message at phone number or e-mail address associated with the Applicant in records”; or 2. Subsequently send a “notice to an address of record confirmed in the records check.”	For immediate one-time access, application owners can use the registration process specified in this table (Section 3.2 above) that: 1. Confirms “the ability of the Applicant to receive telephone communications at a phone number associated with the Applicant in records while recording the Applicant’s voice or using alternative means that establish an equivalent level of non-repudiation.”	**Not Applicable**

Remote registration at Levels 2 and 3 requires confirmation of a financial or utility account number. The requirement for a financial account or utility account number may be satisfied by a cellular or landline telephone service account under the following conditions:

- The phone is associated in Records with the Applicant’s name and address of record; and
- The applicant demonstrates that they are able to send or receive messages at the phone number.

(This Page Intentionally Blank)

3.2.4.5 MAPPING OF FEDERAL PKI CERTIFICATE POLICIES TO E-AUTHENTICATION ASSURANCE LEVELS

The primary mechanism for evaluating the assurance provided by public key certificates issued under organization specific policies is the policy mapping of the *Federal Bridge Certification Authority (FBCA)* policies. These policies include the *Rudimentary, Basic, Medium, PIV-I Card Authentication, Medium-HW,* and *High* assurance policies specified *in X.509 Certificate Policy For The Federal Bridge Certification Authority*[27] *(FBCA)* and in *X.509 Certificate Policy For The U.S. Federal PKI Common Policy Framework*[28]*;* and the class policy specified in the *Citizen and Commerce Class Common Certificate Policy*[29].

These policies incorporate all aspects of the credential lifecycle in greater detail than specified in this document. These policies also include security controls (e.g., multi-party control and system auditing for CSPs) that are outside the scope of this document. However, the FPKI policies are based on work that largely predates this specification, and the security requirements are not always strictly aligned with those specified here. As a result, this section provides an overall mapping between FPKI certificate policies and the e-authentication Levels instead of a strict evaluation of compliance. There are known discrepancies, such as FIPS 201's allowance for pseudonyms on credentials issued to personnel in dangerous jobs, or the ability to issue PIV credentials based on a single federal government issued identity credential. While these discrepancies are recognized, the overall level of assurance provided by these policies is deemed to meet the requirements based on the additional controls.

Table 9 below summarizes how certificates issued under the Common Policy Framework correspond to the e-authentication assurance levels. Table 9 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 identity proofing, token, and status reporting requirements. (For this evaluation, a strict compliance mapping should be used, rather than the rough mapping used for the FPKI policies.) For Levels 3 and 4, agencies shall depend upon the mappings provided by the FPKI.

The FPKI has also added two policies, *Medium Commercial Best Practices (Medium-CBP)* and *Medium Hardware Commercial Best Practices (Medium HW-CBP)* to support recognition of non-Federal PKIs. In terms of e-authentication levels, the Medium CBP and Medium HW-CBP are equivalent to Medium and Medium-HW, respectively (with the exception of some personnel security requirements and subscriber cryptographic module requirements.)

[27] The *X.509 Certificate Policy For The Federal Bridge Certification Authority* is available at http://www.idmanagement.gov/documents/certificate-policy-federal-bridge-certificate-authority.

[28] The *X.509 Certificate Policy For The U.S. Federal PKI Common Policy Framework* is available at http://www.idmanagement.gov/documents/federal-pki-common-policy-framework-certificate-authority.

[29] The *Citizen and Commerce Class Common Certificate Policy* is available at http://www.idmanagement.gov/documents/citizen-and-commerce-class-common-certificate-policy.

Table 9 Certificate Policies and the E-authentication Assurance Levels

Certificate Policy	Selected Policy Components			Overall Equivalence
	Identity Proofing	Token	Token and Credential Management	
Common-Auth PIVI-Auth SHA1-Auth[30]	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 PIVI-HW SHA1-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
FBCA Basic[31]	Meets Level 3	Meets Level 3	Meets Level 3	Meets Level 3
FBCA Medium	Meets Level 4	Meets Level 3	Meets Level 4	Meets Level 3
FBCA Medium-HW	Meets Level 4	Meets Level 4	Meets Level 4	Meets Level 4
FBCA High	Meets Level 4	Meets Level 4	Meets Level 4	Meets Level 4
Common-cardAuth PIVI-cardAuth SHA1-cardAuth	Meets Level 4	Meets Level 2	Meets Level 4	Meets Level 2

3.2.5 AUTHENTICATION MECHANISM REQUIREMENTS

This section covers the mechanical authentication process of a Claimant who already has registered a token. The authentication process shall provide sufficient information to uniquely identify the registration information provided by the Subscriber and verified by the RA in the issuance of the credential. The technical requirements for authentication mechanisms (tokens, protocols, and security protections) are described in this section.

In the e-authentication context, a token contains a secret to be used in authentication processes. Tokens are possessed by a Claimant and controlled through one or more of the traditional authentication factors (something you know, have, or are). Tokens may exist in hardware (e.g., a smart card), software (e.g., a software cryptographic module), or may only exist in human memory. The output of a token is the token authenticator, which is the value that is provided to the protocol stack for transmission to the Verifier to prove that the Claimant possesses and

[30] For ***all*** SHA1 policies, the SHA1 policies have been deprecated and are not acceptable after December 31, 2013.
[31] For ***all*** FBCA policies, these policies are not asserted in the user certificates, but equivalence is established through policy mapping at the *Federal Bridge CA*.

controls the token. The token authenticator may be the token secret, or a transformation of the token secret.

3.2.5.1 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 something you *know*, a biometric is something you *are*, and a cryptographic identification device is something you *have*. Tokens may be single-factor 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. There are no additional factors required to activate the token, so this is considered single factor.
- 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 smart card is something you have, and something you know (the PIN) is required to activate the token.

This document does not differentiate between tokens that require two factors and three factors, as two factors are sufficient to achieve the highest level recognized in this document. Other applications or environments may require such a differentiation.

3.2.5.2 TOKEN TYPES

The NIST SP 800-63 guidelines and CMS recognize the following types of tokens for e-authentication.

- *Memorized Secret Token* – A secret shared between the Subscriber and the CSP. Memorized Secret Tokens are typically character strings (e.g., passwords and passphrases) or numerical strings (e.g., PINs). The token authenticator presented to the Verifier in an authentication process is the secret itself (e.g. the password or passphrase itself). Memorized Secret Tokens are something you know.
- *Pre-registered Knowledge Token* – A series of responses to a set of prompts or challenges. These responses may be thought of as a set of shared secrets. The set of prompts and responses are established by the Subscriber and CSP during the registration process. The token authenticator is the set of memorized responses to pre-registered prompts during a single run of the authentication process. An example of a Pre-registered Knowledge Token would be establishing responses for prompts such as *"What was your first pet's name?"* During the authentication process, the Claimant is asked to provide the appropriate responses to a subset of the prompts. Alternatively, a Subscriber might select and memorize an image during the registration process. In an authentication process, the Claimant is prompted to identify the correct images from a set(s) of similar images. Transactions from previously authenticated sessions could be accepted as Pre-registered Knowledge Tokens. Pre-registered Knowledge Tokens are something you know.
- *Look-up Secret Token* – A physical or electronic token that stores a set of secrets shared between the Claimant and the CSP. The Claimant uses the token to look up the appropriate

secret(s) needed to respond to a prompt from the Verifier (the token input). 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. The token authenticator is the secret(s) identified by the prompt. Look-up Secret Tokens are something you have.

- *Out of Band Token* – A physical token that is uniquely addressable and can receive a Verifier-selected secret for one-time use. The device is possessed and controlled by the Claimant and supports private communication (i.e., Verifier's message can be sent directly to Claimant's device) over a channel that is separate from the primary channel for e-authentication. The token authenticator is the received secret and is presented to the Verifier using the primary channel for e-authentication. For example, a Claimant attempts to log into a website and receives a text message on his or her cellular phone, Personal Digital Assistants (PDA), pager, or landline (pre-registered with the CSP during the registration phase) with a random authenticator to be presented as a part of the electronic authentication protocol. Out of Band Tokens are something you have.
- *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 and control of the device. The token authenticator is the one-time password. For example, a one-time password device may display six (6) characters at a time. SF OTP Devices are something you have.
- *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. The token authenticator is highly dependent on the specific cryptographic device and protocol, but it is generally some type of signed message. For example, in TLS, there is a "certificate verify" message. SF Cryptographic Devices are something you have.
- *Multi-factor (MF) Software Cryptographic Token* – A cryptographic key 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. The token authenticator is highly dependent on the specific cryptographic protocol, but it is generally some type of signed message. For example, in TLS, there is a "certificate verify" message. The MF Software Cryptographic Token is something you have, and it may be activated by either something you know or something you are.
- *Multi-factor (MF) One-Time Password (OTP) Device* – A 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., Universal Serial Bus [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. The token authenticator is the one-time password. For example, a one-time password device may display six (6) characters at a time. The MF OTP Device is something you have, and it may be activated by either something you know or something you are.

- *Multi-factor (MF) Cryptographic Device* – A hardware device that contains a protected cryptographic key that requires activation through a second authentication factor. Authentication is accomplished by proving possession of the device and control of the key. The token authenticator is highly dependent on the specific cryptographic device and protocol, but it is generally some type of signed message. For example, in TLS, there is a "certificate verify" message. The MF Cryptographic Device is something you have, and it may be activated by either something you know or something you are.

3.2.5.3 TOKEN USAGE

An authentication process may involve a single token, or a combination of two or more tokens, as described below:

- *Single-token authentication* – The Claimant presents a single token authenticator to prove his or her identity to the Verifier. For example, when a Claimant attempts to log into a password-protected website, the Claimant enters a username and password. In this instance, only the password would be considered to be a token.
- *Multi-token authentication* – The Claimant presents token authenticators generated by two or more tokens to prove his or her identity to the Verifier. The combination of tokens is characterized by the combination of factors used by the tokens (both inherent in the manifestation of the tokens, and those used to activate the tokens). A Verifier that requires a Claimant to enter a password and use a single-factor cryptographic device is an example of multi-token authentication. The combination is considered multi-factor, since the password is something you know and the cryptographic device is something you have.

3.2.5.4 MULTI-STAGE AUTHENTICATION USING TOKENS

Multi-stage authentication processes, which use a single-factor token to obtain a second token, ***do not constitute multi-factor authentication***. The level of assurance associated with the compound solution is the assurance level of the *weakest* token.

For example, 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 successful authentication using a password or 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 to obtain the cryptographic token.

3.2.5.5 ASSURANCE LEVEL ESCALATION

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*. NIST SP 800-63 recognizes a special case of multi-token authentication, where a primary token is used to establish a secure session, and a secondary token is used *later in the session* to present a second token authenticator. Even though the two tokens were not used at the same time, CMS recognizes the result as a multi-token authentication scheme (which may upgrade the overall level of assurance of the session). In these authentication scenarios, the level of assurance achieved by the two stages in combination is the same as a multi-token authentication scheme using the same set of tokens. Table 13 describes the highest level of assurance achievable through a combination of two token types.

3.2.6 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 lost, damaged, 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, tampered with, 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 the secret. Additionally, an Attacker may determine the secret through offline attacks on network traffic from an authentication attempt. Finally, an Attacker may be able to gain information about a Subscriber's Pre-registered Knowledge researching the subscriber or through other social engineering techniques. (For example, the subscriber might refer to his or her first pet in a conversation or blog.)
- *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 Subscriber 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 in Table 10, along with some examples.

