NIST Special Publication
NIST SP 800-63B-4 ipd

Digital Identity Guidelines

Authentication and Lifecycle Management

Initial Public Draft

David Temoshok
James L. Fenton
Yee-Yin Choong
Naomi Lefkovitz
Andrew Regenscheid
Justin P. Richer

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-63b-4.ipd

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication
NIST SP 800-63B-4 ipd

Digital Identity Guidelines

Authentication and Lifecycle Management

Initial Public Draft

David Temoshok
Naomi Lefkovitz
Applied Cybersecurity Division
Information Technology Laboratory

Yee-Yin Choong
Information Access Division
Information Technology Laboratory

Andrew Regenscheid
Computer Security Division
Information Technology Laboratory

James L. Fenton
Altmode Networks

Justin P. Richer
Bespoke Engineering

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-63b-4.ipd

December 2022

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Authority

NIST Technical Series Policies

Copyright, Fair Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History

Approved by the NIST Editorial Review Board on YYYY-MM-DD [will be added upon final publication]

How to Cite this NIST Technical Series Publication

Temoshok D, Fenton JL, Choong YY, Lefkovitz N, Regenscheid A, Richer JP (2022) Digital Identity Guidelines: Authentication and Lifecycle Management. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-63B-4 ipd. https://doi.org/10.6028/NIST.SP.800-63b-4.ipd

Author ORCID iDs

David Temoshok: 0000-0001-6195-0331
James L. Fenton: 0000-0002-2344-4291
Yee-Yin Choong: 0000-0002-3889-6047
Naomi Lefkovitz: 0000-0003-3777-3106
Andrew Regenscheid: 0000-0002-3930-527X
Justin P. Richer: 0000-0003-2130-5180

Public Comment Period

December 16, 2022 - ~~March 24~~ April 14, 2023

Submit Comments

mailto:dig-comments@nist.gov

All comments are subject to release under the Freedom of Information Act (FOIA).

Reports on Computer Systems Technology

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

Abstract

These guidelines provide technical requirements for federal agencies implementing digital identity services and are not intended to constrain the development or use of standards outside of this purpose. These guidelines focus on the authentication of subjects interacting with government information systems over networks, establishing that a given claimant is a subscriber who has been previously authenticated. The result of the authentication process may be used locally by the system performing the authentication or may be asserted elsewhere in a federated identity system. This document defines technical requirements for each of the three authenticator assurance levels. This publication will supersede NIST Special Publication (SP) 800-63B.

Keywords

authentication; authentication assurance; credential service provider; digital authentication; digital credentials; electronic authentication; electronic credentials; passwords.

Note to Reviewers

The rapid proliferation of online services over the past few years has heightened the need for reliable, equitable, secure, and privacy-protective digital identity solutions.

Revision 4 of NIST Special Publication 800-63 *Digital Identity Guidelines* intends to respond to the changing digital landscape that has emerged since the last major revision of this suite was published in 2017 — including the real-world implications of online risks. The guidelines present the process and technical requirements for meeting digital identity management assurance levels for identity proofing, authentication, and federation, including requirements for security and privacy as well as considerations for fostering equity and the usability of digital identity solutions and technology.

Taking into account feedback provided in response to our June 2020 Pre-Draft Call for Comments, as well as research conducted into real-world implementations of the guidelines, market innovation, and the current threat environment, this draft seeks to:

1. **Advance Equity:** This draft seeks to expand upon the risk management content of previous revisions and specifically mandates that agencies account for impacts to individuals and communities in addition to impacts to the organization. It also elevates risks to mission delivery – including challenges to providing services to all people who are eligible for and entitled to them – within the risk management process and when implementing digital identity systems. Additionally, the guidance now mandates continuous evaluation of potential impacts across demographics, provides biometric performance requirements, and additional parameters for the responsible use of biometric-based technologies, such as those that utilize face recognition.
2. **Emphasize Optionality and Choice for Consumers:** In the interest of promoting and investigating additional scalable, equitable, and convenient identify verification options, including those that do and do not leverage face recognition technologies, this draft expands the list of acceptable identity proofing alternatives to provide new mechanisms to securely deliver services to individuals with differing means, motivations, and backgrounds. The revision also emphasizes the need for digital identity services to support multiple authenticator options to address diverse consumer needs and secure account recovery.
3. **Deter Fraud and Advanced Threats:** This draft enhances fraud prevention measures from the third revision by updating risk and threat models to account for new attacks, providing new options for phishing resistant authentication, and introducing requirements to prevent automated attacks against enrollment processes. It also opens the door to new technology such as mobile driver's licenses and verifiable credentials.
4. **Address Implementation Lessons Learned:** This draft addresses areas where implementation experience has indicated that additional clarity or detail was required to effectively operationalize the guidelines. This includes re-working the federation assurance levels, providing greater detail on trusted referees, clarifying guidelines on identity attribute validation sources, and improving address confirmation requirements.

NIST is specifically interested in comments on and recommendations for the following topics:

Authentication and Lifecycle Management

- Are emerging authentication models and techniques – such as FIDO passkey, verifiable credentials, and mobile driver's licenses – sufficiently addressed and accommodated, as appropriate, by the guidelines? What are the potential associated security, privacy, and usability benefits and risks?
- Are the controls for phishing resistance as defined in the guidelines for AAL2 and AAL3 authentication clear and sufficient?
- How are session management thresholds and reauthentication requirements implemented by agencies and organizations? Should NIST provide thresholds or leave session lengths to agencies based on applications, users, and mission needs?
- What impacts would the proposed biometric performance requirements for this volume have on real-world implementations of biometric technologies?

General

- Is there an element of this guidance that you think is missing or could be expanded?
- Is any language in the guidance confusing or hard to understand? Should we add definitions or additional context to any language?
- Does the guidance sufficiently address privacy?
- Does the guidance sufficiently address equity?
  – What equity assessment methods, impact evaluation models, or metrics could we reference to better support organizations in preventing or detecting disparate impacts that could arise as a result of identity verification technologies or processes?
- What specific implementation guidance, reference architectures, metrics, or other supporting resources may enable more rapid adoption and implementation of this and future iterations of the Digital Identity Guidelines?
- What applied research and measurement efforts would provide the greatest impact on the identity market and advancement of these guidelines?

Reviewers are encouraged to comment and suggest changes to the text of all four draft volumes of of the NIST SP 800-63-4 suite. NIST requests that all comments be submitted by 11:59pm Eastern Time on March 24, 2023. Please submit your comments to dig-comments@nist.gov. NIST will review all comments and make them available at the NIST Identity and Access Management website. Commenters are encouraged to use the comment template provided on the NIST Computer Security Resource Center website.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

    i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or

    ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: mailto:dig-comments@nist.gov.

Table of Contents

List of Tables

List of Figures

Acknowledgments

The authors would like to thank their fellow collaborators on the current revision of this special publication, Christine Abruzzi, Ryan Galluzzo, Sarbari Gupta, Connie LaSalle, and Diana Proud-Madruga, as well as Kerrianne Buchanan and Greg Fiumara for their contributions and review. The authors would like to also acknowledge the past contributions of Donna F. Dodson, W. Timothy Polk, Emad A. Nabbus, Paul A. Grassi, Elaine M. Newton, Ray Perlner, William E. Burr, Kristen K. Greene, Mary F. Theofanos, Kaitlin Boeckl, Kat Megas, Ellen Nadeau, Ben Piccarreta, and Danna Gabel O'Rourke.

1. Purpose

This section is informative.

This publication and its companion volumes, [SP800-63], [SP800-63A], and [SP800-63C], provide technical guidelines to organizations for the implementation of digital identity services.

This document, SP 800-63B, provides requirements to credential service providers (CSPs) for remote user authentication at each of three authentication assurance levels (AALs).

2. Introduction

This section is informative.

Digital authentication is the process of determining the validity of one or more authenticators used to claim a digital identity. Authentication establishes that a subject attempting to access a digital service is in control of the technologies used to authenticate. For services in which return visits are applicable, successfully authenticating provides reasonable risk-based assurances that the subject accessing the service today is the same as the one who accessed the service previously.

The ongoing authentication of subscribers is central to the process of associating a subscriber with their online activity (i.e., with their *subscriber account*). Subscriber authentication is performed by verifying that the claimant controls one or more *authenticators* (called *tokens* in some earlier versions of SP 800-63) associated with a given subscriber account. A successful authentication results in the assertion of a pseudonymous or non-pseudonymous identifier and optionally other identity information to the relying party (RP).

This document provides recommendations on types of authentication processes, including choices of authenticators, that may be used at various *authentication assurance levels* (AALs). It also provides recommendations on the lifecycle of authenticators, including revocation in the event of loss or theft.

This technical guideline applies to digital authentication of subjects to systems over a network. It does not address the authentication of a person for physical access (e.g., to a building), though some credentials used for digital access may also be used for physical access authentication. This technical guideline also requires that federal systems and service providers participating in authentication protocols be authenticated to subscribers.

The AAL characterizes the strength of an authentication transaction as an ordinal category. Stronger authentication (a higher AAL) requires malicious actors to have better capabilities and to expend greater resources in order to successfully subvert the authentication process. Authentication at higher AALs can effectively reduce the risk of attacks. A high-level summary of the technical requirements for each of the AALs is provided below; see Sec. 4 and Sec. 5 of this document for specific normative requirements.

Authentication Assurance Level 1: AAL1 provides some assurance that the claimant controls an authenticator bound to the subscriber account. AAL1 requires either single-factor or multi-factor authentication using a wide range of available authentication technologies. Successful authentication requires that the claimant prove possession and control of the authenticator through a secure authentication protocol.

Authentication Assurance Level 2: AAL2 provides high confidence that the claimant controls one or more authenticators bound to the subscriber account. Proof of possession and control of two different authentication factors is required through secure authentication protocols. Approved cryptographic techniques are required at AAL2 and above.

Authentication Assurance Level 3: AAL3 provides very high confidence that the claimant controls one or more authenticators bound to the subscriber account. Authentication at AAL3 is based on proof of possession of a key through a cryptographic protocol. AAL3 authentication requires a hardware-based authenticator and a phishing-resistant authenticator (see Sec. 5.2.5); the same device may fulfill both these requirements. In order to authenticate at AAL3, claimants are required to prove possession and control of two distinct authentication factors through secure authentication protocols. Approved cryptographic techniques are required.

The following list states which sections of the document are normative and which are informative:

- 1 Purpose *Informative*
- 2 Introduction *Informative*
- 3 Definitions and Abbreviations *Informative*
- 4 Authentication Assurance Levels *Normative*
- 5 Authenticator and Verifier Requirements *Normative*
- 6 Authenticator Lifecycle Management *Normative*
- 7 Session Management *Normative*
- 8 Threat and Security Considerations *Informative*
- 9 Privacy Considerations *Informative*
- 10 Usability Considerations *Informative*
- 11 Equity Considerations *Informative*
- References *Informative*
- Appendix A Strength of Memorized Secrets *Informative*
- Appendix B Change Log *Informative*

3. Definitions and Abbreviations

See [SP800-63], Appendix A for a complete set of definitions and abbreviations.

4. Authentication Assurance Levels

This section is normative.

To satisfy the requirements of a given AAL and be recognized as a subscriber, a claimant **SHALL** be authenticated with a process whose strength is equal to or greater than the requirements at that level. The result of an authentication process is an identifier that **SHALL** be used each time that subscriber authenticates to that RP. The identifier **MAY** be pseudonymous. Subscriber identifiers **SHOULD NOT** be reused for a different subject but **SHOULD** be reused when a previously enrolled subject is re-enrolled by the CSP. Other attributes that identify the subscriber as a unique subject **MAY** also be provided.

Detailed normative requirements for authenticators and verifiers at each AAL are provided in Sec. 5.

See [SP800-63] Sec. 5 for details on how to choose the most appropriate AAL.

[FIPS140] requirements are satisfied by FIPS 140-3 or newer revisions.

Personal information collected during and subsequent to identity proofing **MAY** be made available to the subscriber by the digital identity service. The release or online availability of any PII or other personal information, whether self-asserted or validated, by federal government agencies requires multi-factor authentication in accordance with [EO13681]. Therefore, federal government agencies **SHALL** select a minimum of AAL2 when PII or other personal information is made available online.

4.1. Authentication Assurance Level 1

AAL1 provides some assurance that the claimant controls an authenticator bound to the subscriber account. AAL1 requires either single-factor or multi-factor authentication using a wide range of available authentication technologies. Successful authentication requires that the claimant prove possession and control of the authenticator through a secure authentication protocol.

4.1.1. Permitted Authenticator Types

AAL1 authentication **SHALL** occur by the use of any of the following authenticator types, which are defined in Sec. 5:

- Memorized secret (Sec. 5.1.1)
- Look-Up secret (Sec. 5.1.2)
- Out-of-band device (Sec. 5.1.3)
- Single-factor one-time password (OTP) device (Sec. 5.1.4)
- Multi-factor OTP device (Sec. 5.1.5)
- Single-factor cryptographic software (Sec. 5.1.6)

- Single-factor cryptographic device (Sec. 5.1.7)
- Multi-factor cryptographic software (Sec. 5.1.8)
- Multi-factor cryptographic device (Sec. 5.1.9)

4.1.2. Authenticator and Verifier Requirements

Cryptographic authenticators used at AAL1 **SHALL** use approved cryptography. Software-based authenticators that operate within the context of an operating system **MAY**, where applicable, attempt to detect compromise (e.g., by malware) of the user endpoint in which they are running and **SHOULD NOT** complete the operation when such a compromise is detected.