Table 10 Token Threats

Token Threats/ Attacks	Description	Examples
Theft	A physical token is stolen by an Attacker.	A hardware cryptographic device is stolen.
		A one-time password is stolen.
		A look-up secret token is stolen.
		A cell phone is stolen.
Discovery	The responses to token prompts are easily discovered through searching various data sources.	The question "What high school did you attend?" is asked as a Preregistered Knowledge Token, when the answer is commonly found on social media websites.
Duplication	The Subscriber's token has been copied with or without his or her knowledge.	Passwords written on paper are disclosed.
		Passwords stored in an electronic file are copied.
		Software PKI token (private key) copied.

Token Threats/ Attacks	Description	Examples
		Look-up 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.	Passwords are learned by watching keyboard entry.
		Passwords are learned by keystroke logging software.
		A PIN is captured from PIN pad device.
Offline cracking	The token is exposed using analytical methods outside the authentication mechanism.	A key is extracted by differential power analysis on stolen hardware cryptographic token.
		A software PKI token is subjected to dictionary attack to identify the correct password to use to decrypt the private key.
Phishing or pharming	The token secret or authenticator is captured by fooling the Subscriber into thinking the Attacker is a Verifier or RP.	A password is revealed by Subscriber to a website impersonating the Verifier.
		A password is revealed by a bank Subscriber in response to an email inquiry from a Phisher pretending to represent the bank.
		A password is revealed by the 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.	A password is revealed by the Subscriber to an officemate asking for the password on behalf of the Subscriber's boss.
		A password is revealed by a Subscriber in a telephone inquiry from an Attacker masquerading as a 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 are used to guess passwords.
		Online guessing is used to guess token authenticators for a onetime password token registered to a legitimate Claimant.

3.2.7 THREAT MITIGATION STRATEGIES

Token related mechanisms that assist in mitigating the threats identified above are summarized in Table 11.

Table 11 Mitigating Token Threats

Token Threats/ Attack	Threat Mitigation Mechanisms
Theft	• Use multi-factor tokens that need to be activated through a PIN or biometric.
Duplication	• Use tokens that are difficult to duplicate, such as hardware cryptographic tokens.

Token Threats/ Attack	Threat Mitigation Mechanisms
Discovery	• Use methods in which the responses to prompts cannot be easily discovered.
Eavesdropping	• Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator. • Use tokens that generate authenticators based on a token input value. • Establish tokens through a separate channel.
Offline cracking	• Use a token with a high entropy token secret • 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.
Social engineering	• Use tokens with dynamic authenticators where knowledge of one authenticator does not assist in deriving a subsequent authenticator.
Online guessing	• Use tokens that generate high entropy authenticators.

There are several other strategies that may be applied to mitigate the threats described in Table 11:

- *Multiple factors* raise the threshold for successful attacks. If an Attacker needs to steal a cryptographic token and guess a password, then the work to discover both factors 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 do not 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.
- *Periodic training* may be performed to ensure the Subscriber understands when and how to report compromise (or suspicion of compromise) or otherwise recognize patterns of behavior that may signify an Attacker attempting to compromise the token.
- *Out of band techniques* may be employed to verify proof of possession of registered devices (e.g., cell phones).

3.2.8 TOKEN ASSURANCE LEVELS

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

Single Token Authentication

Table 12 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, then the token may be used at lower levels but shall satisfy the requirements given at whatever level is listed. If there is more than one box under “Verifier Requirements” for a given token type, then it is only necessary to satisfy the requirements in one box.

(This Page Intentionally Blank)

Table 12 Token Requirements Per Assurance Level

Token Type	Level	Token Requirements	Verifier Requirements
Memorized Secret Token	**Level 1**	The memorized secret may be a user chosen string consisting of 6 or more characters chosen from an alphabet of 90 or more characters, a randomly generated PIN consisting of 4 or more digits, or a secret with equivalent entropy.	The Verifier shall implement a throttling mechanism that effectively limits the number of failed authentication attempts an Attacker can make on the Subscriber's account to 100 or fewer in any 30-day period. Note: While an implementation that simply counted all failed authentication attempts in each calendar month and locked out the account when the limit was exceeded would technically meet the requirement, this is a poor choice for reasons of system availability. Refer to NIST SP 800 63, Section 8.2.3, *Throttling Mechanisms*, for more detailed advice.
	Level 2	The memorized secret may be a randomly generated PIN consisting of 6 or more digits, a user generated string consisting of 8 or more characters chosen from an alphabet of 90 or more characters, or a secret with equivalent entropy. CSP implements dictionary or composition rule to constrain user-generated secrets.	The Verifier shall implement a throttling mechanism that effectively limits the number of failed authentication attempts an Attacker can make on the Subscriber's account to 100 or fewer in any 30-day period. Note: While an implementation that simply counted all failed authentication attempts in each calendar month and locked out the account when the limit was exceeded would technically meet the requirement, this is a poor choice for reasons of system availability. Refer to NIST SP 800 63, Section 8.2.3, *Throttling Mechanisms*, for more detailed advice.

Token Type	Level	Token Requirements	Verifier Requirements
Pre-Registered Knowledge Token	**Level 1**	The secret provides at least 14 bits of entropy.	The Verifier shall implement a throttling mechanism that effectively limits the number of failed authentication attempts an Attacker can make on the Subscriber's account to 100 or fewer in any 30-day period. Note: While an implementation that simply counted all failed authentication attempts in each calendar month and locked out the account when the limit was exceeded would technically meet the requirement, this is a poor choice for reasons of system availability. Refer to NIST SP 800 63, Section 8.2.3, *Throttling Mechanisms*, for more detailed advice.
		The entropy in the secret cannot be directly calculated, e.g., user chosen or personal knowledge questions. If the questions are not supplied by the user, the user shall select prompts from a set of at least 5 questions.	For these purposes, an empty answer is prohibited. The Verifier shall verify the answers provided for at least 3 questions, and shall implement a throttling mechanism that effectively limits the number of failed authentication attempts an Attacker can make on the Subscriber's account to 100 or fewer in any 30-day period. Note: While an implementation that simply counted all failed authentication attempts in each calendar month and locked out the account when the limit was exceeded would technically meet the requirement, this is a poor choice for reasons of system availability. Refer to NIST SP 800 63, Section 8.2.3, *Throttling Mechanisms*, for more detailed advice.

Token Type	Level	Token Requirements	Verifier Requirements
	Level 2	The secret provides at least 20 bits of entropy.	The Verifier shall implement a throttling mechanism that effectively limits the number of failed authentication attempts an Attacker can make on the Subscriber's account to 100 or fewer in any 30-day period. Note: While an implementation that simply counted all failed authentication attempts in each calendar month and locked out the account when the limit was exceeded would technically meet the requirement, this is a poor choice for reasons of system availability. Refer to NIST SP 800 63, Section 8.2.3, *Throttling Mechanisms*, for more detailed advice.
		The entropy in the secret cannot be directly calculated, e.g., user chosen or personal knowledge questions. If the questions are not supplied by the user, the user shall select prompts from a set of at least 7 questions.	For these purposes, an empty answer is prohibited. The Verifier shall verify the answers provided for at least 5 questions, and shall implement a throttling mechanism that effectively limits the number of failed authentication attempts an Attacker can make on the Subscriber's account to 100 or fewer in any 30-day period. Note: While an implementation that simply counted all failed authentication attempts in each calendar month and locked out the account when the limit was exceeded would technically meet the requirement, this is a poor choice for reasons of system availability. Refer to NIST SP 800 63, Section 8.2.3, *Throttling Mechanisms*, for more detailed advice.
Look-up Secret	**Level 2**	The token authenticator has 64 bits of entropy.	Not Applicable.

Token Type	Level	Token Requirements	Verifier Requirements
Token		The token authenticator has at least 20 bits of entropy.	The Verifier shall implement a throttling mechanism that effectively limits the number of failed authentication attempts an Attacker can make on the Subscriber's account to 100 or fewer in any 30-day period. Note: While an implementation that simply counted all failed authentication attempts in each calendar month and locked out the account when the limit was exceeded would technically meet the requirement, this is a poor choice for reasons of system availability. Refer to NIST SP 800 63, Section 8.2.3, *Throttling Mechanisms*, for more detailed advice.
Out of Band Token	**Level 2**	The token is uniquely addressable and supports communication over a channel that is separate from the primary channel for e-authentication.	The Verifier generated secret shall have at least 64 bits of entropy.
			The Verifier generated secret shall have at least 20 bits of entropy and the Verifier shall implement a throttling mechanism that effectively limits the number of failed authentication attempts an Attacker can make on the Subscriber's account to 100 or fewer in any 30-day period. Note: While an implementation that simply counted all failed authentication attempts in each calendar month and locked out the account when the limit was exceeded would technically meet the requirement, this is a poor choice for reasons of system availability. Refer to NIST SP 800 63, Section 8.2.3, *Throttling Mechanisms*, for more detailed advice.
Single-factor (SF) One-Time Password (OTP) Device	**Level 2**	Shall 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 nonce may be a date and time, or a counter generated on the device.	The one-time password shall have a limited lifetime, on the order of minutes. The cryptographic module performing the verifier function shall be validated at FIPS 140-2 Level 1 or higher. (Products validated under subsequent versions of FIPS 140-2 are also acceptable.)

Token Type	Level	Token Requirements	Verifier Requirements
Single-factor (SF) Cryptographic Device	**Level 2**	The cryptographic module shall be validated at FIPS 140-2 Level 1 or higher. (Products validated under subsequent versions of FIPS 140-2 are also acceptable.) Note: Products validated under subsequent versions of FIPS 140-2 are also acceptable.	Verifier generated token input (e.g., a nonce or challenge) has at 64 bits of entropy.
Multi-factor (MF) Software Cryptographic Token	**Level 3**	The cryptographic module shall be validated at FIPS 140-2 Level 1 or higher. (Products validated under subsequent versions of FIPS 140-2 are also acceptable.) 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.	Verifier generated token input (e.g., a nonce or challenge) has at least 64 bits of entropy.
Multi-factor (MF) One-Time Password (OTP) Hardware Token	**Level 4**	Cryptographic module shall be FIPS 140-2 validated Level 2 or higher; with physical security at FIPS 140-2 Level 3 or higher. (Products validated under subsequent versions of FIPS 140-2 are also acceptable.) 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 onetime password. 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.	The one-time password shall have a limited lifetime of less than 2 minutes.
Multi-factor (MF) Hardware Cryptographic Token	**Level 4**	Cryptographic module shall be FIPS 140-2 validated, Level 2 or higher; with physical security at FIPS 140-2 Level 3 or higher. (Products validated under subsequent versions of FIPS 140-2 are also acceptable.) Shall require the entry of a password, PIN, or biometric to activate the authentication key. Shall not allow the export of authentication keys.	Verifier generated token input (e.g., a nonce or challenge) has at least 64 bits of entropy.

Multi-Token Authentication

Table 13 shows the highest possible assurance level that can be achieved when two token types are combined for a multi-token authentication scheme. Table 13 displays the highest possible assurance level that can be achieved by the token combination. Note that Table 13 displays only tokens that exhibit the properties of "something you have" and "something you know." The Table 13 boxes marked with an "X" denote that the combination already appears in the table.

Table 13 Assurance Levels for Multi-Token E-authentication Schemes

	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 Device	MF OTP Hardware Device	MF Hardware 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 Device	X	X	X	X	X	X	Level 3	Level 4	Level 4

	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 Device	MF OTP Hardware Device	MF Hardware Crypto-graphic Device
MF OTP Hardware Device	X	X	X	X	X	X	X	Level 4	Level 4
MF Hardware Cryptographic Device	X	X	X	X	X	X	X	X	Level 4

(This Page Intentionally Blank)

The principles used in generating Table 13 are as follows. Level 3 can be achieved using two (2) tokens rated at Level 2 that represent two different factors of authentication. Since this specification does not address the use of biometrics as a stand-alone token for remote authentication, achieving Level 3 with separate Level 2 tokens implies something you have and something you know:

Token (Level 2, *something you have*) + Token (Level 2, *something you know*) → Token (Level 3)

In all other cases, combinations of tokens are considered to achieve the level of the highest rated token.