Communication between the claimant and verifier **SHALL** be via an authenticated protected channel to provide confidentiality of the authenticator output and resistance to adversary-in-the-middle (AitM) attacks.

Verifiers operated by or on behalf of federal government agencies at AAL1 **SHALL** be validated to meet the requirements of [FIPS140] Level 1.

4.1.3. Reauthentication

Periodic reauthentication of subscriber sessions **SHALL** be performed as described in Sec. 7.2. At AAL1, reauthentication of the subscriber **SHOULD** be repeated at least once per 30 days during an extended usage session, regardless of user activity. The session **SHOULD** be terminated (i.e., logged out) when this time limit is reached.

4.1.4. Security Controls

The CSP **SHALL** employ appropriately tailored security controls from the baseline security controls defined in [SP800-53] or equivalent federal (e.g., [FEDRAMP]) or industry standard that the organization has determined for the information systems, applications, and online services that these guidelines are used to protect. The CSP **SHALL** ensure that the minimum assurance-related controls for the appropriate systems, or equivalent, are satisfied.

4.1.5. Records Retention Policy

The CSP **SHALL** comply with its respective records retention policies in accordance with applicable laws, regulations, and policies, including any National Archives and Records Administration (NARA) records retention schedules that may apply. If the CSP opts to retain records in the absence of any mandatory requirements, the CSP **SHALL** conduct a risk management process, including assessments of privacy and security risks, to determine how long records should be retained and **SHALL** inform the subscriber of that retention policy.

4.2. Authentication Assurance Level 2

AAL2 provides high confidence that the claimant controls authenticators bound to the subscriber account. Proof of possession and control of two distinct authentication factors is required through secure authentication protocols. Approved cryptographic techniques are required at AAL2 and above.

4.2.1. Permitted Authenticator Types

At AAL2, authentication **SHALL** occur by the use of either a multi-factor authenticator or a combination of two single-factor authenticators. A multi-factor authenticator requires two factors to execute a single authentication event, such as a cryptographically secure device with an integrated biometric sensor that is required to activate the device. Authenticator requirements are specified in Sec. 5.

When a multi-factor authenticator is used, any of the following **MAY** be used:

- Multi-Factor Out-of-Band Authenticator (Sec. 5.1.3.4)
- Multi-Factor OTP Device (Sec. 5.1.5)
- Multi-Factor Cryptographic Software (Sec. 5.1.8)
- Multi-Factor Cryptographic Device (Sec. 5.1.9)

When a combination of two single-factor authenticators is used, the combination **SHALL** include a Memorized Secret authenticator (Sec. 5.1.1) and one physical authenticator (i.e., "something you have") from the following list:

- Look-Up Secret (Sec. 5.1.2)
- Out-of-Band Device (Sec. 5.1.3)
- Single-Factor OTP Device (Sec. 5.1.4)
- Single-Factor Cryptographic Software (Sec. 5.1.6)
- Single-Factor Cryptographic Device (Sec. 5.1.7)

> Note: When biometric authentication meets the requirements in Sec. 5.2.3, the device has to be authenticated in addition to the biometric match. A biometric characteristic is recognized as a factor, but not recognized as an authenticator by itself. Therefore, when conducting authentication with a biometric characteristic, it is unnecessary to use two authenticators because the associated device serves as "something you have," while the biometric match serves as "something you are."

4.2.2. Authenticator and Verifier Requirements

Cryptographic authenticators used at AAL2 **SHALL** use approved cryptography. Authenticators procured by federal government agencies **SHALL** be validated to meet the requirements of [FIPS140] Level 1. Software-based authenticators that operate within the context of an operating system **MAY**, where applicable, attempt to detect compromise (e.g., by malware) of the platform in which they are running. They **SHOULD NOT** complete the operation when such a compromise is detected. At least one authenticator used at AAL2 **SHALL** be replay resistant as described in Sec. 5.2.8. Authentication at AAL2 **SHOULD** demonstrate authentication intent from at least one authenticator as discussed in Sec. 5.2.9.

Communication between the claimant and verifier **SHALL** be via an authenticated protected channel to provide confidentiality of the authenticator output and resistance to AitM attacks.

Verifiers operated by or on behalf of federal government agencies at AAL2 **SHALL** be validated to meet the requirements of [FIPS140] Level 1.

When a biometric factor is used in authentication at AAL2, the performance requirements stated in Sec. 5.2.3 **SHALL** be met, and the verifier **SHOULD** make a determination that the biometric sensor and subsequent processing meet these requirements.

OMB Memorandum [M-22-09] requires federal government agencies to offer at least one phishing-resistant authenticator option to public users at AAL2. While phishing resistance as described in Sec. 5.2.5 is not generally required for authentication at AAL2, verifiers **SHOULD** encourage the use of phishing-resistant authenticators at AAL2 whenever practical since phishing is a significant threat vector.

4.2.3. Reauthentication

Periodic reauthentication of subscriber sessions **SHALL** be performed as described in Sec. 7.2. At AAL2, authentication of the subscriber **SHALL** be repeated at least once per 12 hours during an extended usage session, regardless of user activity. Reauthentication of the subscriber **SHALL** be repeated following any period of inactivity lasting 30 minutes or longer. The session **SHALL** be terminated (i.e., logged out) when either of these time limits is reached.

Reauthentication of a session that has not yet reached its time limit **MAY** require only a memorized secret or a biometric in conjunction with the still-valid session secret. The verifier **MAY** prompt the user to cause activity just before the inactivity timeout.

4.2.4. Security Controls

The CSP **SHALL** employ appropriately tailored security controls from the baseline security controls defined in [SP800-53] or equivalent federal (e.g., [FEDRAMP]) or industry standard that the organization has determined for the information systems, applications, and online services that these guidelines are used to protect. The CSP **SHALL** ensure that the minimum assurance-related controls for the appropriate systems, or equivalent, are satisfied.

4.2.5. Records Retention Policy

The CSP **SHALL** comply with its respective records retention policies in accordance with applicable laws, regulations, and policies, including any NARA records retention schedules that may apply. If the CSP opts to retain records in the absence of any mandatory requirements, the CSP **SHALL** conduct a risk management process, including assessments of privacy and security risks to determine how long records should be retained and **SHALL** inform the subscriber of that retention policy.

4.3. Authentication Assurance Level 3

AAL3 provides very high confidence that the claimant controls authenticators bound to the subscriber account. Authentication at AAL3 is based on proof of possession of a key through a cryptographic protocol. AAL3 authentication **SHALL** use a hardware-based authenticator and an authenticator that provides phishing resistance — the same device **MAY** fulfill both these requirements. In order to authenticate at AAL3, claimants **SHALL** prove possession and control of two distinct authentication factors through secure authentication protocols. Approved cryptographic techniques are required.

4.3.1. Permitted Authenticator Types

AAL3 authentication **SHALL** occur by the use of one of a combination of authenticators satisfying the requirements in Sec. 4.3. Possible combinations are:

- Multi-Factor Cryptographic Device (Sec. 5.1.9)
- Single-Factor Cryptographic Device (Sec. 5.1.7) used in conjunction with a Memorized Secret (Sec. 5.1.1)
- Multi-Factor OTP device (software or hardware) (Sec. 5.1.5) used in conjunction with a Single-Factor Cryptographic Device (Sec. 5.1.7)
- Multi-Factor OTP device (hardware only) (Sec. 5.1.5) used in conjunction with a Single-Factor Cryptographic Software (Sec. 5.1.6)
- Single-Factor OTP device (hardware only) (Sec. 5.1.4) used in conjunction with a Multi-Factor Cryptographic Software Authenticator (Sec. 5.1.8)

4.3.2. Authenticator and Verifier Requirements

Communication between the claimant and verifier **SHALL** be via an authenticated protected channel to provide confidentiality of the authenticator output and resistance to AitM attacks. At least one cryptographic authenticator used at AAL3 **SHALL** be phishing resistant as described in Sec. 5.2.5 and **SHALL** be replay resistant as described in Sec. 5.2.8. All authentication and reauthentication processes at AAL3 **SHALL** demonstrate authentication intent from at least one authenticator as described in Sec. 5.2.9.

Multi-factor authenticators used at AAL3 **SHALL** be hardware cryptographic modules validated at [FIPS140] Level 2 or higher overall with at least [FIPS140] Level 3 physical security. Single-factor cryptographic devices used at AAL3 **SHALL** be validated at [FIPS140] Level 1 or higher overall with at least [FIPS140] Level 3 physical security.

Verifiers at AAL3 **SHALL** be validated at [FIPS140] Level 1 or higher.

Verifiers at AAL3 **SHALL** be verifier compromise resistant as described in Sec. 5.2.7 with respect to at least one authentication factor.

Hardware-based authenticators and verifiers at AAL3 **SHOULD** resist relevant side-channel (e.g., timing and power-consumption analysis) attacks.

When a biometric factor is used in authentication at AAL3, the verifier **SHALL** make a determination that the biometric sensor and subsequent processing meet the performance requirements stated in Sec. 5.2.3.

4.3.3. Reauthentication

Periodic reauthentication of subscriber sessions **SHALL** be performed as described in Sec. 7.2. At AAL3, authentication of the subscriber **SHALL** be repeated at least once per 12 hours during an extended usage session, regardless of user activity, as described in Sec. 7.2. Reauthentication of the subscriber **SHALL** be repeated following any period of inactivity lasting 15 minutes or longer. Reauthentication **SHALL** use both authentication factors. The session **SHALL** be terminated (i.e., logged out) when either of these time limits is reached. The verifier **MAY** prompt the user to cause activity just before the inactivity timeout.

4.3.4. Security Controls

The CSP **SHALL** employ appropriately tailored security controls from the baseline security controls defined in [SP800-53] or equivalent federal (e.g., [FEDRAMP]) or industry standard that the organization has determined for the information systems, applications, and online services that these guidelines are used to protect. The CSP **SHALL** ensure that the minimum assurance-related controls for the appropriate systems, or equivalent, are satisfied.

4.3.5. Records Retention Policy

The CSP **SHALL** comply with its respective records retention policies in accordance with applicable laws, regulations, and policies, including any NARA records retention schedules that may apply. If the CSP opts to retain records in the absence of any mandatory requirements, the CSP **SHALL** conduct a risk management process, including assessments of privacy and security risks, to determine how long records should be retained and **SHALL** inform the subscriber of that retention policy.

4.4. Privacy Requirements

The CSP **SHALL** employ appropriately tailored privacy controls defined in [SP800-53] or equivalent industry standard.

If CSPs process attributes for purposes other than identity proofing, authentication, or attribute assertions (collectively "identity service"), related fraud mitigation, or to comply with law or legal process, CSPs **SHALL** implement measures to maintain predictability and manageability commensurate with the privacy risk arising from the additional processing. Measures **MAY** include providing clear notice, obtaining subscriber consent, or enabling selective use or disclosure of attributes. When CSPs use consent measures, CSPs **SHALL NOT** make consent for the additional processing a condition of the identity service.

Regardless of whether the CSP is an agency or private sector provider, the following requirements apply to a federal agency offering or using the authentication service:

1. The agency **SHALL** consult with their Senior Agency Official for Privacy (SAOP) and conduct an analysis to determine whether the collection of PII to issue or maintain authenticators triggers the requirements of the *Privacy Act of 1974* [PrivacyAct] (see Sec. 9.4).
2. The agency **SHALL** publish a System of Records Notice (SORN) to cover such collections, as applicable.
3. The agency **SHALL** consult with their SAOP and conduct an analysis to determine whether the collection of PII to issue or maintain authenticators triggers the requirements of the *E-Government Act of 2002* [E-Gov].
4. The agency **SHALL** publish a Privacy Impact Assessment (PIA) to cover such collection, as applicable.

4.5. Summary of Requirements

Table 1 provides a non-normative summary of the requirements for each of the AALs.

Table 1. AAL Summary of Requirements

Requirement	**AAL1**	**AAL2**	**AAL3**
Permitted authenticator types	Memorized Secret; Look-up Secret; Out-of-Band; SF OTP Device; MF OTP Device; SF Crypto Software; SF Crypto Device; MF Crypto Software; MF Crypto Device	MF Out-of-Band; MF OTP Device; MF Crypto Software; MF Crypto Device; or Memorized Secret plus: Look-up Secret, Out-of-Band, SF OTP Device, SF Crypto Software, SF Crypto Device	MF Crypto Device; SF Crypto Device plus Memorized Secret; SF OTP Device plus MF Crypto Device or Software; SF OTP Device plus SF Crypto Software plus Memorized Secret
FIPS 140 validation	Level 1 (Government agency verifiers)	Level 1 (Government agency authenticators and verifiers)	Level 2 overall (MF authenticators) Level 1 overall (verifiers and SF Crypto Devices) Level 3 physical security (all authenticators)
Reauthentication	30 days	12 hours or 30 minutes inactivity; one authentication factor	12 hours or 15 minutes inactivity; both authentication factors
Security controls	[SP800-53] Low Baseline (or equivalent)	[SP800-53] Moderate Baseline (or equivalent)	[SP800-53] High Baseline (or equivalent)
AitM resistance	Required	Required	Required
Phishing resistance	Not required	Recommended	Required
Verifier-compromise resistance	Not required	Not required	Required
Replay resistance	Not required	Required	Required
Authentication intent	Not required	Recommended	Required

5. Authenticator and Verifier Requirements

This section is normative.