For example, a Memorized Secret Token combined with a Look-up Secret Token can be used to achieve Level 3 authentication, since the Look-up Secret Token is “something you have” and the Memorized Secret Token is “something you know.” However, combining a MF Software Cryptographic token (“something you know” rated at Level 3) and a Memorized Secret Token (also “something you know” rated at Level 2) achieves an overall level of 3, because the addition of the Memorized Secret Token does not increase the assurance of the combination—both are “something you know.”

It should be noted that to achieve Level 4 with a single token or token combination, one of the tokens must be usable with an authentication process that strongly resists man-in-the-middle (MitM) attacks. While it is possible to meet this requirement with a wide variety of token types, certain choices of tokens may complicate the task of designing a protocol that meets Level 4 requirements for the authentication process. In particular, one-time password devices that rely exclusively d may need to be suppl stance.

Refer to NI nformation on Token M

3.2.9 A

The authen a certain degree of a change, as well as mar ntication activity. O arried on a protected se

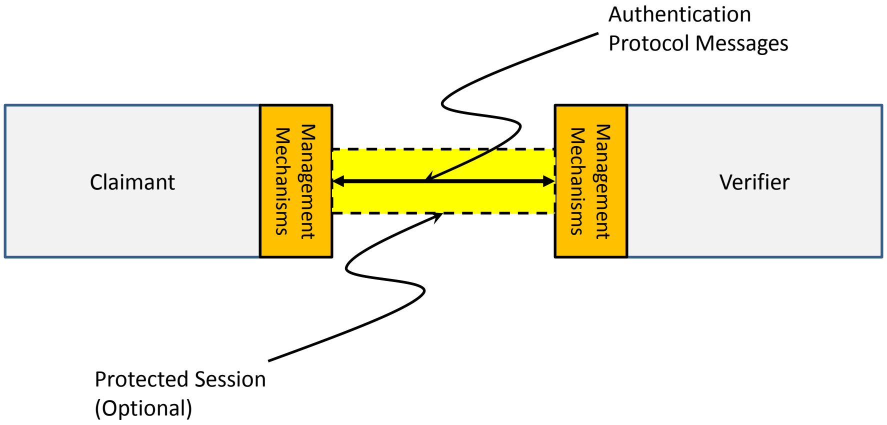


Figure 2 Authentication Process Model

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 session may be created between the two parties; this protected session may be used to exchange the remaining messages of the authentication protocol run, or to exchange session data between the two parties.

Management mechanisms 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, mechanisms may be implemented on the Verifier to limit the rate of online guessing of passwords by an Attacker who is trying to authenticate as a legitimate Claimant. Further, detection of authentication transactions originating from an unexpected location or channel for a Claimant, or indicating use of an unexpected hardware or software configuration, may indicate increased risk levels and motivate additional confirmation of the Claimant's identity.

At the conclusion of the authentication protocol run, the verifier might issue a secondary authentication credential, such as a cookie, to the Claimant and rely upon it to authenticate the claimant in the near future.

3.2.10 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.

RAs, CSPs, and Verifiers 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, and Verifiers 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. Table 14 lists the types of threats posed to the authentication process.

Table 14 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.

Type of Attack	Description	Example
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 he or she is interacting with the legitimate Verifier.
Eavesdropping	An Attacker listens passively to the authentication protocol to capture information that 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.
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/RP 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 Hypertext Transfer Protocol (HTTP) requests sent by the Subscriber.
Man-in-the-middle (MitM)	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.
		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.

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

- Denial of Service 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 or otherwise exploit 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 or otherwise exploiting the Claimant's 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 (refer to *CMS Information Security Acceptable Risk Safeguards (ARS), CMS Minimum Security Requirements (CMSR)* manual[32]). Hardware tokens prevent malicious software from extracting and copying the token secret. However, malicious code may still misuse the token, particularly if activation data is presented to the token via the computer.

3.2.10.1 THREAT MITIGATION STRATEGIES

The following are strategies that can be incorporated in authentication processes 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 prior 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 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 a 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 MitM 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

[32] The ARS manual can be found at http://www.cms.gov//Research-Statistics-Data-and-Systems/CMS-Information-Technology/InformationSecurity/.

and pharming attacks, it is recommended that the Claimant authenticate the Verifier using cryptographic mechanisms prior to submitting the token authenticator to the 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. 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 RP. 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[33] for an Attacker to carry out an off-line attack where he or she records an authentication protocol run and then analyzes 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. Protected session protocols, such as TLS, provide eavesdropping resistance.
- ***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 "freshness" 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 RP 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 RP to authenticate the transfer of all session data.

  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 Uniform Resource Locator (URL) of the legitimate RP. The malicious website is generally constructed so that a web browser will automatically send an HTTP request to the RP 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 RP, the request will generally be sent in the same session and with

[33] "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.

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 RP. This can be done by inserting random data, supplied by the RP, into any linked URL with side effects and into a hidden field within any form on the RP's website. This mechanism, however, is not effective if the Attacker can run scripts on the RP's website (Cross Site Scripting or XSS). To prevent XSS vulnerabilities, the RP 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 MitM attack when both parties (i.e., 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:

  a. *Weak man-in-the-middle resistance* – A protocol is said to be weakly resistant to MitM 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 MitM 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 MitM resistance can also be provided by a zero-knowledge password protocol, such as EKE, SPEKE, or 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).

  b. *Strong man-in-the-middle resistance* – A protocol is said to be strongly resistant to MitM 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. 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 MitM attacks.

Note that systems can supplement the mitigation strategies listed above by enforcing appropriate security policies. For example, device identity, system health checks, and configuration management can be used to mitigate the risk that the Claimant’s system has been compromised.

Refer to NIST SP 800-63 Section 8.2.3, *Throttling Mechanisms*, and Section 8.2.4, *Phishing & Pharming (Verifier Impersonation): Supplementary Countermeasures*, for additional guidance.

3.2.11 AUTHENTICATION PROCESS ASSURANCE LEVELS

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

3.2.11.1 THREAT RESISTANCE PER ASSURANCE LEVEL

Authentication process assurance levels can be defined in terms of required threat resistance. Table 15 lists the threat resistance requirements per assurance level:

Table 15 Required Authentication Protocol Threat Resistance per Assurance Level

	Threat Resistance Requirements			
Authentication Process Attacks/Threats	**Level 1**	**Level 2**	**Level 3**	**Level 4**
Online guessing	Yes	Yes	Yes	Yes
Replay	Yes	Yes	Yes	Yes
Session hijacking	No	Yes	Yes	Yes
Eavesdropping	No	Yes	Yes	Yes
Phishing/pharming (verifier impersonation)	No	No	Yes[34]	Yes
Man-in-the-middle (MitM)	No	Weak	Weak	Strong
Denial of service/flooding[35]	No	No	No	No

3.2.11.2 REQUIREMENTS PER ASSURANCE LEVEL

This section states the requirements levied on the authentication process to achieve the required threat resistance 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.

[34] Long-term authentication secrets shall be protected at this level. Short-term secrets may or may not be protected.

[35] Although there are techniques used to resist flood attacks, no protocol has comprehensive resistance to stop flooding.

Level 1

Although there is no identity-proofing requirement at this level, the authentication mechanism provides some assurance that the same Claimant who participated in previous transactions 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 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 possesses and 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. 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. At 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 Claimant by use of a challenge-response protocol. A password sent through a TLS protocol session is another example. Other common protocols that meet Level 1 requirements include Authenticated Post Office Protocol (APOP) [RFC 1939][36], S/KEY [RFC 1760][37], and password-based versions of Kerberos.

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 online 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.

Session data transmitted between the Claimant and the RP following a successful Level 2 authentication shall 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).

[36] RFC 1939, *Post Office Protocol - Version 3,* is available at http://tools.ietf.org/html/rfc1939.

[37] RFC 1760, *The S/KEY One-Time Password System,* is available at http://tools.ietf.org/html/rfc1760.

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).

Level 3

Level 3 provides multi-factor remote network authentication. At least two authentication factors are required. Level 3 authentication is based on proof of possession of the allowed types of tokens through a cryptographic protocol. Level 3 also permits any of the token methods of Level 4. Refer to Section 3.2.8, *Token Assurance Levels*, for single tokens and token combinations that can 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, possibly via the Claimant. 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.

Level 4

Level 4 is intended to provide the highest practical remote network authentication assurance. Refer to Section 3.2.8, *Token Assurance Levels*, for single tokens and token combinations that can 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 3.2.10, *Authentication Process Threats*, 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 RPs 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.

Level 4 tokens are required to be hardware cryptographic modules validated at FIPS 140-2 Level 2 or higher with physical security at FIPS 140-2 Level 3 or higher. Level 4 token requirements can be met by using the PIV authentication key of a FIPS 201 compliant PIV Card.

It should be noted that, in multi-token schemes, the token used to provide strong MitM resistance need not be a hardware token. For example, if a software cryptographic token is used to open a client-authenticated TLS session, and the output of a multifactor OTP device is sent by the claimant in that session, then the resultant protocol will still provide Level 4 assurance.

3.2.12 ASSERTIONS

Assertions are statements from a Verifier to an RP that contain information about a Subscriber. Assertions are used when the RP and the Verifier are not collocated (i.e., they are connected through a shared network). The RP uses the information in the assertion to identify the Claimant and make authorization decisions about his or her access to resources controlled by the RP. 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 RP.

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 RPs without being aware of 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 RPs through the use of a common "federated" identifier. In this context, a federation is a group of entities (RPs, 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. Attributes are often used in determining access privileges for Attributes Based Access Control (ABAC) or Role Based Access Control (RBAC).

It is important to note that assertion schemes are fairly complex multiparty protocols, and therefore have fairly subtle security requirements, which shall 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 RP are similar to the authentication mechanisms presented in Section 3.2.9, *Authentication Process*, while interactions between the Verifier and RP are similar to the token and credential verification services presented in NIST SP 800-63 Section 7, *Token and Credential Management*. Many of the requirements presented in this section will, therefore, be similar to corresponding requirements in those two sections.

There are three basic models for assertion-based authentication. After successful authentication with the Verifier, the Subscriber is issued an assertion or an assertion reference, which the Subscriber uses to authenticate to the RP:

- 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 RP. The assertion is used by the Claimant/Subscriber to authenticate to the RP. (This is usually handled automatically by the Subscriber's browser.
- 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 RP. In this model, the assertion reference is used by the Claimant/ Subscriber to authenticate to the RP. The RP then uses the assertion reference to explicitly request the assertion from the Verifier.
- The Proxy Model – In the proxy 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 includes it when interacting directly with the RP, acting as an intermediary between the Claimant and the RP.

For more information on the above assertion models and on other assertion technologies and schemes, refer to NIST SP 800-63 Section 9, *Assertions*.

The next section describes the three more common types of assertion technologies: Web browser cookies, SAML assertions, and Kerberos tickets.

3.2.12.1 COOKIES

One type of assertion widely in use is Web cookie technology. Cookies are text files used by a browser to store information provided by a particular website. The contents of the cookie are sent back to the website each time the browser requests a page from the same website. The website 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 the 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***[38] – 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.

Cookies are effective as assertions for Internet single-sign-on where the RP 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 RP and the Verifier are part of disparate domains*.

Cookies are also often used by the Claimant to re-authenticate to a server. 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 an RP when it requests the cookie from a Claimant who wishes to re-authenticate to it. Often, the cookie contains a random number, and the assertion data that it represents does not leave the server. 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 therefore, confidentiality and integrity requirements for assertion data at Level 2 and below may be satisfied by access controls rather than by cryptographic methods. (The cookie itself, however, does need to be protected.)

3.2.12.2 SECURITY ASSERTION MARKUP LANGUAGE (SAML) ASSERTIONS

SAML is an XML-based framework for creating and exchanging authentication and attribute information between trusted entities over the Internet. SAML is an authentication *protocol* that is used between servers. SAML implementations still need something that actually performs the login. For example, when a Lightweight Directory Access Protocol (LDAP) server authenticates a user, the *authentication authority* is the *LDAP server*, even though the LDAP server may be using SAML to *communicate* the authorization. As of this writing, the latest specification for SAML is version 2.0, issued March 15, 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; and the *Bindings* that define the underlying communication protocols (such as HTTP or Simple Object Access Protocol [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.

[38] CMS websites are to be operated within the restrictions addressed in OMB directives M-10-22, *Guidance for Online Use of Web Measurement and Customization Technologies*, and M-10-23 and *Guidance for Agency Use of Third-Party Websites and Applications*. Restrictions on the use of persistent cookies, and other similar tracking technologies, are addressed within these memoranda. Also reference CMS ARS control requirement SC-CMS-2, *Website Usage*.

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 10:32 pm 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, and information on specific web servers. For example, subject "John" for action "Read" on "Webserver1002" given evidence "Role".

Authorization statements are beyond the scope of this document and will not be discussed.

3.2.12.3 KERBEROS TICKETS

The *Kerberos Network Authentication Service*[39] 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 RP.

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 RP.

To authenticate using the session key, the Claimant sends the ticket to the RP 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 RP.

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 (Public Key Cryptography for Initial Authentication in Kerberos) variant of Kerberos, a public key certificate. It should be noted that most variants of Kerberos based on a shared secret key between the Subscriber and Verifier derive this key from a user-

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

generated password. As such, they are *vulnerable to offline dictionary attack* by a passive eavesdropper.

In addition to delivering the session key to the subscriber, 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 RP 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 RP.

3.2.12.4 ASSERTION THREATS

Threats

In this section, it is assumed that the two endpoints of the assertion transmission (namely, the Verifier and the RP) 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 RP. Other Attackers are assumed to lurk within the shared transmission medium (e.g., Internet) and 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 RP 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 by the Verifier – 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 repudiation by the Subscriber – Since it is possible for a compromised or malicious subscriber to issue assertions to the wrong party, a subscriber can repudiate any transaction with the RP that was authenticated using only a bearer assertion.
- Assertion redirect – An Attacker uses the assertion generated for one RP to obtain access to a second RP.

- Assertion reuse – An Attacker attempts to use an assertion that has already been used once with the intended RP.

In addition to reliable and confidential transmission of assertion data from the Verifier to the RP, assertion protocols have a further goal: in order for the Subscriber to be recognized by the RP, he or she shall be issued some secret information, the knowledge of which distinguishes the Subscriber from Attackers who wish to impersonate the Subscriber. In the case of holder-of-key assertions, this secret is generally the Subscriber's long-term token secret, which would already have been established with the CSP prior to the initiation of the assertion protocol.

In other cases, however, the Verifier will generate a temporary secret and transmit it to the authenticated Subscriber for this purpose. Since, when this secret is used to authenticate to the RP, it generally replaces the token authenticator in the type of protocols described in Section 3.2.9, *Authentication Process*, this temporary secret will be referred to here as a secondary authenticator. Secondary authenticators include assertions in the direct model, session keys in Kerberos, assertion references in the indirect model, and cookies used for authentication. 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 RP. 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 RP 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 RP to be useful, the binding between the secret used to authenticate to the RP and the assertion data referring to the Subscriber shall be strong.

- Assertion substitution – A subscriber may attempt to impersonate a more privileged subscriber by subverting the communication channel between the Verifier and RP, for example by reordering the messages, to convince the RP 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.

Threat Mitigation Strategies

Logically speaking, an assertion is issued by a Verifier and consumed by an RP – these are the two end points of the session that needs to be secured to protect the assertion. In the direct model, the session in which the assertion is passed traverses the Subscriber. Furthermore, in the current web environment, the assertion may pass through two separate secure sessions (one between the Verifier and the Subscriber, and the other between the Subscriber and the RP), with a break in session 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 RP; this protocol session needs to be protected. All of the threat mitigation strategies in Section 3.2.9, *Authentication Process*, 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:
    - The assertion may be digitally signed by the Verifier. The RP should check the digital signature to verify that it was issued by a legitimate Verifier.
    - The assertion may be sent over a protected session such as TLS. In order to protect the integrity of assertions from malicious attack, the Verifier shall be authenticated.
- Assertion disclosure – To mitigate this threat, one of the following mechanisms may be implemented:
    - The assertion may be sent over a protected session to an authenticated RP. Note that, in order to protect assertions against both disclosure and manufacture/modification using a protected session, both the RP and the Verifier need to be authenticated.
    - If assertions are signed by the Verifier, they may be encrypted for a specific RP 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 session, 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 session or equivalent between Verifier and RP.
- Assertion repudiation by the Verifier – To mitigate this threat, the assertion may be digitally signed by the Verifier using a key that supports nonrepudiation. The RP should check the digital signature to verify that it was issued by a legitimate Verifier.
- Assertion repudiation by the Subscriber – To mitigate this threat, the Verifier may issue holder of key, rather than bearer assertions. The Subscriber can then prove possession of the asserted key to the RP. If the asserted key matches the subscriber's long-term credential (as provided by the CSP), it will be clear to all parties involved that it was the Subscriber who authenticated to the RP rather than a compromised Verifier impersonating the Subscriber.
- Assertion redirect – To mitigate this threat, the assertion may include the identity of the RP for whom it was generated. The RP 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:
    - The assertion includes a timestamp and has a short lifetime of validity. The RP checks the timestamp and lifetime values to ensure that the assertion is currently valid. The lifetime value may either be in the assertion or set by the RP.

  - The RP 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:
  - 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.
  - 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 RP.
  - The Subscriber may authenticate to the RP 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 shall be in place throughout the lifetime of any secondary authenticators used in the assertion protocol:
  - In order to protect the secondary authenticator while it is in transit between the Verifier and the Subscriber, the secondary authenticator shall be sent via a protected session established during the primary authentication of the Subscriber using his or her token. This requirement is the same as the requirement in Section 3.2.9, *Authentication Process,* to protect sensitive data (in this case the secondary authenticator) from session hijacking attacks.
  - In order to protect the secondary authenticator from capture as it is submitted to the RP, the secondary authenticator shall be used in an authentication protocol, which protects against eavesdropping and man-in-the-middle attacks as described in Section 3.2.9, *Authentication Process*.
  - In order to protect the secondary authenticator after it has been used, it shall never be transmitted on an unprotected session 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 other RP. This is possible if the secondary authenticator is specific to a single RP, and if that RP 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:
  - 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 to the request by the RP.
  - 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.

Refer to NIST SP 800-63 Section 9.2.1, *Threat Mitigation Strategies*, for information on mitigating secondary authenticator threats.

3.2.13 ASSERTION ASSURANCE LEVELS

Table 16 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 16 Assertion 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 by Verifier	No	No	Yes[40]	Yes[40]
Assertion repudiation by Subscriber	No	No	No	Yes[40]
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

3.3 HUMAN USER AUTHENTICATION METHOD SELECTION CRITERIA

3.3.1 OVERVIEW OF AUTHENTICATION CRITERIA

All CMS *Human User* authentication requirements (this section does ***not*** apply to *Machine-to-Machine* authentication) are stipulated in the *CMS Information Security Acceptable Risk Safeguards (ARS), CMS Minimum Security Requirements (CMSR)* manual[41], in the *Identification and Authentication (IA)* family of security controls.

Determining the “proper” authentication mechanism for a given *Human User* authentication situation is made difficult because there are three different governing Federal authentication requirements (HSPD-12, OMB M-04-04, and the SP NIST 800-53 control requirements). To compound the confusion, each governing standard comes from a *different* source, they came out *years* apart, and each was developed to address a completely *different* view of the issues. As such, there is no official *standard* guidance (other than *this* standard) that ties all of these mandates together for implementation.

The *defining* of these different mandates is handled in different (and segregated) sections within this standard because they are *not directly related to each other*—by statute or directive. However, they are still *all* required within the scope of each *separate mandate*.

[40] Except for Kerberos.

[41] The ARS manual can be found at http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/InformationSecurity/Information-Security-Library.html.

HSPD-12 (PIV Cards)

The HSPD-12 mandate (see Section 3.1) is required for all *Physical* and *Logical* access control for *federal employees and contractors* (including contracted researchers). The only exception is for situations that *do not require identity assurance* (i.e., "public" systems not requiring a login.) For all others—i.e., anything requiring *any* identity assurance (as re-asserted in OMB M-11-11)—PIV cards are required. The HSPD-12 mandate makes no distinction over:

1. The *trustworthiness* of the network the user is using to facilitate the access (*"trusted"* or *"untrusted"*),
2. What *level of access* the user desires (*"privileged"* or *"non-privileged"*), or
3. Whether the *access method* is *"local"* or *"remote"*.

PIV is always required for ***any*** logical access that requires *any* identity assertion by a PIV card holder.

OMB 04-04 (E-authentication)

The OMB M-04-04 Directive (see Section 3.2) is applicable for *"remote"* human-user authentication over *"untrusted"* networks (i.e., the Internet). The e-authentication levels (1 through 4) are *only* applicable to those users that are:

1. Accessing *"remotely"*, ***and***
2. Accessing over an *"untrusted"* network (i.e., the Internet.)

The required authentication level (1 through 4) is contingent on the type of information being accessed and the risk associated with a breach or disclosure of such data. NIST SP 800-63 provides a detailed description of the assessment process required to assign the appropriate e-authentication levels for various data types. The CMS CISO has completed that assessment against each of the known CMS data types as defined by FIPS 199, and the results detailed in Table 5, *CMS Information Types & E-authentication Level Determination*.

NOTE: PIV is still ***always*** required for *any* logical access (even *"remote"*; *"trusted connection or not"*) for *all* PIV holders access where an identity assertion is required. As such, ***OMB M-04-04 only applies*** for *"Remote"* human-user authentication over *"untrusted"* networks (i.e., the Internet), of personnel (non-Federal employees and non-contractors) that are ***not covered*** under HSPD-12 PIV requirements. ***PIV is always required*** for ***any*** logical access that requires *any* identity assertion by a PIV card holder.

NIST SP 800-53 Control Requirements

The authentication requirements associated with IA-2 and IA-8 (see the CMS ARS) are required by NIST SP 800-53, and differentiate the *multi-factor authentication* requirements for either *"organizational"* users (addressed in IA-2), or *"non-organizational"* users (addressed in IA-8).

Control IA-2 and enhancements 1, 2, 3, and 4 detail the *"Organizational"* authentication requirements (multifactor) for:

1. *"Privileged"* users,
2. *"Non-privileged"*,
3. *"Network"* access, and
4. *"Local"* access.

Control IA-8 clarifies (and enhances) with authentication of *"non-organizational"* users. These non-organizational users *still* fall under the e-authentication requirements of OMB M-04-04. Under the NIST SP 800-53 requirements, the use of multifactor authentication is dependent on:

- The *Trustworthiness* of the *Access Method* (*"network"* or *"local"*),
- The *Level of Access* (*"privileged"* or *"non-privileged"*), and
- The *Type of User* (*"organizational"* or *"non-organizational"*.)

NOTE:

- HSPD-12 still overrides *everything* when a PIV card holder is involved—***PIV is always required*** for ***any*** logical access that requires *any* identity assertion by a PIV card holder.
- Other OMB 04-04 requirements *still* apply for *"remote"* human-user (non-PIV covered users) authentication over *"untrusted"* networks (i.e., the Internet).