This section provides the detailed requirements specific to each type of authenticator. With the exception of reauthentication requirements specified in Sec. 4 and the requirement for phishing resistance at AAL3 described in Sec. 5.2.5, the technical requirements for each of the authenticator types are the same regardless of the AAL at which the authenticator is used.

5.1. Requirements by Authenticator Type

5.1.1. Memorized Secrets

A Memorized Secret authenticator — commonly referred to as a *password* or, if numeric, a *PIN* — is a secret value intended to be chosen and memorized by the user. Memorized secrets need to be of sufficient complexity and secrecy that it would be impractical for an attacker to guess or otherwise discover the correct secret value. A memorized secret is *something you know*.

The requirements in this section apply to centrally verified memorized secrets that are used as an independent authentication factor, sent over an authenticated protected channel to the verifier of a CSP. Memorized secrets that are used locally by a multi-factor authenticator are referred to as *activation secrets* and discussed in Sec. 5.2.11.

5.1.1.1. Memorized Secret Authenticators

Memorized secrets **SHALL** be at least 8 characters in length. Memorized secrets **SHALL** be either chosen by the subscriber or assigned randomly by the CSP.

If the CSP disallows a chosen memorized secret because it is on a blocklist of commonly used, expected, or compromised values (see Sec. 5.1.1.2), the subscriber **SHALL** be required to choose a different memorized secret. No other complexity requirements for memorized secrets **SHALL** be imposed. A rationale for this is presented in Appendix A *Strength of Memorized Secrets*.

5.1.1.2. Memorized Secret Verifiers

Verifiers **SHALL** require memorized secrets to be at least 8 characters in length. Verifiers **SHOULD** permit memorized secrets to be at least 64 characters in length. All printing ASCII [RFC20] characters as well as the space character **SHOULD** be acceptable in memorized secrets. Unicode [ISO/ISC 10646] characters **SHOULD** be accepted as well. Verifiers **MAY** make allowances for likely mistyping, such as removing leading and trailing whitespace characters prior to verification or allowing verification of memorized secrets with differing case for the leading character, provided memorized secrets remain at least 8 characters in length after such processing.

Verifiers **SHALL** verify the entire submitted memorized secret (i.e., not truncate the secret). For purposes of the above length requirements, each Unicode code point **SHALL** be counted as a single character.

If Unicode characters are accepted in memorized secrets, the verifier **SHOULD** apply the normalization process for stabilized strings using either the NFKC or NFKD normalization defined in Sec. 12.1 of *Unicode Normalization Forms* [UAX15]. This process is applied before hashing the byte string representing the memorized secret. Subscribers choosing memorized secrets containing Unicode characters **SHOULD** be advised that some characters may be represented differently by some endpoints, which can affect their ability to authenticate successfully.

Memorized secret verifiers **SHALL NOT** permit the subscriber to store a hint that is accessible to an unauthenticated claimant. Verifiers **SHALL NOT** prompt subscribers to use specific types of information (e.g., "What was the name of your first pet?", a technique known as knowledge-based authentication (KBA) or security questions) when choosing memorized secrets.

When processing requests to establish and change memorized secrets, verifiers **SHALL** compare the prospective secrets against a blocklist that contains values known to be commonly used, expected, or compromised. For example, the list **MAY** include, but is not limited to:

- Passwords obtained from previous breach corpuses.
- Dictionary words.
- Repetitive or sequential characters (e.g. 'aaaaaa', '1234abcd').
- Context-specific words, such as the name of the service, the username, and derivatives thereof.

If the chosen secret is found in the blocklist, the CSP or verifier **SHALL** advise the subscriber that they need to select a different secret, **SHALL** provide the reason for rejection, and **SHALL** require the subscriber to choose a different value. Since the blocklist is used to defend against brute-force attacks and unsuccessful attempts are rate limited as described below, the blocklist **SHOULD** be of a size sufficient to prevent subscribers from choosing memorized secrets that attackers are likely to guess before reaching the attempt limit. Excessively large blocklists **SHOULD NOT** be used because they frustrate subscribers' attempts to establish an acceptable memorized secret and do not provide significantly improved security.

Verifiers **SHALL** offer guidance to the subscriber to assist the user in choosing a strong memorized secret. This is particularly important following the rejection of a memorized secret on the above list as it discourages trivial modification of listed (and likely very weak) memorized secrets [Blocklists].

Verifiers **SHALL** implement a rate-limiting mechanism that effectively limits the number of failed authentication attempts that can be made on the subscriber account as described in Sec. 5.2.2.

Verifiers **SHALL NOT** impose other composition rules (e.g., requiring mixtures of different character types or prohibiting consecutively repeated characters) for memorized secrets. Verifiers **SHALL NOT** require users to periodically change memorized secrets. However, verifiers **SHALL** force a change if there is evidence of compromise of the authenticator.

Verifiers **SHALL** allow the use of password managers. To facilitate their use, verifiers **SHOULD** permit claimants to use "paste" functionality when entering a memorized secret. Password manangers may increase the likelihood that users will choose stronger memorized secrets.

In order to assist the claimant in successfully entering a memorized secret, the verifier **SHOULD** offer an option to display the secret — rather than a series of dots or asterisks — while it is entered and until it is submitted to the verifier. This allows the claimant to confirm their entry if they are in a location where their screen is unlikely to be observed. The verifier **MAY** also permit the claimant's device to display individual entered characters for a short time after each character is typed to verify correct entry. This is common on mobile devices.

The verifier **SHALL** use approved encryption and an authenticated protected channel when requesting memorized secrets in order to provide resistance to eavesdropping and adversary-in-the-middle attacks.

Verifiers **SHALL** store memorized secrets in a form that is resistant to offline attacks. Memorized secrets **SHALL** be salted and hashed using a suitable password hashing scheme. Password hashing schemes take a password, a salt, and a cost factor as inputs and generate a password hash. Their purpose is to make each password guess more expensive for an attacker who has obtained a hashed password file and thereby make the cost of a guessing attack high or prohibitive. A function that is both memory-hard and compute-hard **SHOULD** be used because it increases the cost of an attack. While NIST has not published guidelines on specific password hashing schemes, examples of such functions include Argon2 [Argon2] and scrypt [Scrypt]. Examples of approved one-way functions include Keyed Hash Message Authentication Code (HMAC) [FIPS198-1], any approved hash function in [SP800-107], Secure Hash Algorithm 3 (SHA-3) [FIPS202], CMAC [SP800-38B], Keccak Message Authentication Code (KMAC), Customizable SHAKE (cSHAKE), and ParallelHash [SP800-185]. The chosen output length of the password hashing scheme **SHOULD** be the same as the length of the underlying one-way function output.

The salt **SHALL** be at least 32 bits in length and be chosen arbitrarily so as to minimize salt value collisions among stored hashes. Both the salt value and the resulting hash **SHALL** be stored for each memorized secret authenticator.

For the Password-based Key Derivation Function 2 (PBKDF2) [SP800-132], the cost factor is an iteration count: the more times the PBKDF2 function is iterated, the longer it takes to compute the password hash. Therefore, the iteration count **SHOULD** be as large as verification server performance will allow, typically at least 10,000 iterations.

In addition, verifiers **SHOULD** perform an additional iteration of a keyed hashing or encryption operation using a secret key known only to the verifier. This key value, if used, **SHALL** be generated by an approved random bit generator [SP800-90Ar1] and provide at least the minimum security strength specified in the latest revision of NIST SP 800-131A, *Transitioning the Use of Cryptographic Algorithms and Key Lengths* [SP800-131A] (112 bits as of the date of this publication). The secret key value **SHALL** be stored separately from the hashed memorized secrets (e.g., in a specialized device like a hardware security module). With this additional iteration, brute-force attacks on the hashed memorized secrets are impractical as long as the secret key value remains secret.

5.1.2. Look-Up Secrets

A look-up secret authenticator is a physical or electronic record that stores a set of secrets shared between the claimant and the CSP. The claimant uses the authenticator to look up the appropriate secrets needed to respond to a prompt from the verifier. For example, the verifier could ask a claimant to provide a specific subset of the numeric or character strings printed on a card in table format. A common application of look-up secrets is the use of one-time “recovery keys” stored by the subscriber for use in the event another authenticator is lost or malfunctions. A look-up secret is *something you have*.

5.1.2.1. Look-Up Secret Authenticators

CSPs creating look-up secret authenticators **SHALL** use an approved random bit generator [SP800-90Ar1] to generate the list of secrets and **SHALL** deliver the authenticator securely to the subscriber. Look-up secrets **SHALL** have at least 20 bits of entropy.

Look-up secrets **MAY** be distributed by the CSP in person, by postal mail to the subscriber’s address of record, or by online distribution. If distributed online, look-up secrets **SHALL** be distributed over a secure channel in accordance with the post-enrollment binding requirements in Sec. 6.1.2.

If the authenticator uses look-up secrets sequentially from a list, the subscriber **MAY** dispose of used secrets, but only after a successful authentication.

5.1.2.2. Look-Up Secret Verifiers

Verifiers of look-up secrets **SHALL** prompt the claimant for the next secret from their authenticator or for a specific (e.g., numbered) secret. A given secret from an authenticator **SHALL** be used successfully only once. If the look-up secret is derived from a grid card, each cell of the grid **SHALL** be used only once.

Verifiers **SHALL** store look-up secrets in a form that is resistant to offline attacks. Look-up secrets having at least 112 bits of entropy **SHALL** be hashed with an approved one-way function as described in Sec. 5.1.1.2. Look-up secrets with fewer than 112 bits of entropy **SHALL** be salted and hashed using a suitable password hashing scheme, also described in Sec. 5.1.1.2. The salt value **SHALL** be at least 32 bits in length and arbitrarily chosen so as to minimize salt value collisions among stored hashes. Both the salt value and the resulting hash **SHALL** be stored for each look-up secret.

For look-up secrets that have less than 64 bits of entropy, the verifier **SHALL** implement a rate-limiting mechanism that effectively limits the number of failed authentication attempts that can be made on the subscriber account as described in Sec. 5.2.2.

The verifier **SHALL** use approved encryption and an authenticated protected channel when requesting look-up secrets in order to provide resistance to eavesdropping and AitM attacks.

5.1.3. Out-of-Band Devices

An out-of-band authenticator is a physical device that is uniquely addressable and can communicate securely with the verifier over a distinct communications channel, referred to as the secondary channel. The device is possessed and controlled by the claimant and supports private communication over this secondary channel, separate from the primary channel for authentication. An out-of-band authenticator is *something you have*.

Out-of-band authentiction uses a short-term secret generated by the verifier. The secret's purpose is to securely bind the authentication operation on the primary and secondary channel and establishes the claimant's control of the out-of-band device.

The out-of-band authenticator can operate in one of the following ways:

- The claimant transfers a secret received by the out-of-band device via the secondary channel to the verifier using the primary channel. For example, the claimant may receive the secret (typically a 6-digit code) on their mobile device and type it into their authentication session. This method is shown in Figure 1.

- The claimant transfers a secret received via the primary channel to the out-of-band device for transmission to the verifier via the secondary channel. For example, the claimant may view the secret on their authentication session and either type it into an app on their mobile device or use a technology such as a barcode or QR code to effect the transfer. This method is shown in Figure 2.