Section 3.3.2 provides details explaining how each of these varying requirements are applied at CMS under specifically defined criteria.

3.3.2 DETAILS OF AUTHENTICATION CRITERIA

The primary factor for determining which type of authentication is required is the *population of users* that will be accessing the information system. NIST has segregated[42] the users into two populations; *Organizational* and, *Non-Organizational*, and addresses the applicable authentication requirements in two separate and distinct control requirements. CMS defines these user populations as follows:

- ***Organizational Users*** - Organizational users are defined as personnel who are accessing a CMS system (whether that system is hosted by CMS, or hosted by a CMS contractor) for the purposes of performing duties associated with their CMS employment or contractual relationship with CMS. Organizational user-authentication requirements are stipulated in IA-2 and its enhancements (enhancement applicability is dependent on the system security level.) **For organizational users, *e-authentication* requirements of Section 3.2 are *superseded* by the requirements listed in IA-2, *Identification and Authentication (Organizational Users)*.** At CMS, organizational users include (but are not limited to):
    - CMS employees

[42] These distinctions are made in the IA family of controls enumerated in the NIST SP 800-53, from which the ARS manual, and its associated control requirements, are directly derived.

  - CMS contractor/subcontractor staff
  - CMS-contracted researchers
- ***Non-Organizational Users*** - Non-organizational users are defined as users that are accessing CMS systems for any other purpose other than those defined in the definition of *Organizational users*. **Non-organizational user-authentication requirements are stipulated in IA-8, *Identification and Authentication (Non-Organizational Users),* and are based on the applicable *e-authentication level* of Section 3.2.** At CMS, these users include (but are not limited to):
  - Beneficiaries
  - Providers
  - State Medicaid employees and contractors/subcontractors
  - Non CMS-contracted researchers

Other factors that influence the CMS level of authentication requirement include:

- **The *Level of Access* of the user** - At CMS, *privileged access* is defined as an advanced level of access to a computer or application that includes the ability to perform configuration changes to either the application or the underlying supporting infrastructure. Some applications may have users with more *functionalities* than the normal user population; however, that does not necessarily mean that they would be considered *privileged* users.

  Users with *privileged* access rights require more stringent authentication than those users accessing via *non-privileged* account roles. Users with *privileged* access rights would be considered *organizational* users (and would be subject the requirements stipulated in IA-2).
- ***Access Method* used to connect to the system** - CMS access methods are segregated into three distinct types:
  - *Local access* - Local access is any access to an organizational information system by a user (or process acting on behalf of a user) where such access is obtained by direct connection without the use of a network.
  - *Trusted Network access* is defined as the ability to *authenticate* to a CMS computer or network via a connection through a *trusted data link*.
  - *Untrusted Network access* is defined as the ability to *authenticate* to a CMS computer or network via a connection through an *untrusted data link*.
- **The *Trustworthiness* of the data link** - The trustworthiness of CMS data links is segregated into two distinct types:
  - A *Trusted data link* is defined as a data-link that can be relied upon to enforce CMS security policy and security control requirements (as verified in a CMS system Authorization to Operate [ATO]). Examples include (but are not limited to):
    - Internal CMS Local Area Network (LAN).
    - An *established* encrypted (network layer) Virtual Private Network (VPN) that meets all applicable CMS security requirements. (See *Machine-To-Machine Authentication* requirements in Section 4, and NIST SP 800-77, *Guide to IPsec VPNs*.)

- An *Untrusted data link* is defined as a data-link that cannot be relied upon to enforce CMS security policy and security control requirements. Examples might include (but are not limited to):
  - The Internet.
  - Any network not included in a CMS FISMA system.
  - Networks included in a CMS FISMA system, but identified as non-compliant with CMS network security requirements.

3.3.3 HUMAN AUTHENTICATION MATRIX

Table 17 provides a *high-level* matrix for human authentication requirements—*for systems requiring an identity assertion*—under the various conditions described above. Table 17 provides sufficient detail to describe when different types (and number of factors) of authentication are required, based on NIST SP 800-63, the CMS enterprise e-authentication risk assessment, and NIST SP 800-53 requirements.

However, Table 17 may be difficult to interpret for systems that have ***several different types*** of users, networks interfaces, and user roles. For this reason, EISG has created a simple tool that will assist in determining what types of authentication are required for given situations. This *optional* tool (among others) can be found on the CMS Security library (at http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/InformationSecurity/Information-Security-Library.html) by filtering on “Tool”.

Note that for *non-organizational/non-privileged* users covered under e-authentication requirements (as defined in Section 3.2.3); a *further* evaluation must still be performed to determine the applicable e-authentication Level (1, 2, 3, or 4)—based on the data-types being accessed within the applicable sessions—and the associated requirements thereof. In addition, Table 17 does not override or supplant the PIV requirements associated with the HSPD-12 mandate. HSPD-12 requires that the PIV card be utilized—to varying degrees—for logical access to government systems by PIV cardholders.

Table 17 Human Authentication Matrix

System Security Level *(Defined in Table 5)*	**User Role** *(Defined in Section 3.2.3)*	**User Type** *(Defined in Section 3.2.3)*	**Access Method** *(Defined in Section 3.2.3)* **and Applicable ARS Control Requirement**	**Authentication Required** *(Defined in Section 2.3.1)*
Low	Non-Privileged	Organizational	Local (IA-2)	Single-factor
Low	Non-Privileged	Organizational	Trusted Network (IA-2)	Single-factor
Low	Non-Privileged	Organizational	Untrusted Network (IA-2)	Single-factor
Low	Non-Privileged	Non-Organizational	Local (IA-8)	Single-factor
Low	Non-Privileged	Non-Organizational	Trusted Network (IA-8)	Single-factor
Low	Non-Privileged	Non-Organizational	Untrusted Network (IA-8/ e-authentication)	Single-factor

System Security Level *(Defined in Table 5)*	User Role *(Defined in Section 3.2.3)*	User Type *(Defined in Section 3.2.3)*	Access Method *(Defined in Section 3.2.3)* and Applicable ARS Control Requirement	Authentication Required *(Defined in Section 2.3.1)*
Low	Privileged	Organizational	Local (IA-2)	Multi-factor
Low	Privileged	Organizational	Trusted Network (IA-2)	Multi-factor
Low	Privileged	Organizational	Untrusted Network (IA-2)	Multi-factor
Low	Privileged	Non-Organizational	*< Any >*	*< Not allowed >*[43]
Moderate	Non-Privileged	Organizational	Local (IA-2)	Multi-factor
Moderate	Non-Privileged	Organizational	Trusted Network (IA-2)	Multi-factor
Moderate	Non-Privileged	Organizational	Untrusted Network (IA-2)	Multi-factor
Moderate	Non-Privileged	Non-Organizational	Local (IA-8)	Multi-factor[44]
Moderate	Non-Privileged	Non-Organizational	Trusted Network (IA-8)	Multi-factor[44]
Moderate	Non-Privileged	Non-Organizational	Untrusted Network (IA-8/ e-authentication)	Multi-factor[44]
Moderate	Privileged	Organizational	Local (IA-2)	Multi-factor
Moderate	Privileged	Organizational	Trusted Network (IA-2)	Multi-factor
Moderate	Privileged	Organizational	Untrusted Network (IA-2)	Multi-factor
Moderate	Privileged	Non-Organizational	*< Any >*	*< Not allowed >*
High	Non-Privileged	Organizational	Local (IA-2)	Multi-factor
High	Non-Privileged	Organizational	Trusted Network (IA-2)	Multi-factor
High	Non-Privileged	Organizational	Untrusted Network (IA-2)	Multi-factor
High	Non-Privileged	Non-Organizational	Local (IA-8)	Multi-factor
High	Non-Privileged	Non-Organizational	Trusted Network (IA-8)	Multi-factor
High	Non-Privileged	Non-Organizational	Untrusted Network (IA-8/ e-authentication)	Multi-factor
High	Privileged	Organizational	Local (IA-2)	Multi-factor
High	Privileged	Organizational	Trusted Network (IA-2)	Multi-factor
High	Privileged	Organizational	Untrusted Network (IA-2)	Multi-factor
High	Privileged	Non-Organizational	*< Any >*	*< Not allowed >*

[43] *Privileged* access for *non-organizational* users is not allowed. All users requiring *privileged* access are treated as *organizational* users.

[44] *May* only require E-authentication Level 2 if applicable conditions for PII/PHI (user can only see information about *themselves*), and *no other Moderate-level information is present*. See Table 5 for details.

4 MACHINE-TO-MACHINE AUTHENTICATION

4.1 MACHINE-TO-MACHINE CONNECTIONS

As CMS brings more and more applications onto the Internet, it is imperative to ensure the *security* of the applications, and the *integrity* of the information transferred to *and* from them. For *Machine-to-machine* connections, not only must the system be *physically* secure, but incoming and outgoing data must be protected to prevent compromise of CMS information integrity. In order to establish that connection, *each* machine must ensure that it is connecting to a *trusted* machine on the other end.

If CMS were to use only *human user* authentication mechanisms, *machines* would be disconnected from networks whenever no one (human) is logged in. As a result, the enterprise could not do remote server administration, automated updates to antivirus and group policies, etc. Clearly, this is not the case at CMS. Instead, infrastructure elements are allowed to establish direct *trusted* connectivity from one machine to another, and perform trusted transactions. For the purposes of this security standard, the scope of machine-to-machine authentication is defined as authentication of connections established between hardware, applications, or clients, in the absence of human users, or as surrogates of human users.

This security standard does not discuss the much larger issue of determining the policies and trust relationships that must be established between the different components of the CMS enterprise. Those involve regulatory constraints, operational relationships, professional relationships, architectural constraints, and many other non-technical factors. However, machine authentication is a common component of the policy enforcement mechanisms that are put in place to manage and enforce policy decisions. Using machine authentication is not a substitute for establishing appropriate policy, but serves instead as an effective mechanism to enforce policy.

This document provides high-level standards for machine authentication and the related infrastructure. Hardware, applications, and clients following these guidelines will reduce their development, acquisition, deployment, and operational costs.

Authentication mechanisms for humans (see Section 3) have many complex legal, privacy, accreditation, hiring, firing, role/function, and authorization issues that do not apply to machines. The management of *machines* can be handled within the IT organization, and does not require the management of human resources. Instead, the standards for management of machine connections can be condensed down to a single simple requirement—connections must be authenticated on *both* ends of the connection before they are established and considered trusted.

4.2 DIFFERENCE BETWEEN HUMAN AND MACHINE AUTHENTICATION

Human user authentication (technically Human-to-*Machine* authentication) is performed, *by the machine*, for the purpose of protecting the *Confidentiality*, *Integrity*, and *Availability*, of the systems and data *of the machine* and to authenticate and verify the identity any *user* that might present themselves and request access to *the machine*. For these types of transactions, it is *assumed* (by the machine) that the human already *trusts* the machine (since it is the *human* that *initiated* the connection in the first place.) This simplification of the human-machine relationship effectively discounts the threat of an intermediate entity *impersonating* the machine *to the human*, because the initiation (by the human) relies almost exclusively on the assumption that the human *"knows where they are going"*. That is, the human approached the *correct* machine. While this is a concern, there is nothing the *target machine* can do in this scenario, since the human actively solicited a connection with the machine, and the authentic machine is effectively out-of-the-loop, and unawares of any possible misdirection or mistake. So, it is not normally an immediate concern—*for the authentic machine*—for it to authenticate itself back to the *human*, since there are no steps that it can actively take to *prevent* the human from proceeding with a *non*-authentic machine.

However, in *machine-to-machine* authentication, *each* of the machines acts as both *"the user"* (in one direction) and *"the authenticator"* (in the other direction). As a result, the authentication must be performed by both machines, authenticating *each other* (two-way, dual, or mutual authentication.)