  Note: A third method of out-of-band authentication involving the comparison of secrets received from the primary and secondary channels and approving on the secondary channel is no longer considered acceptable because it was rarely implemented as described. It raised the likelihood that the claimant would just approve without actually comparing the secrets. For example,

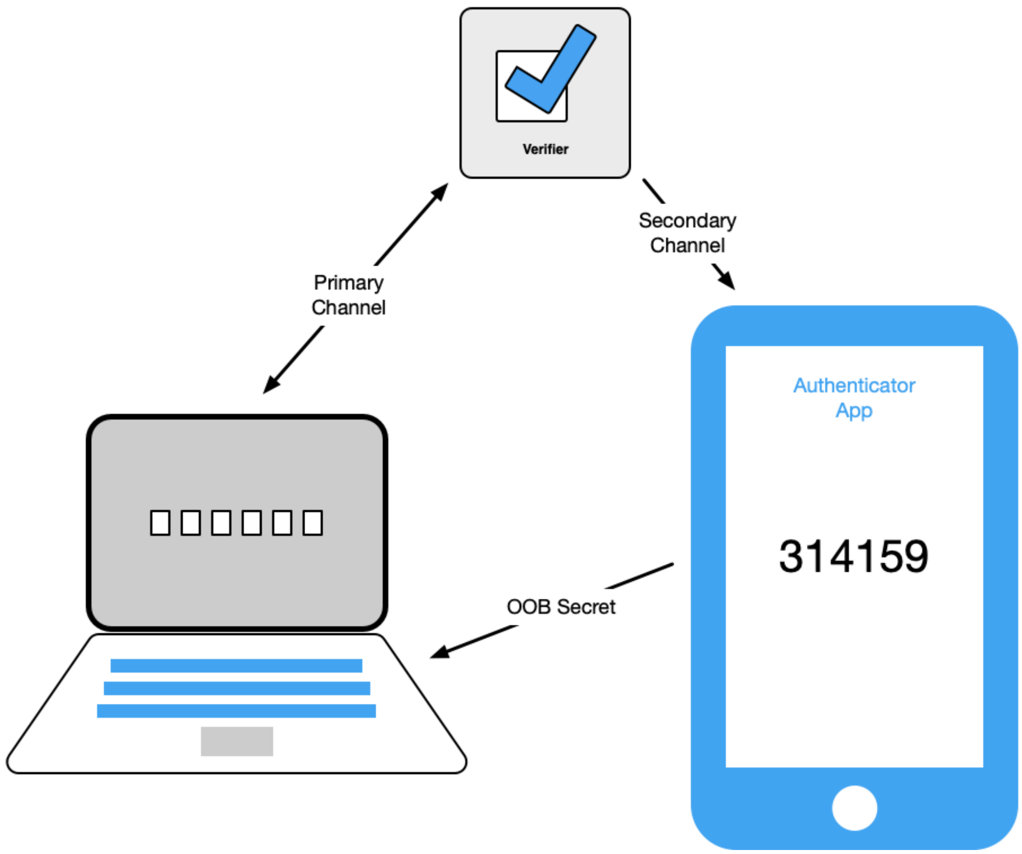

Figure 1. Transfer of Secret to Primary Device

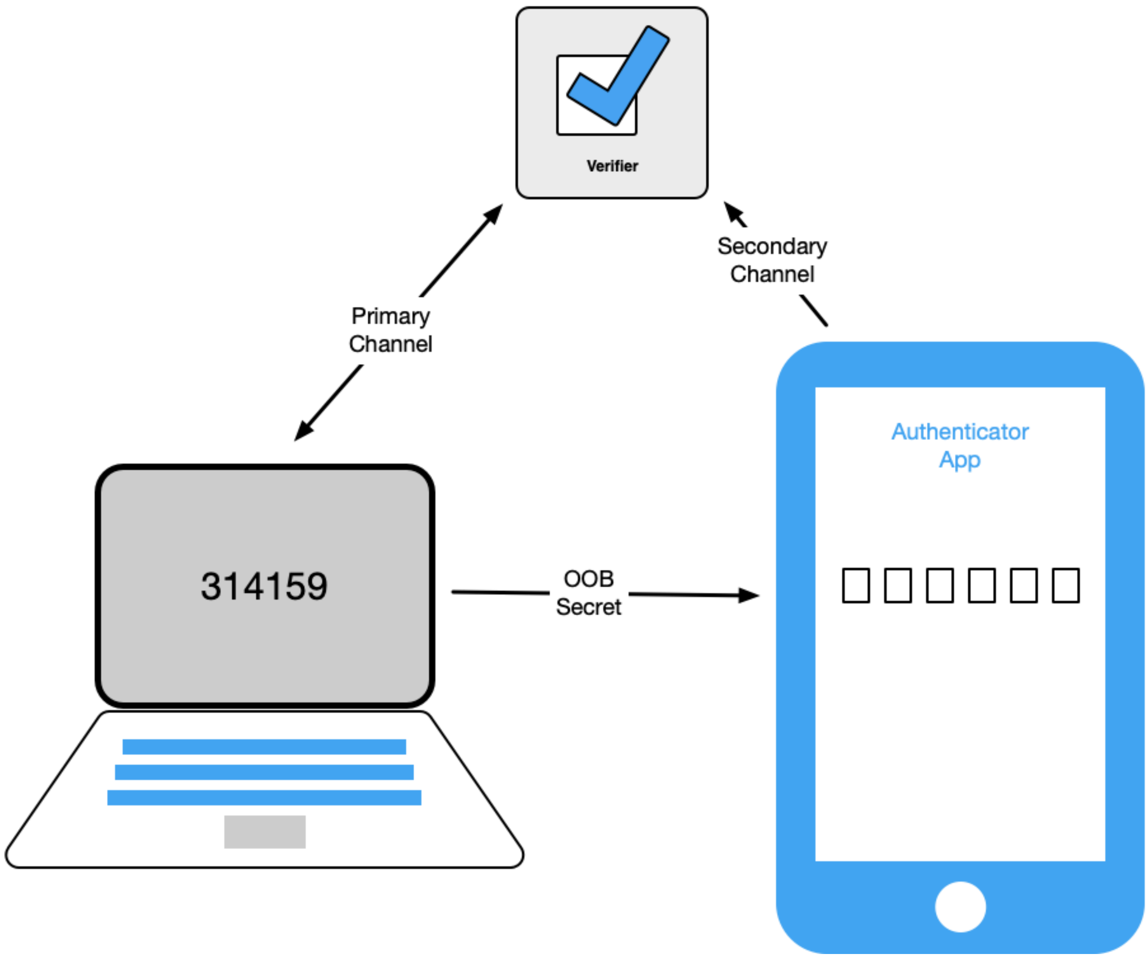

Figure 2. Transfer of Secret to Out-of-band Device

an authenticator that receives a push notification from the verifier and simply asks the claimant to approve the transaction (even if providing some additional information about the authentication) does not meet the requirements of this section.

5.1.3.1. Out-of-Band Authenticators

The out-of-band authenticator **SHALL** establish a separate channel with the verifier in order to retrieve the out-of-band secret or authentication request. This channel is considered to be out-of-band with respect to the primary communication channel (even if it terminates on the same device) provided the device does not leak information from one channel to the other without the authorization of the claimant.

The out-of-band device **SHOULD** be uniquely addressable by the verifier. Communication over the secondary channel **SHALL** be encrypted unless sent via the public switched telephone network (PSTN). For additional authenticator requirements specific to use of the PSTN for out-of-band authentication, see Sec. 5.1.3.3. Channels or addresses that do not prove possession of a specific device, such as voice-over-IP (VOIP) telephone numbers, **SHALL NOT** be used for out-of-band authentication.

Email **SHALL NOT** be used for out-of-band authentication because it also does not prove possession of a specific device and is typically accessed using only a memorized secret.

The out-of-band authenticator **SHALL** uniquely authenticate itself in one of the following ways when communicating with the verifier:

- Establish an authenticated protected channel to the verifier using approved cryptography. The key used **SHALL** be stored in suitably secure storage available to the authenticator application (e.g., keychain storage, TPM, TEE, secure element).
- Authenticate to a public mobile telephone network using a SIM card or equivalent that uniquely identifies the device. This method **SHALL** only be used if a secret is being sent from the verifier to the out-of-band device via the PSTN (SMS or voice).

If a secret is sent by the verifier to the out-of-band device, the device **SHOULD NOT** display the authentication secret while it is locked by the owner (i.e., **SHOULD** require the presentation and verification of a PIN, passcode, or biometric characteristic to view). However, authenticators **SHOULD** indicate the receipt of an authentication secret on a locked device.

If the out-of-band authenticator requests approval over the secondary communication channel — rather than by the presenting a secret that the claimant transfers to the primary communication channel — it **SHALL** accept transfer of the secret from the primary channel and send it to the verifier over the secondary channel to associate the approval with the authentication transaction. The claimant **MAY** perform the transfer manually or use a technology such as a barcode or QR code to effect the transfer.

5.1.3.2. Out-of-Band Verifiers

For additional verification requirements specific to the PSTN, see Sec. 5.1.3.3.

When the out-of-band authenticator is a secure application, such as on a smart phone, the verifier **MAY** send a push notification to that device. The verifier waits for the establishment of an authenticated protected channel with the out-of-band authenticator and verifies its identifying key. The verifier **SHALL NOT** store the identifying key itself, but **SHALL** use a verification method (e.g., an approved hash function or proof of possession of the identifying key) to uniquely identify the authenticator. Once authenticated, the verifier transmits the authentication secret to the authenticator.

Depending on the type of out-of-band authenticator, one of the following **SHALL** take place:

- Transfer of secret from the secondary to the primary channel: The verifier **MAY** signal the device containing the subscriber's authenticator to indicate readiness to authenticate. It **SHALL** then transmit a random secret to the out-of-band authenticator. The verifier **SHALL** then wait for the secret to be returned on the primary communication channel.
- Transfer of secret from the primary to the secondary channel: The verifier **SHALL** display a random authentication secret to the claimant via the primary channel. It **SHALL** then wait for the secret to be returned on the secondary channel from the claimant's out-of-band authenticator.

In all cases, the authentication **SHALL** be considered invalid if not completed within 10 minutes. In order to provide replay resistance as described in Sec. 5.2.8, verifiers **SHALL** accept a given authentication secret only once during the validity period.

The verifier **SHALL** generate random authentication secrets with at least 20 bits of entropy using an approved random bit generator [SP800-90Ar1]. If the authentication secret has less than 64 bits of entropy, the verifier **SHALL** implement a rate-limiting mechanism that effectively limits the number of failed authentication attempts that can be made on the subscriber account as described in Sec. 5.2.2.

Out-of-band verifiers **SHALL** consider all authentication operations to be single-factor unless the CSP has confirmed that the out-of-band authentication meets the requirements of Sec. 5.1.3.4. This requirement **MAY** be satisfied by issuance of the authenticator by the CSP or a trusted third party or by use of an authentication application known by the CSP to meet these requirements.

Out-of-band verifiers that send a push notification to a subscriber device **SHOULD** implement a reasonable limit on the rate or total number of push notifications that will be sent since the last successful authentication.

5.1.3.3. Authentication using the Public Switched Telephone Network

Use of the PSTN for out-of-band verification is restricted as described in this section and in Sec. 5.2.10. If out-of-band verification is to be made using the PSTN, the verifier **SHALL** verify that the pre-registered telephone number being used is associated with a specific physical device. Changing the pre-registered telephone number is considered to be the binding of a new authenticator and **SHALL** only occur as described in Sec. 6.1.2.

Use of the PSTN to deliver out-of-band authentication secrets is potentially not available to some subscribers in areas with limited telephone coverage (particularly in areas without mobile phone service). Accordingly, verifiers **SHALL** ensure that alternative authenticator types are available to all subscribers and **SHOULD** remind subscribers of this limitation of PSTN out-of-band authenticators prior to binding.

Verifiers **SHOULD** consider risk indicators such as device swap, SIM change, number porting, or other abnormal behavior before using the PSTN to deliver an out-of-band authentication secret.

> NOTE: Consistent with the restriction of authenticators in Sec. 5.2.10, NIST may adjust the restricted status of the PSTN over time based on the evolution of the threat landscape and the technical operation of the PSTN.

5.1.3.4. Multi-Factor Out-of-Band Authenticators

Multi-factor out-of-band authenticators operate in a similar manner to single-factor out-of-band authenticators (see Sec. 5.1.3.1) except that they require the presentation and verification of an additional factor, either a memorized secret or a biometric characteristic, prior to allowing the claimant to complete the authentication transaction (i.e., prior to accessing the authentication secret, entering the authentication secret, or confirming the transaction as appropriate for the authentication flow being used). Each use of the authenticator **SHALL** require the presentation of the activation factor.

The use of an activation secret by the authenticator **SHALL** meet the requirements of Sec. 5.2.11. A biometric activation factor **SHALL** meet the requirements of Sec. 5.2.3, including limits on the number of consecutive authentication failures. Submission of the activation factor **SHALL** be a separate operation from unlocking of the host device (e.g., smartphone), although the same activation factor used to unlock the host device **MAY** be used in the authentication operation. The memorized secret or biometric sample used for activation — and any biometric data derived from the biometric sample such as a probe produced through signal processing — **SHALL** be zeroized immediately after the authentication operation.

5.1.4. Single-Factor OTP Device

A single-factor OTP device generates one-time passwords (OTPs). This category includes hardware devices and software-based OTP generators installed on devices such as mobile phones. These devices have an embedded secret that is used as the seed for generation of OTPs and does not require activation through a second factor. The OTP is displayed on the device and manually input for transmission to the verifier, thereby proving possession and control of the device. An OTP device may, for example, display 6 characters at a time. A single-factor OTP device is *something you have*.

Single-factor OTP devices are similar to look-up secret authenticators with the exception that the secrets are cryptographically and independently generated by the authenticator and verifier and compared by the verifier. The secret is computed based on a nonce that may be time-based or from a counter on the authenticator and verifier.

5.1.4.1. Single-Factor OTP Authenticators

Single-factor OTP authenticators contain two persistent values. The first is a symmetric key that persists for the device's lifetime. The second is a nonce that is either changed each time the authenticator is used or is based on a real-time clock.

The secret key and its algorithm **SHALL** provide at least the minimum security strength specified in the latest revision of [SP800-131A] (112 bits as of the date of this publication). The nonce **SHALL** be of sufficient length to ensure that it is unique for each operation of the device over its lifetime. If a subscriber needs to change the device used for a software-based OTP authenticator, they **SHOULD** bind the authenticator application on the new device to their subscriber account as described in Sec. 6.1.2.1 and invalidate the authenticator application that will no longer be used.

The authenticator output is obtained by using an approved block cipher or hash function to combine the key and nonce in a secure manner. The authenticator output **MAY** be truncated to as few as 6 decimal digits (approximately 20 bits of entropy).

If the nonce used to generate the authenticator output is based on a real-time clock, the nonce **SHALL** be changed at least once every 2 minutes.

5.1.4.2. Single-Factor OTP Verifiers

Single-factor OTP verifiers effectively duplicate the process of generating the OTP used by the authenticator. As such, the symmetric keys used by authenticators are also present in the verifier, and **SHALL** be strongly protected against unauthorized disclosure by the use of access controls that limit access to the keys to only those software components on the device requiring access.

When a single-factor OTP authenticator is being associated with a subscriber account, the verifier or associated CSP **SHALL** use approved cryptography to either generate and exchange or to obtain the secrets required to duplicate the authenticator output.

The verifier **SHALL** use approved encryption and an authenticated protected channel when collecting the OTP in order to provide resistance to eavesdropping and AitM attacks.

In order to provide replay resistance as described in Sec. 5.2.8, verifiers **SHALL** accept a given OTP only once while it is valid. In the event a claimant's authentication is denied due to duplicate use of an OTP, verifiers **MAY** warn the claimant in case an attacker has been able to authenticate in advance. Verifiers **MAY** also warn a subscriber in an existing session of the attempted duplicate use of an OTP.

Time-based OTPs [TOTP] **SHALL** have a defined lifetime that is determined by the expected clock drift — in either direction — of the authenticator over its lifetime, plus allowance for network delay and user entry of the OTP.

If the authenticator output has less than 64 bits of entropy, the verifier **SHALL** implement a rate-limiting mechanism that effectively limits the number of failed authentication attempts that can be made on the subscriber account as described in Sec. 5.2.2.

5.1.5. Multi-Factor OTP Devices

A multi-factor OTP device generates OTPs for use in authentication after activation through input of an activation factor. This includes hardware devices and software-based OTP generators installed on devices such as mobile phones. The second factor of authentication may be achieved through some kind of integral entry pad, an integral biometric (e.g., fingerprint) reader, or a direct computer interface (e.g., USB port). The OTP is displayed on the device and manually input for transmission to the verifier. For example, an OTP device may display 6 characters at a time, thereby proving possession and control of the device. The multi-factor OTP device is *something you have*, and it **SHALL** be activated by either *something you know* or *something you are*.

5.1.5.1. Multi-Factor OTP Authenticators

Multi-factor OTP authenticators operate in a similar manner to single-factor OTP authenticators (see Sec. 5.1.4.1), except that they require the presentation and verification of either a memorized secret or a biometric characteristic to obtain the OTP from the authenticator. Each use of the authenticator **SHALL** require the input of the activation factor.

In addition to activation information, multi-factor OTP authenticators contain two persistent values. The first is a symmetric key that persists for the device's lifetime. The second is a nonce that is either changed each time the authenticator is used or is based on a real-time clock.

The secret key and its algorithm **SHALL** provide at least the minimum security strength specified in the latest revision of [SP800-131A] (112 bits as of the date of this publication). The nonce **SHALL** be of sufficient length to ensure that it is unique for each operation of the device over its lifetime. If a subscriber needs to change the device used for a software-based OTP authenticator, they **SHOULD** bind the authenticator application on the new device to their subscriber account as described in Sec. 6.1.2.1 and invalidate the authenticator application that will no longer be used.

The authenticator output is obtained by using an approved block cipher or hash function to combine the key and nonce in a secure manner. The authenticator output **MAY** be truncated to as few as 6 decimal digits (approximately 20 bits of entropy).

If the nonce used to generate the authenticator output is based on a real-time clock, the nonce **SHALL** be changed at least once every 2 minutes.

The use of an activation secret by the authenticator **SHALL** meet the requirements of Sec. 5.2.11. A biometric activation factor **SHALL** meet the requirements of Sec. 5.2.3, including limits on the number of consecutive authentication failures. Submission of the activation factor **SHALL** be a separate operation from unlocking of the host device (e.g., smartphone), although the same activation factor used to unlock the host device **MAY** be used in the authentication operation. The unencrypted key and activation secret or biometric sample — and any biometric data derived from the biometric sample such as a probe produced through signal processing — **SHALL** be zeroized immediately after an OTP has been generated.

5.1.5.2. Multi-Factor OTP Verifiers

Multi-factor OTP verifiers effectively duplicate the process of generating the OTP used by the authenticator, but without the requirement that a second factor be provided. As such, the symmetric keys used by authenticators **SHALL** be strongly protected against unauthorized disclosure by the use of access controls that limit access to the keys to only those software components on the device requiring access.

When a multi-factor OTP authenticator is being associated with a subscriber account, the verifier or associated CSP **SHALL** use approved cryptography to either generate and exchange or to obtain the secrets required to duplicate the authenticator output. The verifier or CSP **SHALL** also establish, by issuance of the authentictor, that the authenticator is a multi-factor device. Otherwise, the verifier **SHALL** treat the authenticator as single-factor, in accordance with Sec. 5.1.4.

The verifier **SHALL** use approved encryption and an authenticated protected channel when collecting the OTP in order to provide resistance to eavesdropping and AitM attacks. In order to provide replay resistance as described in Sec. 5.2.8, verifiers **SHALL** accept a given OTP only once while it is valid. In the event a claimant's authentication is denied due to duplicate use of an OTP, verifiers **MAY** warn the claimant in case an attacker has been able to authenticate in advance. Verifiers **MAY** also warn a subscriber in an existing session of the attempted duplicate use of an OTP.

Time-based OTPs [TOTP] **SHALL** have a defined lifetime that is determined by the expected clock drift — in either direction — of the authenticator over its lifetime, plus allowance for network delay and user entry of the OTP.

If the authenticator output or activation secret has less than 64 bits of entropy, the verifier **SHALL** implement a rate-limiting mechanism that effectively limits the number of failed authentication attempts that can be made on the subscriber account as described in Sec. 5.2.2.

5.1.6. Single-Factor Cryptographic Software

A single-factor cryptographic software authenticator is a cryptographic key stored on disk or some other "soft" media. Authentication is accomplished by proving possession and control of the key. The authenticator output is highly dependent on the specific cryptographic protocol, but it is generally some type of signed message. The single-factor cryptographic software authenticator is *something you have*.

5.1.6.1. Single-Factor Cryptographic Software Authenticators

Single-factor cryptographic software authenticators encapsulate one or more secret keys unique to the authenticator. The key **SHALL** be stored in suitably secure storage available to the authenticator application (e.g., keychain storage, TPM, or TEE if available). The key **SHALL** be strongly protected against unauthorized disclosure by the use of access controls that limit access to the key to only those software components on the device requiring access.

External cryptographic authenticators that do not meet the requirements of cryptographic hardware authenticators (e.g., that have a mechanism to allow private keys to be exported) are also considered to be cryptographic software authenticators. They **SHALL** meet the requirements for connected authenticators in Sec. 5.2.12.

5.1.6.2. Single-Factor Cryptographic Software Verifiers

The requirements for a single-factor cryptographic software verifier are identical to those for a single-factor cryptographic device verifier, described in Sec. 5.1.7.2.

5.1.7. Single-Factor Cryptographic Devices

A single-factor cryptographic device is a hardware device that performs cryptographic operations using protected cryptographic keys and provides the authenticator output via direct connection to the user endpoint. The device uses embedded symmetric or asymmetric cryptographic keys, and does not require activation through a second factor of authentication. Authentication is accomplished by proving possession of the device via the authentication protocol. The authenticator output is provided by direct connection to the user endpoint and is highly dependent on the specific cryptographic device and protocol, but it is typically some type of signed message. A single-factor cryptographic device is *something you have*.

5.1.7.1. Single-Factor Cryptographic Device Authenticators

Single-factor cryptographic device authenticators use tamper-resistant hardware to encapsulate one or more secret keys unique to the authenticator that **SHALL NOT** be exportable (i.e., cannot be removed from the device). The authenticator operates using a secret key to sign a challenge nonce presented through a direct interface between the authenticator and endpoint (e.g., a USB port or secured wireless connection) as specified in Sec. 5.2.12. Alternatively, the authenticator could be a suitably secure processor integrated with the user endpoint itself.

The secret key and its algorithm **SHALL** provide at least the minimum security length specified in the latest revision of [SP800-131A] (112 bits as of the date of this publication). The challenge nonce **SHALL** be at least 64 bits in length. Approved cryptography **SHALL** be used.

Cryptographic device authenticators differ from cryptographic software authenticators because of the greater protection afforded to the embedded authentication secrets by cryptographic devices. In order to be considered a cryptographic device, an authenticator **SHALL** either be a separate piece of hardware or an embedded processor or execution environment, e.g., secure element, trusted execution environment (TEE), trusted platform module (TPM). These hardware authenticators or embedded processors are separate from a host processor such as the CPU on a laptop or mobile device. A cryptographic device authenticator **SHALL** be designed so as to prohibit the export of the authentication secret to the host processor and **SHALL NOT** be capable of being reprogrammed by the host processor so as to allow the secret to be extracted. The authenticator is subject to applicable [FIPS140] requirements of the AAL at which the authenticator is being used.

Single-factor cryptographic device authenticators **SHOULD** require a physical input (e.g., the pressing of a button) in order to operate. This provides defense against unintended operation of the device, which might occur if the endpoint to which it is connected is compromised.

5.1.7.2. Single-Factor Cryptographic Device Verifiers

Single-factor cryptographic device verifiers generate a challenge nonce, send it to the corresponding authenticator, and use the authenticator output to verify possession of the device. The authenticator output is highly dependent on the specific cryptographic device and protocol, but it is generally some type of signed message.

The verifier has either symmetric or asymmetric cryptographic keys corresponding to each authenticator. While both types of keys **SHALL** be protected against modification, symmetric keys **SHALL** additionally be protected against unauthorized disclosure by the use of access controls that limit access to the key to only those software components on the device requiring access.

The challenge nonce **SHALL** be at least 64 bits in length, and **SHALL** either be unique over the authenticator's lifetime or statistically unique (i.e., generated using an approved random bit generator [SP800-90Ar1]). The verification operation **SHALL** use approved cryptography.

5.1.8. Multi-Factor Cryptographic Software

A multi-factor cryptographic software authenticator is a cryptographic key stored on disk or some other "soft" media that requires activation through a second factor of authentication. Authentication is accomplished by proving possession and control of the key. The authenticator output is highly dependent on the specific cryptographic protocol, but it is generally some type of signed message. The multi-factor cryptographic software authenticator is *something you have*, and it **SHALL** be activated by either *something you know* or *something you are*.

5.1.8.1. Multi-Factor Cryptographic Software Authenticators

Multi-factor cryptographic software authenticators encapsulate one or more secret keys unique to the authenticator and accessible only through the presentation and verification of an activation factor, either a memorized secret or a biometric characteristic. The key **SHOULD** be stored in suitably secure storage available to the authenticator application (e.g., keychain storage, TPM, TEE). The key **SHALL** be strongly protected against unauthorized disclosure by the use of access controls that limit access to the key to only those software components on the device requiring access.

External cryptographic authenticators that do not meet the requirements of cryptographic hardware authenticators (e.g., that have a mechanism to allow private keys to be exported) are also considered to be cryptographic software authenticators. They **SHALL** meet the requirementss for connected authenticators in Sec. 5.2.12.

Each authentication operation using the authenticator **SHALL** require the input of the activation factor.

The use of an activation secret by the authenticator **SHALL** meet the requirements of Sec. 5.2.11. A biometric activation factor **SHALL** meet the requirements of Sec. 5.2.3, including limits on the number of consecutive authentication failures. Submission of the activation factor **SHALL** be a separate operation from unlocking of the host device (e.g., smartphone), although the same activation factor used to unlock the host device **MAY** be used in the authentication operation. The activation secret or biometric sample — and any biometric data derived from the biometric sample such as a probe produced through signal processing — **SHALL** be zeroized immediately after an authentication transaction has taken place.

5.1.8.2. Multi-Factor Cryptographic Software Verifiers

The requirements for a multi-factor cryptographic software verifier are identical to those for a single-factor cryptographic device verifier, described in Sec. 5.1.7.2. Verification of the output from a multi-factor cryptographic software authenticator proves use of the activation factor.

5.1.9. Multi-Factor Cryptographic Devices

A multi-factor cryptographic device is a hardware device that performs cryptographic operations using one or more protected cryptographic keys and requires activation through a second authentication factor. Authentication is accomplished by proving possession of the device and control of the key. The authenticator output is provided by direct connection to the user endpoint and is highly dependent on the specific cryptographic device and protocol, but it is typically some type of signed message. The multi-factor cryptographic device is *something you have*, and it **SHALL** be activated by either *something you know* or *something you are*.

5.1.9.1. Multi-Factor Cryptographic Device Authenticators

Multi-factor cryptographic device authenticators use tamper-resistant hardware to encapsulate one or more secret keys unique to the authenticator that **SHALL NOT** be exportable (i.e., cannot be removed from the device). The secret key **SHALL** be accessible only through the presentation and verification of an activation factor, either a biometric characteristic or an activation secret as described in Sec. 5.2.11. The authenticator operates by using a secret key that was unlocked by the activation factor to sign a challenge nonce presented through a direct interface between the authenticator and endpoint (e.g., a USB port or secured wireless connection) as specified in Sec. 5.2.12. Alternatively, the authenticator could be a suitably secure processor integrated with the user endpoint itself (e.g., a hardware TPM).

The secret key and its algorithm **SHALL** provide at least the minimum security length specified in the latest revision of [SP800-131A] (112 bits as of the date of this publication). The challenge nonce **SHALL** be at least 64 bits in length. Approved cryptography **SHALL** be used.