4.3 MUTUAL AUTHENTICATION

Mutual authentication is the process of hardware, applications, or clients authenticating themselves to other hardware, applications, or clients, followed by the other resource authenticating itself back to the originator. Mutual authentication is necessary for a trusted connection because each party must be assured of the identity of the other. Mutual authentication helps to prevent *man-in-the-middle* attacks and facilitates non-repudiation of transactions, which is critical for CMS information confidentiality and integrity. Mutual authentication is required for all machine-to-machine connectivity. Typically, machine-to-machine authentication is established using dual-certificate exchanges (via *Public Key Infrastructure [PKI]*), but other mechanisms are also valid, provided that they address the authentication of both ends of the connection through NIST-compliant standards.

4.3.1 PUBLIC KEY INFRASTRUCTURE (PKI)

A *Public Key Infrastructure (PKI)* is a set of components that allow the creation, management, distribution, storage, and revocation of public keys. An advantage of a PKI infrastructure is that all certificates issued can be trusted unless the certificate either appears on a revocation list; fails an online status check, or if its validity period has expired.

For PKI-based authentication, CMS ***must*** ensure that information systems: i) validate certificates by constructing a certification path with status information to an accepted trust anchor; ii) enforce authorized access to the corresponding private key; and iii) map the authenticated identity to the applicable account.

Details of PKI deployment requirements can be found in NIST SP 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure*, and NIST SP 800-15, *Minimum Interoperability Specification for PKI Components*.[45]

4.4 DEVICE AND OTHER HARDWARE PASSWORDS

4.4.1 COMPUTER FIRMWARE PASSWORDS

Almost all computer systems with *Basic Input Output System (BIOS)* firmware come with the ability to password protect it. On many systems, a variety of BIOS passwords can be set. Each type of password provides different protections. For example, BIOS configuration protection passwords are intended to prevent unauthorized changes to the BIOS configuration, such as altering the boot order so that the computer can be booted from removable media. Another example is BIOS boot passwords, which must be entered before the system boots so that unauthorized users cannot boot the computer. However, most motherboard manufacturers have procedures that can be used to remove BIOS passwords and restore the system to a default configuration. In some cases, it requires shorting PINs on the motherboard or removing and replacing a chip on the motherboard, but in other cases, a much simpler procedure is available. For example, many BIOSs include backdoor passwords that will always work; others can be recovered using custom software programs or specific key sequences while the machine is booting. Because of this, BIOS passwords should be considered only a deterrent and do not provide any protection to data on the disk.

A newer technology replacing BIOS firmware is *Extensible Firmware Interface (EFI)*. EFI passwords can be set to protect the system's *configuration*. That is, EFI protects the system from *corruption* or *tampering*. However, like BIOS passwords, EFI passwords can be circumvented by anyone who has physical access to the system. EFI passwords should be considered a deterrent to unauthorized access but not a true form of protection.

The "gold standard" of firmware protection is defined in NIST SP 800-147, *BIOS Protection Guidelines*, NIST SP 800-155, *BIOS Integrity Measurement Guidelines*, and has culminated in the new open BIOS spec called *Universal Extensible Firmware Interface (UEFI)* 2.3.1, considered the first strongly secure boot firmware standard. It requires trusted roots, digital certificates, and digital signatures. To maximize protections against firmware attacks, make sure purchases include devices that have UEFI (2.3.1 or above) enabled.

UEFI secure boot works as the boot process executes. *Each piece of code* verifies that the signature on the *next piece of code* and, if valid, passes execution on to it. The process of

[45] Both NIST SP 800-32 and NIST SP 800-15 are available at http://csrc.nist.gov/publications/PubsSPs.html

verifying the signature involves creating a cryptographic digest of the code, then testing that against a cryptographic signature included with it.

Unfortunately, UEFI 2.3.1 or later requires different chip sets than pre-UEFI devices; if you don't already have a UEFI 2.3.1 device, it will probably take a new purchase to get one. All Microsoft Windows 8- and 2012-certified computers will have UEFI 2.3.1 capability, and the related *Windows Secure Boot* technology built-in.

4.4.2 HARD DRIVE PASSWORDS

Some hard drives support the use of passwords to restrict access to a hard drive. For example, a drive might have a master password (for administrative purposes) and a user password, and it could support two security modes: high security and maximum security. In a managed IT environment, the user of a system would be given the user password, and the master password would be retained by administrative staff. In high security mode, the drive can be unlocked with either the user or master password, and the hard drive passwords can only be removed from the drive after supplying the master password. In maximum-security mode, the drive can only be unlocked with the user password, and the master password can only be used to erase the drive and remove the hard drive passwords (i.e., the drive must be erased before passwords are removed). Unlike BIOS passwords that are stored on a chip on the motherboard, hard drive passwords are stored on the hard drive itself. Even if the disk is moved to a new system, read and write operations cannot be performed on the drive until one of the passwords are entered.

Although hard drive passwords do provide a higher level of security and a more effective deterrent to a casual attacker, there are tools and services available that can retrieve or reset the hard drive passwords, so *they cannot be relied on to provide a high level of security*. A more effective solution to protect hard drives from unauthorized access is the use of a full disk encryption solution, which encrypts all information on the hard drive and only decrypts it if the appropriate authentication is provided.

4.4.3 TRUSTED PLATFORM MODULE (TPM) PASSWORDS

A *Trusted Platform Module (TPM)* chip is a tamper-resistant integrated circuit built into some motherboards that can perform cryptographic operations (including key generation) and protect small amounts of sensitive information, such as passwords and cryptographic keys. Each TPM chip has an owner password, which is used to gain access to and manage the TPM chip. Although the TPM can be shut off by someone with physical access to the system, it cannot be circumvented: access to the TPM cannot be achieved without the owner password. Therefore, it is important to choose a strong password for the TPM owner password and to protect its confidentiality. If the owner password is lost or forgotten, it can be reset by clearing the TPM, but this action also clears all data stored on the TPM. Therefore, either the owner password or the data on the TPM should be backed up to an alternate secure location, after carefully considering and addressing the security considerations implicit in storing these types of sensitive information.

The UEFI secure boot process is not the same as TPM. UEFI secure boot is simply an architecture for loading and verifying signed firmware images, bootloaders, kernels, and modules. However, TPM does relate to UEFI secure boot in one important way—they can both be used to create a *root of trust*.

The TPM method of creating a root of trust is different from UEFI secure boot. As code executes, it too creates a cryptographic digest of the next piece of code but instead of verifying it, it sends it to the TPM, where it's appended to a chain. At any point in the boot process, the TPM holds the current state of this chain, and using a TPM command, you can then sign the current state with a key. The process using a TPM is commonly referred to as *"Trusted"* boot, as opposed to UEFI *"Secure"* boot.

4.4.4 NETWORK INFRASTRUCTURE DEVICE PASSWORDS

The simplest method of authentication for a network infrastructure device, such as a router or switch, is *local* authentication. Authentication credentials are stored *on the device*, and when a user attempts to authenticate, the presented credentials are compared with *stored passwords* or password hashes. Passwords stored on network infrastructure devices are sometimes unencrypted, so physical security controls *must* be applied to protect the passwords from compromise. These devices often have a single administrative account, so if multiple users need to administer a device, *a* ***centralized authentication system*** *should be configured for those network devices with a separate account and password for each administrator to provide accountability*.

Another common method of network device administration is *Simple Network Management Protocol (SNMP)*. SNMP version 1 and version 2 rely on *clear text* community strings, which are used as passwords to grant access to the device. Since SNMP version 1 and version 2 send community strings across the network *with no cryptographic protection*, they should not be used to configure network infrastructure devices over untrusted networks. SNMP version 3 provides security feature enhancements to SNMP, including encryption and message authentication. If any version of SNMP is used for remote administration, default SNMP community strings such as "public" and "private" should be removed before real community strings are put into place. If both are present on the device at any time, an attacker could retrieve real community strings from the device using the default string.

4.4.5 GENERAL-USE OFFICE DEVICE PASSWORDS

Many general-use office devices, such as printers, scanners, and copiers, can be configured to be network accessible. Although security of these devices is not generally considered a high priority, the specific functionality of the devices should be considered before they are installed in a network environment. For example, many modern copiers are multifunction devices that can be used as printers or scanners and contain a whole OS. By default, any documents scanned into the device are stored for retrieval on a network-accessible server. Without proper authentication in place, any user with network access to the device can retrieve all documents stored in the cache. Unless the temporary loss of availability of the device or loss of confidentiality or integrity of information processed on the device will have minimal impact on the organization, default passwords should not be used. In some cases, simple office devices are designed without consideration given to user management. For example, only a single administrative account is provided and a centralized authentication system cannot be used, so user credentials are shared between administrators. Since these passwords must be shared by administrators, they should be dedicated to these devices and should not be used for any other devices.

4.5 TRANSPORT LAYER SECURITY (TLS)

4.5.1 SECURE SOCKETS LAYER (SSL) IS INSECURE

TLS and SSL are protocols that provide data encryption and authentication between applications and servers in scenarios where that data is being sent across an insecure network, such as checking your email. The terms SSL and TLS are often used interchangeably or in conjunction with each other (TLS/SSL), but one is in fact the predecessor of the other—SSL version 3.0 served as the basis for TLS version 1.0 which, as a result, is sometimes referred to as SSL version 3.1.

There are five protocols in the SSL/TLS family: SSL version 2.0, SSL version 3.0, TLS version 1.0, TLS version 1.1, and TLS version 1.2. Of these:

- SSL version 2.0 is *extremely* insecure and ***must not be used*** for the following reasons:
  - Identical cryptographic keys are used for message authentication and encryption.
  - Has a weak MAC construction that uses the MD5 (deprecated) hash function with a secret prefix, making it vulnerable to length extension attacks.
  - Does not have any protection for the handshake, meaning a *man-in-the-middle* downgrade attack can go undetected.
  - Uses the Transmission Control Protocol (TCP) *connection close* to indicate the end of data. This means that truncation attacks are possible: the attacker simply forges a TCP FIN, leaving the recipient unaware of an illegitimate end of data message.
  - Assumes a single service and a fixed domain certificate, which clashes with the standard feature of *virtual hosting* in Web servers. This means that most websites are practically impaired from using SSL version 2.0.
- SSL version 3.0 is insecure. SSL version 3.0 cipher suites have a weaker key derivation process; half of the master key that is established is fully dependent on the MD5 hash function, which is not resistant to collisions and is, therefore, ***not considered secure***. Under TLS version 1.0, the master key that is established depends on both MD5 and SHA-1, so its derivation process is not considered *as* weak. It is for this reason that SSL version 3.0 implementations cannot be validated under FIPS 140-2.

- TLS version 1.0 is insecure. A vulnerability[46] in the way the TLS version 1.0, and SSL version 3.0, protocols select the *initialization vector* when operating in cipher-block chaining (CBC) modes allows an attacker to perform a chosen-plaintext attack on encrypted traffic. This vulnerability has been addressed in the specification for the TLS versions 1.1 and 1.2.
- TLS version 1.1 and 1.2 are without known security issues. TLS version 1.2 is superior because it offers important features that are unavailable in earlier versions (e.g., TLS version 1.2 supports SHA-2 based Hash-based Message Authentication Code [HMAC]).

4.5.2 TLS DESCRIPTION

The TLS protocol is used to secure communications in a wide variety of online transactions. Such transactions include financial transactions (i.e., banking, trading stocks, e-commerce), healthcare transactions (i.e., viewing medical records or scheduling medical appointments), and social transactions (i.e., email or social networking). Any network service that handles sensitive or valuable data, whether it is PII, financial data, or login information, needs to adequately protect that data. TLS provides a protected channel for sending data between the server and the client. The client is often, but not always, a web browser.

TLS is a layered protocol that runs on top of a reliable transport protocol—typically the *TCP*. Application protocols, such as HTTP and Internet Message Access Protocol (IMAP), can run above TLS. TLS is application independent, and used to provide security to any two communicating applications that transmit data over a network via an application protocol. It can be used to create a virtual private network (VPN) that connects an external system to an internal network, allowing that system to access a multitude of internal services and resources as if it were in the network.

There are three subprotocols in the TLS protocol that are used to control the session connection: the *handshake*, *change cipher spec*, and *alert* protocols. The TLS *handshake* protocol is used to negotiate the session parameters. The *alert* protocol is used to notify the other party of an error condition. The *change cipher spec* protocol is used to change the cryptographic parameters of a session. In addition, the client and the server exchange application data that is protected by the security services provisioned by the negotiated cipher suite. These security services are negotiated and established with the handshake.

The *handshake* protocol consists of a series of message exchanges between the client and the server. The handshake protocol initializes both the client and server to use cryptographic capabilities by negotiating a cipher suite of algorithms and functions, including key establishment, digital signature, confidentiality and integrity algorithms. Clients and servers can be configured so that one or more of the following security services are negotiated during the handshake: *confidentiality*, *message integrity*, *authentication*, and *replay protection*. A *confidentiality* service provides assurance that data is kept secret, preventing eavesdropping. A *message integrity* service provides assurance that unauthorized data modification will be

[46] The *BEAST* exploit is purely a *client-side* vulnerability. Since *BEAST* has been released to the public, most major browsers have addressed through patches and upgrades. However, CMS should endeavor not to use TLS version 1.0 (or earlier) in order to protect against unpatched clients.

detected, thus preventing undetected deletion, addition, or modification of data. An *authentication* service provides assurance of the sender or receiver's identity, thereby detecting forgery. *Replay protection* ensures that an unauthorized user does not capture and successfully replay previous data. In order to comply with NIST guidelines, both the client and the server ***shall*** be configured for data *confidentiality* and *integrity* services.

For a full description of how TLS achieves these four services, refer to NIST SP 800-52, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*.[47]

4.5.3 MINIMUM REQUIREMENTS FOR TLS SERVERS

Specific requirements for TLS are stated as either *implementation* requirements or *configuration* requirements. *Implementation* requirements indicate that CMS ***shall not*** procure TLS server implementations unless they include the required functionality, or can be augmented with additional commercial products to meet requirements. *Configuration* requirements indicate that TLS server administrators are ***required*** *to verify that particular features are enabled*, or in some cases, *configured appropriately*.

4.5.3.1 TLS PROTOCOL VERSION SUPPORT

TLS version 1.1 is required, *at a minimum*, in order to mitigate various attacks on version 1.0 of the TLS protocol. Support for TLS version 1.2 is *strongly*[48] recommended.[49]

Servers that support ***government-only*** applications ***shall*** be configured to support TLS 1.1, and ***should*** be configured to support TLS 1.2. These servers ***shall not*** support TLS 1.0 or ***any*** version of SSL.

Servers that support ***citizen or business-facing*** applications ***shall*** be configured to support version 1.1 and ***should*** be configured to support version 1.2. These servers *may* also be configured to support TLS version 1.0 in order to enable interaction with citizens and businesses. These servers ***shall not*** support SSL version 3.0 or earlier. If TLS version 1.0 is supported, the use of TLS version 1.1 and 1.2 ***shall*** be preferred over TLS version 1.0.

Some server implementations are known to implement version negotiation incorrectly. For example, there are TLS version 1.0 servers that terminate the connection when the client offers a

[47] NIST SP 800-52, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*, is available at http://csrc.nist.gov/publications/PubsSPs.html.

[48] NIST SP 800-52 recommends that agencies develop plans to support TLS version 1.2, configured using approved schemes and algorithms, by January 1, 2015.

[49] The main reason for moving to the newer versions of TLS is the compromising, and thus elimination, of one or more of 3DES/TDEA, MD5, and/or SHA-1. TLS version 1.1 drops the mandatory TLS version 1.0 cipher TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA in favor of TLS_RSA_WITH_3DES_EDE_CBC_SHA, and TLS version 1.2 drops that in favor of TLS_RSA_WITH_AES_128_CBC_SHA. TLS versions after TLS version 1.0 still support MD5, but since SHA-1 is deprecated, and may not be used, there are clear advantages to TLS version 1.1 and later including less dependence on MD5/SHA-1 and better support for arbitrary cipher or hash functions by way of extensions.

version newer than TLS version 1.0, or some servers will refuse the connection if any TLS extensions are included in *ClientHello*[50]. Servers that incorrectly implement TLS version negotiation ***shall not*** be used.

4.5.3.2 TLS SERVER KEYS AND CERTIFICATES

The TLS server ***shall*** be configured with one or more public key certificates and the associated private keys. TLS server implementations ***should*** support multiple server certificates with their associated private keys to support algorithm and key size agility.

There are six options for TLS server certificates that can satisfy the requirement for Approved cryptography: an RSA key encipherment certificate; an RSA signature certificate; an Elliptic Curve Digital Signature Algorithm (ECDSA) signature certificate; a *Digital Signature Algorithm (DSA)*[51] signature certificate; a Diffie-Hellman certificate; and an Elliptic curve Diffie–Hellman (ECDH) certificate.

At a minimum, TLS servers conforming to this specification ***shall*** be configured with an RSA key encipherment certificate, and also ***should*** be configured with an ECDSA signature certificate or RSA signature certificate. If the server is not configured with an RSA signature certificate, an ECDSA signature certificate using a Suite B named curve for the signature and public key in the ECDSA certificate ***should*** be used.[52]

TLS servers ***shall*** be configured with certificates *issued by a CA*, rather than self-signed certificates. Furthermore, TLS server certificates ***shall*** be issued by a CA that publishes revocation information in either a Certificate Revocation List (CRL) [RFC 5280] or in Online Certificate Status Protocol (OCSP) [RFC 6960][53] responses. The source for the revocation information ***shall*** be included in the CA-issued certificate in the appropriate extension to promote interoperability.

A TLS server that has been issued certificates by multiple CAs can select the appropriate certificate, based on the client specified *Trusted CA Keys* TLS extension. A TLS server that has been issued certificates for multiple names can select the appropriate certificate, based on the client specified *Server Name* TLS extension. A TLS server may also contain multiple names in the *Subject Alternative Name* extension of the server certificate in order to support multiple server names of the same name form (e.g., DNS Name) or multiple server names of multiple name forms (e.g., DNS Names, IP Address, etc.)

[50] Earlier versions of the TLS specification were not fully clear on what the record layer version number (*TLSPlaintext.version*) should contain when sending *ClientHello*. Thus, TLS servers compliant with the TLS specification ***must*** accept any value {03,XX} as the record layer version number for *ClientHello*. This ***should*** be the latest (highest valued) version supported by the client.

[51] In the names for the TLS cipher suites, *Digital Signature Algorithm (DSA)* is referred to as *Digital Signature Standard (DSS),* for historical reasons.

[52] The Suite B curves are known as P-256 and P-384. These curves are defined in [FIPS 186-4] and their inclusion in Suite B is documented in [RFC 6460].

[53] RFC 6960, *X.509 Internet Public Key Infrastructure, Online Certificate Status Protocol - OCSP,* is available at http://tools.ietf.org/html/rfc6960.

NIST SP 800-52, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*[54] specifies a detailed profile for server certificates. Basic guidelines for DSA, DH, and ECDH certificates are provided. NIST SP 800-52 also specifies requirements for revocation checking. System administrators ***shall*** use these standards to identify an appropriate source for certificates.

4.6 LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP)

4.6.1 LDAP DESCRIPTION

The *Lightweight Directory Access Protocol (LDAP)* is an application protocol for accessing and maintaining distributed directory information services over an *Internet Protocol (IP)* network. LDAP acts as a *directory service* that is responsible for managing access to the resources of a network. It is sort of a "telephone directory" for the resources within a network. LDAP is called lightweight because it is a smaller and easier protocol, derived from the X.500 DAP (Directory Access Protocol) defined in the Open Systems Interconnection (OSI) network protocol stack.

LDAP includes a list of all *servers*, *clients*, *users*, *groups*, and *shared resources*, such as (shared) *files*, and *printers*. Clients (i.e., *human users* and/or *machines*, etc.) can interact with directory service resources through LDAP by using authentication that is a minimum of a *User ID* and *password*. LDAP operates over TCP ports 389 (*unencrypted* data transfers) and 636 when using TLS for providing *encrypted* data transfers. LDAP is specified in a series of *Internet Engineering Task Force* (IETF) *Standard Track Request for Comments* (RFCs). The latest specification is Version 3, published as RFC 4511[55].

The most common usage of LDAP is to provide a "single sign-on" where one password for an account-holder is shared between many services, such as applying an enterprise login code to web pages (so that users log in only once to enterprise computers, and then are automatically logged into the enterprise intranet.)

4.6.2 LDAP THREATS

Basic threats to an LDAP directory service include, but are not limited to:

- Unauthorized access to directory data via data-retrieval operations.
- Unauthorized access to directory data by monitoring access of others.
- Unauthorized access to reusable client authentication information by monitoring access of others.
- Unauthorized modification of directory data.

[54] NIST SP 800-52, *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*, is available at http://csrc.nist.gov/publications/PubsSPs.html.
[55] RFC 4511, *Lightweight Directory Access Protocol (LDAP): The Protocol*, is available at http://tools.ietf.org/html/rfc4511.

- Unauthorized modification of configuration information.
- Denial of Service: Use of resources (commonly in excess) in a manner intended to deny service to others.
- Spoofing:
    - Tricking a user or client into believing that information came from the directory when in fact it did not—either by modifying data in transit, or misdirecting the client's transport connection.
    - Tricking a user or client into sending privileged information to a hostile entity that appears to be the directory server but is not.
    - Tricking a directory server into believing that information came from a particular client when in fact it came from a hostile entity.
- Hijacking: An attacker seizes control of an established protocol session.

Threats 1, 4, 5, 6, 7, and 8 are *active* attacks. Threats 2 and 3 are *passive* attacks. Threats 1, 4, 5, and 6 are due to *hostile clients*. Threats 2, 3, 7, and 8 are due to *hostile agents* on the path *between client and server,* ***or*** hostile agents *posing as a server* (e.g., IP spoofing.)

4.6.3 LDAP SECURITY

The LDAP protocol suite can be protected with the following security mechanisms:

- Client authentication by means of the *Simple Authentication and Security Layer (SASL)* mechanism set, possibly backed by the TLS credentials exchange mechanism,
- Client authorization by means of access control based on the requestor's authenticated identity,
- Data integrity protection by means of the TLS protocol or data-integrity SASL mechanisms,
- Protection against snooping by means of the TLS protocol or data-encrypting SASL mechanisms,
- Resource limitation by means of administrative limits on service controls, and
- Server authentication by means of the TLS protocol or SASL mechanism.

Simply allowing implementations to pick-and-choose the security mechanisms that will be implemented is not a strategy that leads to interoperability, or security. In the absence of mandates, clients will continue to be developed that do not support any security function supported by the server, or worse, they will only support mechanisms that provide inadequate security for most circumstances.

LDAP can also allow for *Proxy authentication*, where a process such as a web app is acting on behalf of many users, but *caution is advised*: if an application is given the ability to act for *any* user then the consequences of a successful exploit will likely be unacceptable. Wherever possible, LDAP operations performed by an application should be done *using the credentials of the user* that triggered them.