Cryptographic device authenticators differ from cryptographic software authenticators because of the greater protection afforded to the embedded authentication secrets by cryptographic devices. In order to be considered a cryptographic device, an authenticator **SHALL** either be a separate piece of hardware or an embedded processor or execution environment, e.g., secure element, trusted execution environment (TEE), trusted platform module (TPM). A cryptographic device authenticator **SHALL** be designed so as to prohibit the export of the authentication secret to the host processor and **SHALL NOT** be capable of being reprogrammed by the host processor so as to allow the secret to be extracted. The authenticator is subject to applicable [FIPS140] requirements of the AAL at which the authenticator is being used.

Each authentication operation using the authenticator **SHOULD** require the input of the activation factor. Input of the activation factor **MAY** be accomplished via either direct input on the device or via a hardware connection (e.g., USB, smartcard).

The use of an activation secret by the authenticator **SHALL** meet the requirements of Sec. 5.2.11. A biometric activation factor **SHALL** meet the requirements of Sec. 5.2.3, including limits on the number of consecutive authentication failures. Submission of the activation factor **SHALL** be a separate operation from unlocking of the host device (e.g., smartphone), although the same activation factor used to unlock the host device **MAY** be used in the authentication operation. The activation secret or biometric sample — and any biometric data derived from the biometric sample such as a probe produced through signal processing — **SHALL** be zeroized immediately after an authentication transaction has taken place.

5.1.9.2. Multi-Factor Cryptographic Device Verifiers

The requirements for a multi-factor cryptographic device verifier are identical to those for a single-factor cryptographic device verifier, described in Sec. 5.1.7.2. Verification of the authenticator output from a multi-factor cryptographic device proves use of the activation factor.

5.2. General Authenticator Requirements

5.2.1. Physical Authenticators

CSPs **SHALL** provide subscriber instructions on how to appropriately protect the authenticator against theft or loss. The CSP **SHALL** provide a mechanism to invalidate the authenticator immediately upon notification from subscriber that loss or theft of the authenticator is suspected.

5.2.2. Rate Limiting (Throttling)

When required by the authenticator type descriptions in Sec. 5.1, the verifier **SHALL** implement controls to protect against online guessing attacks. Unless otherwise specified in the description of a given authenticator, the verifier **SHALL** limit consecutive failed authentication attempts on a single subscriber account to no more than 100.

Additional techniques **MAY** be used to reduce the likelihood that an attacker will lock the legitimate claimant out as a result of rate limiting. These include:

- Requiring the claimant to complete a bot-detection and mitigation challenge before attempting authentication.
- Requiring the claimant to wait following a failed attempt for a period of time that increases as the subscriber account approaches its maximum allowance for consecutive failed attempts (e.g., 30 seconds up to an hour).

- Accepting only authentication requests that come from an allowlist of IP addresses from which the subscriber has been successfully authenticated before.
- Leveraging other risk-based or adaptive authentication techniques to identify user behavior that falls within, or out of, typical norms. These might, for example, include use of IP address, geolocation, timing of request patterns, or browser metadata.

When the subscriber successfully authenticates, the verifier **SHOULD** disregard any previous failed attempts for that user from the same IP address.

5.2.3. Use of Biometrics

The use of biometrics (*something you are*) in authentication includes both measurement of physical characteristics (e.g., fingerprint, iris, facial characteristics) and behavioral characteristics (e.g., typing cadence). Both classes are considered biometric modalities, although different modalities may differ in the extent to which they establish authentication intent as described in Sec. 5.2.9.

For a variety of reasons, this document supports only limited use of biometrics for authentication. These reasons include:

- The biometric False Match Rate (FMR) does not provide confidence in the authentication of the subscriber by itself. In addition, FMR does not account for spoofing attacks.
- Biometric comparison is probabilistic, whereas the other authentication factors are deterministic.
- Biometric template protection schemes provide a method for revoking biometric credentials that is comparable to other authentication factors (e.g., PKI certificates and passwords). However, the availability of such solutions is limited, and standards for testing these methods are under development.
- Biometric characteristics do not constitute secrets. They can often be obtained online or, in the case of a facial image, by taking a picture of someone with or without their knowledge. Latent fingerprints can be lifted from objects someone touches, and iris patterns can be captured with high resolution images. While presentation attack detection (PAD) technologies can mitigate the risk of these types of attacks, additional trust in the sensor or biometric processing is required to ensure that PAD is operating in accordance with the needs of the CSP and the subscriber.

Therefore, the limited use of biometrics for authentication is supported with the following requirements and guidelines:

Biometrics **SHALL** be used only as part of multi-factor authentication with a physical authenticator (*something you have*).

The biometric system **SHALL** operate with a false-match rate (FMR) [ISO/IEC2382-37] of 1 in 10000 or better. This FMR **SHALL** be achieved under conditions of a conformant attack (i.e., zero-effort impostor attempt) as defined in [ISO/IEC30107-1].

The biometric system **SHOULD** implement presentation attack detection (PAD). Testing of the biometric system to be deployed **SHOULD** demonstrate at least 90% resistance to presentation attacks for each relevant attack type (i.e., species), where resistance is defined as the number of thwarted presentation attacks divided by the number of trial presentation attacks. Testing of presentation attack resistance **SHALL** be in accordance with Clause 12 of [ISO/IEC30107-3]. The PAD decision **MAY** be made either locally on the claimant's device or by a central verifier.

The biometric system **SHALL** allow no more than 5 consecutive failed authentication attempts or 10 consecutive failed attempts if PAD, meeting the above requirements, is implemented. Once that limit has been reached, the biometric authenticator **SHALL** impose a delay of at least 30 seconds before each subsequent attempt, with an overall limit of no more than 50 consecutive failed authentication attempts (100 if PAD is implemented). Once the overall limit is reached, the biometric system **SHALL** disable biometric user authentication and offer another factor (e.g., a different biometric modality or an activation secret if it is not already a required factor) if such an alternative method is already available.

The verifier **SHALL** make a determination of sensor and endpoint performance, integrity, and authenticity. Acceptable methods for making this determination include, but are not limited to:

- Authentication of the sensor or endpoint
- Certification by an approved accreditation authority
- Runtime interrogation of signed metadata (e.g., attestation) as described in Sec. 5.2.4.

Biometric comparison can be performed locally on the claimant's device or at a central verifier. Since the potential for attacks on a larger scale is greater at central verifiers, comparison **SHOULD** be performed locally.

If comparison is performed centrally:

- Use of the biometric as an authentication factor **SHALL** be limited to one or more specific devices that are identified using approved cryptography. Since the biometric has not yet unlocked the main authentication key, a separate key **SHALL** be used for identifying the device.
- Biometric revocation, referred to as biometric template protection in [ISO/IEC24745], **SHALL** be implemented.

- An authenticated protected channel between sensor (or an endpoint containing a sensor that resists sensor replacement) and verifier **SHALL** be established and the sensor or endpoint **SHALL** be authenticated prior to capturing the biometric sample from the claimant.
- All transmission of biometrics **SHALL** be over an authenticated protected channel.

Biometric samples collected in the authentication process **MAY** be used to train comparison algorithms or — with user consent — for other research purposes. Biometric samples and any biometric data derived from the biometric sample such as a probe produced through signal processing **SHALL** be zeroized immediately after any training or research data has been derived.

Biometric authentication technologies **SHALL** provide similar performance for subscribers of different demographic types (racial background, gender, ethnicity, etc.).

5.2.4. Attestation

An attestation is information conveyed to the verifier regarding a connected authenticator or the endpoint involved in an authentication operation. Information conveyed by attestation **MAY** include, but is not limited to:

- The provenance (e.g., manufacturer or supplier certification), health, and integrity of the authenticator and endpoint
- Security features of the authenticator
- Security and performance characteristics of biometric sensors
- Sensor modality

If this attestation is signed, it **SHALL** be signed using a digital signature that provides at least the minimum security strength specified in the latest revision of [SP800-131A] (112 bits as of the date of this publication).

Attestation information **MAY** be used as part of a verifier's risk-based authentication decision.

5.2.5. Phishing (Verifier Impersonation) Resistance

Phishing attacks, previously referred to in SP 800-63B as "verifier impersonation," are attempts by fraudulent verifiers and RPs to fool an unwary claimant into presenting an authenticator to an impostor. In some prior versions of SP 800-63, protocols resistant to phishing attacks were also referred to as "strongly MitM resistant."

The term *phishing* is widely used to describe a variety of similar attacks. For the purposes of this document, phishing resistance is the ability of the authentication protocol to detect and prevent disclosure of authentication secrets and valid authenticator outputs to an impostor relying party without reliance on the vigilance of the subscriber. The means by which the subscriber was directed to the impostor relying party are not relevant. For example, regardless of whether the subscriber was directed there via search engine optimization or prompted by email, it is considered to be a phishing attack.

Approved cryptographic algorithms **SHALL** be used to establish phishing resistance where it is required. Keys used for this purpose **SHALL** provide at least the minimum security strength specified in the latest revision of [SP800-131A] (112 bits as of the date of this publication).

Authenticators that involve the manual entry of an authenticator output, such as out-of-band and OTP authenticators, **SHALL NOT** be considered phishing resistant because the manual entry does not bind the authenticator output to the specific session being authenticated. In an AitM attack, an impostor verifier could replay the OTP authenticator output to the verifier and successfully authenticate.

While an individual authenticator may be phishing resistant, phishing resistance for a given subscriber account is only achieved when all methods of authentication are phishing resistant.

Two methods of phishing resistance are recognized: channel binding and verifier name binding. Channel binding is considered more secure than verifier name binding because it is not vulnerable to mis-issuance or misappropriation of relying party certificates, but either method satisfies the requirements for phishing resistance.

5.2.5.1. Channel Binding

An authentication protocol with channel binding **SHALL** establish an authenticated protected channel with the verifier. It **SHALL** then strongly and irreversibly bind a channel identifier that was negotiated in establishing the authenticated protected channel to the authenticator output (e.g., by signing the two values together using a private key controlled by the claimant for which the public key is known to the verifier). The verifier **SHALL** validate the signature or other information used to prove phishing resistance. This prevents an impostor verifier, even one that has obtained a certificate representing the actual verifier, from successfully relaying that authentication on a different authenticated protected channel.

An example of a phishing resistant authentication protocol that uses channel binding is client-authenticated TLS, because the client signs the authenticator output along with earlier messages from the protocol that are unique to the particular TLS connection being negotiated.

5.2.5.2. Verifier Name Binding

An authentication protocol with authenticator name binding **SHALL** establish an authenticated protected channel with the verifier. It **SHALL** then generate an authenticator output that is cryptographically bound to a verifier identifier that is authenticated as part of the protocol. In the case of domain name system (DNS) identifiers, the verifier identifier **SHALL** be either the authenticated hostname of the verifier or a parent domain that is at least one level below the public suffix [PSL] associated with that hostname. The binding **MAY** be established by choosing an associated authenticator secret, by deriving an authenticator secret using the verifier identifier, by cryptographically signing the authenticator output with the verifier identifier, or similar cryptographically secure means.

5.2.6. Verifier-CSP Communications

In situations where the verifier and CSP are separate entities (as shown by the dotted line in [SP800-63] Figure 1), communications between the verifier and CSP **SHALL** occur through a mutually authenticated secure channel (such as a client-authenticated TLS connection) using approved cryptography.

5.2.7. Verifier Compromise Resistance

Use of some types of authenticators requires that the verifier store a copy of the authenticator secret. For example, an OTP authenticator (described in Sec. 5.1.4) requires that the verifier independently generate the authenticator output for comparison against the value sent by the claimant. Because of the potential for the verifier to be compromised and stored secrets stolen, authentication protocols that do not require the verifier to persistently store secrets that could be used for authentication are considered stronger, and are described herein as being *verifier compromise resistant*. Note that such verifiers are not resistant to all attacks. A verifier could be compromised in a different way, such as being manipulated into always accepting a particular authenticator output.

Verifier compromise resistance can be achieved in different ways, for example:

- Use a cryptographic authenticator that requires the verifier store a public key corresponding to a private key held by the authenticator.
- Store the expected authenticator output in hashed form. This method can be used with some look-up secret authenticators (described in Sec. 5.1.2), for example.

To be considered verifier compromise resistant, public keys stored by the verifier **SHALL** be associated with the use of approved cryptographic algorithms and **SHALL** provide at least the minimum security strength specified in the latest revision of [SP800-131A] (112 bits as of the date of this publication).

Other verifier compromise resistant secrets **SHALL** use approved hash algorithms and the underlying secrets **SHALL** have at least the minimum security strength specified in the latest revision of [SP800-131A] (112 bits as of the date of this publication). Secrets (e.g., memorized secrets) having lower complexity **SHALL NOT** be considered verifier compromise resistant when hashed because of the potential to defeat the hashing process through dictionary lookup or exhaustive search.

5.2.8. 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. Replay resistance is in addition to the replay-resistant nature of authenticated protected channel protocols, since the output could be stolen prior to entry into the protected channel. Protocols that use nonces or challenges to prove the "freshness" of the transaction are resistant to replay attacks since the verifier will easily detect when old protocol messages are replayed since they will not contain the appropriate nonces or timeliness data.

Examples of replay-resistant authenticators are OTP devices, cryptographic authenticators, and look-up secrets.

In contrast, memorized secrets are not considered replay resistant because the authenticator output — the secret itself — is provided for each authentication.

5.2.9. Authentication Intent

An authentication process demonstrates intent if it requires the subject to explicitly respond to each authentication or reauthentication request. The goal of authentication intent is to make it more difficult for authenticators (e.g., multi-factor cryptographic devices) to be used without the subject's knowledge, such as by malware on the endpoint. Authentication intent **SHALL** be established by the authenticator itself, although multi-factor cryptographic devices **MAY** establish intent by reentry of the activation factor for the authenticator.

Authentication intent **MAY** be established in a number of ways. Authentication processes that require the subject's intervention establish intent (e.g., a claimant entering an authenticator output from an OTP device). Cryptographic devices that require user action for each authentication or reauthentication operation also establish intent (e.g., pushing a button or reinsertion).

Depending on the modality, presentation of a biometric characteristic may or may not establish authentication intent. Behavioral biometrics similarly may or may not establish authentication intent because they do not always require a specific action on the claimant's part.

5.2.10. Restricted Authenticators

As threats evolve, authenticators' capability to resist attacks typically degrades. Conversely, some authenticators' performance may improve, for example, when changes to their underlying standards increases their ability to resist particular attacks.

To account for these changes in authenticator performance, NIST places additional restrictions on authenticator types or specific classes or instantiations of an authenticator type.

The use of a *restricted authenticator* requires that the implementing organization assess, understand, and accept the risks associated with that authenticator and acknowledge that risk will likely increase over time. It is the responsibility of the organization to determine the level of acceptable risk for their systems and associated data and to define any methods for mitigating excessive risks. If at any time the organization determines that the risk to any party is unacceptable, then that authenticator **SHALL NOT** be used.

Further, the risk of an authentication error is typically borne by multiple parties, including the implementing organization, organizations that rely on the authentication decision, and the subscriber. Because the subscriber may be exposed to additional risk when an organization accepts a restricted authenticator and that the subscriber may have a limited understanding of and ability to control that risk, the CSP **SHALL**:

1. Offer subscribers at least one alternate authenticator that is not restricted and can be used to authenticate at the required AAL.
2. Provide meaningful notice to subscribers regarding the security risks of the restricted authenticator and availability of alternatives that are not restricted.
3. Address any additional risk to subscribers in its risk assessment.
4. Develop a migration plan for the possibility that the restricted authenticator is no longer acceptable at some point in the future and include this migration plan in its digital identity acceptance statement.

5.2.11. Activation Secrets

Memorized secrets that are used as an activation factor for a multi-factor authenticator are referred to as *activation secrets*. An activation secret is used to decrypt a stored secret key used for authentication or is compared against a locally held stored verifier to provide access to the authentication key. In either of these cases, the activation secret **SHALL** remain within the authenticator and its associated user endpoint.

Authenticators making use of activation secrets **SHALL** require the secrets to be at least 6 characters in length. Activation secrets **MAY** be entirely numeric (i.e., a PIN). If alphanumeric (rather than only numeric) values are permitted, all printing ASCII [RFC20] characters as well as the space character **SHOULD** be accepted. Unicode [ISO/IEC 10646] characters **SHOULD** be accepted as well in alphanumeric secrets. The authenticator **SHALL** contain a blocklist (either specified by specific values or by an algorithm) of at least 10 commonly used activation values and **SHALL** prevent their use as activation secrets.

The authenticator or verifier **SHALL** implement a retry-limiting mechanism that effectively limits the number of consecutive failed activation attempts using the authenticator to ten (10). If the entry of an incorrect activation secret causes the authenticator to generate an invalid output that is sent to the central verifier, rate limiting **MAY** be implemented by the verifier. In all other cases, rate limiting **SHALL** be implemented in the authenticator. Once the limit of 10 attempts is reached, the authenticator **SHALL** be disabled and a different authenticator **SHALL** be required for authentication.

If the authenticator verifies the activation secret locally (rather than using it for decryption of a key), verification **SHALL** be performed within a hardware-based authenticator or in a secure element (e.g., TEE, TPM) that releases the authentication secret only upon presentation of the correct activation secret. In other circumstances (i.e., software-based multi-factor authenticators), the authenticator **SHALL** use the memorized secret as a key to decrypt its stored authentication secret. Approved cryptography **SHALL** be used.

5.2.12. Connected Authenticators

Cryptographic authenticators require a direct connection between the authenticator and the endpoint being authenticated. This connection **MAY** be wired (e.g., USB or direct connection with a smartcard) or wireless (e.g., NFC, Bluetooth). While in most cases wired connections can be presumed to be secure from eavesdropping and adversary-in-the-middle attacks, additional precautions are required for authenticators that are connected via wireless technologies.

Wired authenticator connections include both authenticators that are embedded in endpoints (e.g., in a TPM) and those that are connected via an external interface, such as USB. Claimants **SHOULD** be advised to use trusted hardware (cables, etc.) for external connections for additional assurance that they have not been compromised.

Wireless authenticator connections are potentially vulnerable to threats including eavesdropping, injection, and relay attacks. The potential for such attacks depends on the effective range of the wireless technology being used.

Wireless technologies having an effective range of 1 meter or more (e.g., Bluetooth LE) **SHALL** use an authenticated encrypted connection between the authenticator and endpoint. A pairing process **SHALL** be used to establish a key for encrypted communication between the authenticator and endpoint. A temporary wired connection between the devices **MAY** also be used to establish the key in lieu of the pairing process. The pairing process **SHALL** be authenticated through the use of a pairing code. The pairing code **SHALL** be associated with either the authenticator or endpoint and **SHALL** have at least 20 bits or 6 decimal digits of entropy. The pairing code **MAY** be printed on the associated device and **SHALL** be conveyed between the devices by manual entry or by using a QR code or similar representation that is optically communicated. An example of this is the pairing code used with the virtual contact interface specified in [SP800-73]. The entire authentication transaction **SHALL** be encrypted using a key established by the pairing process.

When a wireless technology with an effective range of less than 1 meter is in use (e.g., NFC), the activation secret, if any, transmitted from the endpoint to authenticator **SHALL** be encrypted using a key established through a pairing process between the devices or through a temporary wired connection. An authenticated connection using a pairing code meeting the above requirements **SHOULD** be used. If the authenticator is configured to require authenticated pairing, pairing code **SHALL** be used.

> Note: Encryption of only the activation secret, and not the entire authentication transaction, may expose sensitive information such as the identity of the relying party, although this would require the attacker to be very close to the subscriber. Special care should be taken with authenticators containing personally identifiable information that do not require authenticated pairing to protect that information against "skimming" and eavesdropping attacks.

The key established as a result of the pairing process **MAY** be either temporary (valid for a limited number of transactions or time) or persistent. A mechanism for endpoints to remove persistent keys **SHALL** be provided.

Where cryptographic operations are required, approved cryptography **SHALL** be used. All communication of authentication data between authenticators and endpoints **SHALL** occur directly between those devices or through an authenticated protected channel between the authenticator and endpoint.

6. Authenticator Lifecycle Management

This section is normative.

A number of events can occur over the lifecycle of a subscriber's authenticator that affect that authenticator's use. These events include binding, loss, theft, unauthorized duplication, expiration, and revocation. This section describes the actions to be taken in response to those events.

6.1. Authenticator Binding

Authenticator binding refers to the establishment of an association between a specific authenticator and a subscriber account, enabling the authenticator to be used — possibly in conjunction with other authenticators — to authenticate for that subscriber account.

Authenticators **SHALL** be bound to subscriber accounts either

- by issuance by the CSP as part of enrollment or
- by registration of a subscriber-provided authenticator that is acceptable to the CSP.

These guidelines refer to the *binding* rather than the issuance of an authenticator to accommodate both options.

Throughout the digital identity lifecycle, CSPs **SHALL** maintain a record of all authenticators that are or have been associated with each subscriber account. The CSP or verifier **SHALL** maintain the information required for throttling authentication attempts when required, as described in Sec. 5.2.2. The CSP **SHALL** also verify the type of user-provided authenticator (e.g., single-factor cryptographic device vs. multi-factor cryptographic device) so verifiers can determine compliance with requirements at each AAL.

The record created by the CSP **SHALL** contain the date and time the authenticator was bound to the subscriber account. The record **SHOULD** include information about the source of the binding (e.g., IP address, device identifier) of any device associated with the enrollment. If available, the record **SHOULD** also contain information about the source of unsuccessful authentications attempted with the authenticator.

When any new authenticator is bound to a subscriber account, the CSP **SHALL** ensure that the binding protocol and the protocol for provisioning the associated keys are done at a level of security commensurate with the AAL at which the authenticator will be used. For example, protocols for key provisioning **SHALL** use authenticated protected channels or be performed in person to protect against adversary-in-the-middle attacks. Binding of multi-factor authenticators **SHALL** require multi-factor authentication or equivalent (e.g., association with the session in which identity proofing has been just completed) be used in order to bind the authenticator. The same conditions apply when a key pair is generated by the authenticator and the public key is sent to the CSP.

As part of the binding process, the CSP **MAY** require additional information about the new authenticator or the endpoint it is associated with to determine that they are suitable for the AAL being requested and to attempt to determine that the endpoint and authenticator are free from malware.

6.1.1. Binding at Enrollment

The following requirements apply when an authenticator is bound to a subscriber account as part of the enrollment process.

The CSP **SHALL** bind at least one — and **SHOULD** bind at least two — physical (*something you have*) authenticators to the subscriber account, in addition to a memorized secret or one or more biometric characteristics. Binding of multiple authenticators provides a means to recover from the loss or theft of the subscriber's primary authenticator. Preservation of online material or an online reputation makes it undesirable to lose control of a subscriber account due to the loss of an authenticator. The second authenticator makes it possible to securely recover from an authenticator loss.

If enrollment and binding cannot be completed in a single physical encounter or electronic transaction (i.e., within a single protected session), the following methods **SHALL** be used to ensure that the same party acts as the applicant throughout the processes:

For remote transactions:

1. The applicant **SHALL** identify themselves in each new binding transaction by presenting a temporary secret which was either established during a prior transaction, or sent to the applicant's phone number, email address, or postal address of record.
2. Long-term authenticator secrets **SHALL** only be issued to the applicant within a protected session.

For in-person transactions:

1. The applicant **SHALL** identify themselves in person by either using a secret as described in remote transaction (1) above, or through use of a biometric that was recorded during a prior encounter.
2. Temporary secrets **SHALL NOT** be reused.
3. If the CSP issues long-term authenticator 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.

6.1.2. Post-Enrollment Binding

6.1.2.1. Binding of an Additional Authenticator at Existing AAL

With the exception of memorized secrets, CSPs and verifiers **SHOULD** encourage subscribers to maintain at least two valid authenticators of each factor that they will be using. For example, a subscriber who usually uses an OTP device as a physical authenticator **MAY** also be issued a number of look-up secret authenticators, or register a device for out-of-band authentication, in case the physical authenticator is lost, stolen, or damaged. See Sec. 6.1.2.3 for more information on replacement of memorized secret authenticators.

Accordingly, CSPs **SHOULD** permit the binding of additional authenticators to a subscriber account. Before adding the new authenticator, the CSP **SHALL** first require the subscriber to authenticate at the AAL (or a higher AAL) at which the new authenticator will be used. A separate authentication using existing authenticators **SHALL** be performed following the request to bind a new authenticator, and **SHALL** be valid for 20 minutes. When an authenticator is added, the CSP **SHOULD** send a notification to the subscriber via a mechanism that is independent of the transaction binding the new authenticator (e.g., email to an address previously associated with the subscriber). The CSP **MAY** limit the number of authenticators that are bound in this manner.

6.1.2.2. Adding an Additional Factor to a Single-Factor Subscriber Account

If the subscriber account has only one authentication factor bound to it and an additional authenticator of a different authentication factor is to be added, the subscriber **MAY** request that the subscriber account be upgraded to AAL2.

Before binding the new authenticator, the CSP **SHALL** require the subscriber to authenticate at AAL1. The CSP **SHOULD** send a notification of the event to the subscriber via a mechanism independent of the transaction binding the new authenticator (e.g., email to an address previously associated with the subscriber).

6.1.2.3. Account Recovery

The situation where a subscriber loses control of authenticators necessary to successfully authenticate is commonly referred to as *account recovery*.

If a subscriber that has been identity proofed loses all authenticators necessary to complete authentication, that subscriber **SHALL** repeat the identity proofing process described in [SP800-63A]. If the CSP has retained information from the evidence used in the original identity proofing process (pursuant to a privacy risk assessment as described in [SP800-63A] Sec. 5.2.2) that is sufficient to perform verification of the subscriber and if that evidence is still valid, it **MAY** repeat only the verification portion of the identity proofing process as described in [SP800-63A].

The CSP **SHALL** require the claimant to authenticate using an authenticator of the remaining factor, if any, to confirm binding to the existing subscriber account. Reestablishment of authentication factors at IAL3 **SHALL** be done in person or through a supervised remote process as described in [SP800-63A] Sec. 5.6.8, and **SHALL** perform a successful biometric comparison against the biometric characteristic collected during the original identity proofing process.

The CSP **SHOULD** send a notification of the event to the subscriber. This **MAY** be the same notice that is required as part of the identity proofing process.