During the LDAP client's establishment of a session with the server (a bind operation), there are three possible levels of *authentication* that may be negotiated:

1. ***No Authentication*** - This mode of operation would be applicable to a read-only directory, containing no sensitive data, accessible to "anyone", and TCP connection hijacking or IP spoofing is not a problem. This is generally discouraged in a complex multi-zone environment.
2. ***Simple Bind*** - In this mode, the contents of the bind API's password parameter would be sent in *clear text*. However, ***clear text*** *passwords should* ***never*** *be sent* without some form of NIST-compliant encryption provided by a lower layer protocol. ***(Note that "Base64 encoding" is considered to be clear text.)***
3. ***SASL***- The *Simple Authentication and Security Layer (SASL)* mode allows the use of any method or mechanism defined by the SASL framework. Although SASL allows the selection from a half dozen security mechanisms (including Kerberos, S/Key, Generic Security Service Application Program Interface [GSSAPI], Challenge Response Authentication Mechanism [CRAM]-MD5, TLS, and ANONYMOUS), TLS is the most widely accepted for use with LDAP version 3.

Because some of the above LDAP authentication mechanisms transmit credentials in *plaintext* form (i.e., *unencrypted*), or do not provide data security services, or are subject to passive attacks, it is necessary to ensure secure interoperability by identifying a *mandatory-to-implement* mechanism for establishing TLS services. As Microsoft's *Active Directory* continues to gain momentum as a primary user authentication directory, more application developers are requesting the use of LDAP for user authentication within their applications. By default *Microsoft domain controllers* **do not** *provide a secure method for third-party connections when using LDAP*. This can create a false sense of security and the potential for loss of confidentiality. CMS does not allow the transmission or storage of authentication credentials in (unencrypted) *plaintext*. The CMS security standard for LDAP requires the implementation of LDAP over TLS to secure communication between application servers and domain controllers.

4.6.4 LDAP TLS

Any LDAP server products used within CMS are required to support *TLS* to support authentication. Most LDAP products will *also* support the older SSL encryption. However, SSL has been deprecated (by NIST) for several years and there are known attacks against it that ***will not be fixed***—therefore TLS (vice SSL) should be used.

The correct and standard TLS approach is to start LDAP *without encryption* and then negotiate the TLS security layer. If necessary, the server can be configured to refuse all operations other than '*Start TLS'* until TLS is in place. It may still be necessary to permit at least the root DSE[56] to be read without TLS protection, as many LDAP clients need to read that to detect the server's ability to do TLS at all.

[56] The *Root DSE* is the top-level *Directory System Agent (**D**SA)* ***S****pecific* ***E****ntry* in a local directory server.

One important function of TLS is to provide *proof* to the client that it has connected to the *correct server,* and that there is no *man-in-the-middle* attack in progress. To achieve this protection it is vital for all client systems to have trustworthy copies of the appropriate X.509 *Certificate Authority (CA)* certificate, and for them to implement the correct validation checks during TLS setup.

Once TLS is in place on the connection, the client should *reread* the root DSE and any other information that it plans to rely on. Servers may give different answers on secure connections, and in any case, it is unwise to trust any information received over an unprotected link.

If *Simple Bind* authentication is in use, then TLS *must* be used, to prevent exposure of passwords on the network. As LDAP is often used to validate passwords for other services, *Simple Binding* is likely to be a very common situation. CMS servers should ***disallow** the use of passwords when TLS is not in use*. Very few server products have this standard as their default settings, so adhering to this requirement usually requires manual configuration to implement. Most *Security Technical Implementation Guides (STIGs)* enforce these settings in their base configurations. *Simple Bind* authentication choice is normally ***not** suitable* for authentication on *untrusted networks* (such as the Internet) where there is no network or transport layer confidentiality.

Where possible (and *preferred* at CMS), more secure mechanisms based on SASL should be used. SASL EXTERNAL[57] along with client-side certificates and TLS provide the most comprehensive protection, but require the creation and management of an X.509 certificate for each *"user"*. X.509 certificates may be purchased from a commercial *Certificate Authority (CA)*, or they can be locally generated and maintained. The EXTERNAL SASL mechanism may be used to request the LDAP server make use of security credentials exchanged by a lower layer. However, if a TLS session has not been established between the client and server *prior to making this SASL EXTERNAL Bind request*, and there is *no other external source of authentication credentials* (e.g., IP-level security), or if, during the process of establishing the TLS session, the *server did not request the client's authentication credentials*, the SASL EXTERNAL bind *must fail*.

4.6.5 LDAP PASSWORDS

Most LDAP systems store and validate passwords—mainly because, for many LDAP implementations, it is their *primary function*. Servers normally default to storing passwords in clear text (not desired), or in a, encrypted form that can be converted back to clear text (preferred). Wherever possible, passwords should be stored using a nonreversible *cryptographic hash* including a significant amount of *salt*. This provides the best possible protection against the recovery of passwords from stolen disks or backup tapes. FIPS-compliant implementation of SHA-2 is the best commonly implemented hash at present. Note that passwords protected using

[57] The EXTERNAL mechanism allows a client to request the server to use credentials established by means external to the mechanism to authenticate the client. The external means may be, for instance, IP Security [RFC 4301] or TLS [RFC 4346] services. In absence of some a priori agreement between the client and the server, the client cannot make any assumption as to what external means the server has used to obtain the client's credentials, nor make an assumption as to the form of credentials. For example, the client cannot assume that the server will use the credentials the client has established via TLS.

Advanced Encryption Standard (AES) and other symmetric algorithms are likely to be recoverable from stolen media with very little effort as the encryption keys are often likely to be present on the same media.

4.6.6 LDAP INJECTION

LDAP injection is a specific form of attack that can be employed to compromise websites that construct LDAP statements from data provided by users. This is done by changing LDAP statements so dynamic Web applications can run with invalid permissions, allowing the attacker to alter, add, or delete content.

LDAP injection works in much the same manner as *Structured Query Language (SQL)* injection, a type of security exploit in which the attacker adds SQL code to a Web form input box to gain access to resources or make changes to data. The main reason that LDAP injection (and similar exploits) are on the rise is the fact that security is not sufficiently emphasized in application development. To protect the integrity of websites and applications, experts recommend the implementation of simple precautions during development, such as controlling the types and numbers of characters that are accepted by input boxes. At the very least, asterisks, logical (AND "&", OR "|" and NOT "!") and relational (=, >=, <=,~=) operators must be filtered at the application layer.

4.7 INTERNET PROTOCOL SECURITY (IPSEC)

4.7.1 IPSEC OVERVIEW

Internet Protocol Security (IPsec) is a framework of open standards that can be used for encrypting TCP/IP traffic within networking environments. IPsec is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing *mutual authentication* between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used in protecting data flows between a pair of hosts (host-to-host), between a pair of security gateways (network-to-network), or between a security gateway and a host (network-to-host). This in turn provides network level data integrity, data confidentiality, data origin authentication, and replay protection. To secure data moving over the intranet, extranet, and Internet, IPsec can be used. IPsec can also be used to secure remote access connections.

A few security features provided by IPsec are listed here:

- *Authentication*; a digital signature is used to verify the identity of the sender of the information. IPsec can use Kerberos, a preshared key, or digital certificates for authentication.
- *Data integrity*; a hash algorithm is used to ensure that data is not tampered with. A checksum called a *Hash Message Authentication Code (HMAC)* is calculated for the data of the packet.

- *Data privacy*; encryption algorithms are utilized to ensure that data being transmitted is undecipherable.
- *Anti-replay*; prevents an attacker from resending packets in an attempt to gain access to the private network.
- *Nonrepudiation;* public key digital signatures are used to prove message origin.
- *Dynamic rekeying*; keys can be created during data sending to protect segments of the communication with different keys.
- *Key generation*; the Diffie-Hellman key agreement algorithm is used to enable two computers to exchange a shared encryption key.
- *IP Packet filtering*; the packet filtering capability of IPsec can be used to filter and block specific types of traffic, based on either of the following elements or on a combination of them:
  - IP addresses
  - Protocols
  - Ports

IPsec supports two encryption modes: *Transport* and *Tunnel*. Transport mode encrypts only the data portion (payload) of each packet, but leaves the header untouched. The more secure Tunnel mode encrypts both the header and the payload. On the receiving side, an IPsec-compliant device decrypts each packet. Tunnel mode is used to form a traditional VPN, where the tunnel generally creates a secure tunnel across an untrusted network.

For IPsec to work, the sending and receiving devices must share a public key. This is accomplished through a protocol known as *Internet Security Association and Key Management Protocol/Oakley (ISAKMP/Oakley)*, which allows the receiver to obtain a public key and authenticate the sender using digital certificates.

Authentication Header (AH) and *Encapsulating Security Payload (ESP)* are the two main wire-level protocols used by IPsec, and they authenticate (AH) and encrypt+authenticate (ESP) the data flowing over that connection. They are typically used independently, though it is possible (but uncommon) to use them both together.

4.7.2 IPSEC CRYPTOGRAPHY

Setting up an IPsec connection involves many crypto choices, but this is simplified substantially by the fact that any given connection can use at most two, or (rarely) three, at a time.

Authentication calculates an *Integrity Check Value (ICV)* over the packet's contents, and is usually built on top of a cryptographic hash such as MD5 (not FIPS compliant), SHA-1 (deprecated), or SHA-2[58] (preferred). *Authentication* incorporates a secret key known to both ends, and this allows the recipient to compute the ICV in the same way. If the recipient gets the same value, the sender has effectively authenticated itself (relying on the property that cryptographic hashes cannot practically be reversed). AH always provides authentication, and ESP does so optionally.

Encryption uses a secret key to encrypt the data before transmission, and this hides the actual contents of the packet from eavesdroppers. There are quite a few choices for algorithms available in IPsec, with AES being the NIST-compliant solution.

4.7.3 IPSEC AUTHENTICATION

Authentication deals with verifying the identity of the computer sending the data, or the identity of the computer receiving the data. The methods which IPsec can use to authenticate the sender or receiver of data are:

- *Digital certificates*: Provides the *most secure* means of authenticating identities. A public key certificate should be used in situations that include Internet access, remote access to enterprise resources, external business partner communications, or computers that do not run the Kerberos (version 5 or better) authentication protocol. This requires that at least one trusted certification authority (CA) has been configured.
- *Kerberos authentication*: A downside of using the Kerberos authentication protocol is that the identity of the computer remains unencrypted up to the point that the whole payload is encrypted at authentication.
- *Preshared keys*: You should ***only*** use preshared keys when none of the former authentication methods can be used. The preshared key is normally stored in plaintext and is not considered a secure method. Preshared keys should ***only*** be used for testing purposes.

Because the preshared keys method is considered the *least* secure supported authentication method, you should only use preshared keys when you cannot use the digital certificates or the Kerberos (version 5 or better) authentication protocol. Preshared keys should only be used in testing environments.

[58] SHA-2 is a cryptographic hash function similar to MD5 (not NIST-compliant) and SHA-1 (NIST-deprecated) and it generates a 224, 256, 384 or 512-bit message digest or, in other words, a hash value from a variable length input depending upon the function used. Similar to SHA-1, SHA-2 is used for implementation under secure protocols, namely TLS, SSL, PGP, IPSec & S/MIME. SHA-2 is being enforced by the US government for implementation at the national level for all government projects and the private sector has also been encouraged to adopt the SHA-2 version of hashing as it is the most secure to date.

5 APPROVED

Teresa Fryer

Digitally signed by Teresa Fryer
DN: cn=Teresa Fryer, o, ou, email=teresa.fryer@cms.hhs.gov, c=US
Date: 2014.04.29 15:33:17 -04'00'

Teresa Fryer
CMS Chief Information Security Officer and
Director, Enterprise Information Security Group

This document will be reviewed periodically, but no less than annually, by the EISG, and updated as necessary to reflect changes in policy or process. If you have any questions regarding the accuracy, completeness, or content of this document, please contact the EISG at mailto:ciso@cms.hhs.gov.