Subscriber accounts that have not been identity proofed (i.e., without IAL) cannot be recovered because there is no reliable means for reassociating the subscriber with that account. Such accounts **SHALL** be treated as abandoned and a new subscriber account **SHALL** be established.

Replacement of a lost (i.e., forgotten) memorized secret is problematic because it is very common. Additional “backup” memorized secrets do not mitigate this because they are just as likely to also have been forgotten. If a biometric is bound to the subscriber account, the biometric characteristic and associated physical authenticator **SHOULD** be used to establish a new memorized secret.

As an alternative to the above re-proofing process when there is no biometric bound to the subscriber account, the CSP **MAY** bind a new memorized secret with authentication using two physical authenticators, along with a confirmation code that has been sent to one of the subscriber’s addresses of record. The confirmation code **SHALL** consist of at least 6 random alphanumeric characters generated by an approved random bit generator [SP800-90Ar1]. Confirmation codes **SHALL** be valid for at most:

- 21 days, when sent to a postal address of record within the contiguous United States;
- 30 days, when sent to a postal address of record outside the contiguous United States;
- 10 minutes, when sent to a telephone of record (SMS or voice); or
- 24 hours, when sent to an email address of record.

6.1.2.4. External Authenticator Binding

External authenticator binding refers to the process of binding an authenticator to a subscriber account when it is not connected to (or embedded in) the authenticated endpoint. This process is typically used when adding authenticators that are embedded in a new endpoint, or when connectivity limitations prevent the newly bound authenticator from being connected to an authenticated endpoint.

The binding process **MAY** begin with a request from an endpoint that has authenticated to the CSP obtaining a binding code from the CSP that is input into the endpoint associated with the new authenticator and sent to that CSP. Alternatively, the endpoint associated with the new authenticator **MAY** obtain a binding code from the CSP, which is input to an authenticated endpoint and sent to the CSP.

In addition to the requirements given in Sec. 6.1.2.1, Sec. 6.1.2.2, and Sec. 6.1.2.3 above as applicable, the following requirements **SHALL** apply when binding an external authenticator:

- An authenticated protected session **SHALL** be established by the endpoint associated with the new authenticator and the CSP.
- The subscriber **MAY** be prompted to enter an identifier by which they are known by the CSP on the endpoint associated with the new authenticator.
- The CSP **SHALL** generate a *binding code* using an approved random number generator and send it to either the new authenticator endpoint or the authenticated endpoint approving the binding. The binding code **SHALL** have at least 40 bits of entropy if used in conjunction with an identifier entered on the previous step; otherwise a binding code with at least 112 bits of entropy **SHALL** be required.
- The subscriber **SHALL** transfer the binding code to the other endpoint. This transfer **SHALL** be either manual or via a local out-of-band method such as a QR code. The binding code **SHALL NOT** be communicated over any insecure channel such as email or PSTN (SMS or voice).
- The binding code **SHALL** be usable only once and **SHALL** be valid for a maximum of 10 minutes.
- Following the binding of the new authenticator (or issuance of a certificate, in the case of PKI-based authenticators), the CSP **SHOULD** encourage the subscriber to authenticate with the new authenticator to confirm that the process has completed successfully.
- The CSP **SHALL** provide clear instruction on what the subscriber should do in the event of an authenticator binding mishap, such as a button or contact address to allow a mis-bound authenticator to be quickly invalidated as appropriate. This **MAY** be provided in the authenticated session or in the binding notification described in Sec. 6.1.2.1, Sec. 6.1.2.2, and Sec. 6.1.2.3 above.

Binding an external authenticator is a potentially risky operation because of the potential for the subscriber to be tricked into using a binding code by an attacker or supplying a binding code to an attacker. In some cases, QR codes obtained from a trusted source (such as from an authenticated session, especially when that authentication is phishing resistant) are considered to be more robust against such attacks, because they typically contain the URL of the CSP as well as the binding code. There is less potential for the subscriber to be fooled into entering a binding code at a phishing site as a result.

6.1.3. Binding to a Subscriber-provided Authenticator

A subscriber may already possess authenticators suitable for authentication at a particular AAL. For example, they may have a two-factor authenticator from a social network provider, considered AAL2 and IAL1, and would like to use those credentials at an RP that requires IAL2.

CSPs **SHOULD**, where practical, accommodate the use of subscriber-provided authenticators in order to relieve the burden to the subscriber of managing a large number of authenticators. Binding of these authenticators **SHALL** be done as described in Sec. 6.1.2. In situations where the authenticator strength is not self-evident (e.g., between single-factor and multi-factor authenticators of a given type), the CSP **SHALL** assume the use of the weaker authenticator unless it is able to establish that the stronger authenticator is in fact being used (e.g., by verification with the issuer or manufacturer of the authenticator).

6.1.4. Renewal

The subscriber **SHOULD** bind a new or updated authenticator an appropriate amount of time before an existing authenticator's expiration. The process for this **SHOULD** conform closely to the binding process for an additional authenticator described in Sec. 6.1.2.1. The CSP **MAY** periodically take other actions, such as reconfirming address of record, either as a part of the renewal process or separately. Following successful use of the replacement authenticator, the CSP **MAY** invalidate the authenticator that is expiring.

6.2. Loss, Theft, Damage, and Unauthorized Duplication

Compromised authenticators include those that have been lost, stolen, or subject to unauthorized duplication. Generally, one must assume that a lost authenticator has been stolen or compromised by someone that is not the legitimate subscriber of the authenticator. Damaged or malfunctioning authenticators are also considered compromised to guard against any possibility of extraction of the authenticator secret. One notable exception is a memorized secret that has been forgotten without other indications of having been compromised, such as having been obtained by an attacker.

Suspension, revocation, or destruction of compromised authenticators **SHOULD** occur as promptly as practical following detection. Organizations **SHOULD** establish time limits for this process.

To facilitate secure reporting of the loss, theft, or damage to an authenticator, the CSP **SHOULD** provide the subscriber with a method of authenticating to the CSP using a backup or alternate authenticator. This backup authenticator **SHALL** be either a memorized secret or a physical authenticator. Either could be used, but only one authentication factor is required to make this report. Alternatively, the subscriber **MAY** establish an authenticated protected channel to the CSP and verify information collected during the proofing process. The CSP **MAY** choose to verify an address of record (i.e., email, telephone, postal) and suspend authenticators reported to have been compromised. The suspension **SHALL** be reversible if the subscriber successfully authenticates to the CSP using a valid (i.e., not suspended) authenticator and requests reactivation of an authenticator suspended in this manner. The CSP **MAY** set a time limit after which a suspended authenticator can no longer be reactivated.

6.3. Expiration

CSPs **MAY** issue authenticators that expire. If and when an authenticator expires, it **SHALL NOT** be usable for authentication. When an authentication is attempted using an expired authenticator, the CSP **SHOULD** give an indication to the subscriber that the authentication failure is due to expiration rather than some other cause.

The CSP **SHALL** require subscribers to surrender or prove destruction of any physical authenticator containing attribute certificates signed by the CSP as soon as practical after expiration or receipt of a renewed authenticator.

6.4. Invalidation

Invalidation of an authenticator (sometimes referred to as revocation or termination) refers to removal of the binding between an authenticator and a subscriber account.

CSPs **SHALL** invalidate authenticators promptly when a subscriber account ceases to exist (e.g., subscriber's death, discovery of a fraudulent subscriber), when requested by the subscriber, or when the CSP determines that the subscriber no longer meets its eligibility requirements.

The CSP **SHALL** require subscribers to surrender or certify destruction of any physical authenticator containing subscriber attributes, such as certificates signed by the CSP, as soon as practical after invalidation takes place. This is necessary to protect the privacy of the subscriber and to block the use of any certificates in offline situations between invalidation and expiration of the certificates.

Further requirements on the invalidation of PIV authenticators are found in [FIPS201].

7. Session Management

This section is normative.

Once an authentication event has taken place, it is often desirable to allow the subscriber to continue using the application across multiple subsequent interactions without requiring them to repeat the authentication event. This requirement is particularly true for federation scenarios — described in [SP800-63C] — where the authentication event necessarily involves several components and parties coordinating across a network.

To facilitate this behavior, a *session* **MAY** be started in response to an authentication event, and continue the session until such time that it is terminated. The session **MAY** be terminated for any number of reasons, including but not limited to an inactivity timeout, an explicit logout event, or other means. The session **MAY** be continued through a reauthentication event — described in Sec. 7.2 — wherein the subscriber repeats some or all of the initial authentication event, thereby re-establishing the session.

Session management is preferable over continual presentation of credentials as the poor usability of continual presentation often creates incentives for workarounds such as caching of activation factors, negating authentication intent and obscuring the freshness of the authentication event.

7.1. Session Bindings

A session occurs between the software that a subscriber is running — such as a browser, application, or operating system (i.e., the session subject) — and the RP or CSP that the subscriber is accessing (i.e., the session host). A session secret **SHALL** be shared between the subscriber's software and the service being accessed. This secret binds the two ends of the session, allowing the subscriber to continue using the service over time. The secret **SHALL** be presented directly by the subscriber's software or possession of the secret **SHALL** be proven using a cryptographic mechanism.

Continuity of authenticated sessions **SHALL** be based upon the possession of a session secret issued by the verifier at the time of authentication and optionally refreshed during the session. The nature of a session depends on the application, such as:

- a web browser session with a "session" cookie, or
- an instance of a mobile application that retains a session secret.

Session secrets **SHALL NOT** be persistent (retained across a restart of the associated application or a reboot of the host device).

The secret used for session binding **SHALL** be generated by the session host in direct response to an authentication event. A session **SHOULD** inherit the AAL properties of the authentication event which triggered its creation. A session **MAY** be considered at a lower AAL than the authentication event but **SHALL NOT** be considered at a higher AAL than the authentication event.

Secrets used for session binding **SHALL** meet all of the following requirements:

1. Secrets are generated by the session host during an interaction, typically immediately following authentication.
2. Secrets are generated by an approved random bit generator [SP800-90Ar1] and contain at least 64 bits of entropy.
3. Secrets are erased or invalidated by the session subject when the subscriber logs out.
4. Secrets are sent to and received from the device using an authenticated protected channel.
5. Secrets will time out and are not accepted after the times specified in Sections 4.1.3, 4.2.3, and 4.3.3, as appropriate for the AAL.
6. Secrets are not made available to insecure communications between the host and subscriber's endpoint.

In addition, secrets used for session binding **SHOULD** be erased on the subscriber endpoint when they log out or when the secret is deemed to have expired. They **SHOULD NOT** be placed in insecure locations such as HTML5 Local Storage due to the potential exposure of local storage to cross-site scripting (XSS) attacks.

Authenticated sessions **SHALL NOT** fall back to an insecure transport, such as from https to http, following authentication.

URLs or POST content **SHALL** contain a session identifier that **SHALL** be verified by the RP to protect against cross-site request forgery.

There are several mechanisms for managing a session over time. The following sections give different examples along with additional requirements and considerations particular to each example technology. Additional informative guidance is available in the OWASP *Session Management Cheat Sheet* [OWASP-session].

7.1.1. Browser Cookies

Browser cookies are the predominant mechanism by which a session will be created and tracked for a subscriber accessing a service. Cookies are not authenticators, but they are suitable as short-term secrets (for the duration of a session).

Cookies used for session maintenance **SHALL** meet all of the following requirements:

1. Cookies are tagged to be accessible only on secure (HTTPS) sessions.
2. Cookies are accessible to the minimum practical set of hostnames and paths.

In addition, session maintenance cookies **SHOULD** be tagged to be inaccessible via JavaScript (HttpOnly). They **SHOULD** contain only an opaque string (such as a session identifier), and **SHOULD NOT** contain cleartext PII. They **SHOULD** be tagged to expire at, or soon after, the session's validity period. This latter requirement is intended to limit the accumulation of cookies, but **SHALL NOT** be depended upon to enforce session timeouts.

7.1.2. Access Tokens

An access token — such as found in OAuth — is used to allow an application to access a set of services on a subscriber's behalf following an authentication event. The presence of an OAuth access token **SHALL NOT** be interpreted by the RP as presence of the subscriber, in the absence of other signals. The OAuth access token, and any associated refresh tokens, **MAY** be valid long after the authentication session has ended and the subscriber has left the application.

7.1.3. Device Identification

Other methods of secure device identification — including but not limited to mutual TLS, token binding, or other mechanisms — **MAY** be used to enact a session between a subscriber and a service.

7.2. Reauthentication

Periodic reauthentication of sessions **SHALL** be performed to confirm the continued presence of the subscriber at an authenticated session (i.e., that the subscriber has not walked away without logging out).

A session **SHALL NOT** be extended past the guidelines in Sections 4.1.3, 4.2.3, and 4.3.3 (depending on AAL) based on presentation of the session secret alone. Prior to session expiration, the reauthentication time limit **SHALL** be extended by prompting the subscriber for the authentication factors specified in Table 2.

When a session has been terminated, due to a time-out or other action, the subscriber **SHALL** be required to establish a new session by authenticating again.

Table 2. AAL Reauthentication Requirements

AAL	Requirement
1	Presentation of any one factor
2	Presentation of a memorized secret or biometric
3	Presentation of all factors

> Note: At AAL2, a memorized secret or biometric, and not a physical authenticator, is required because the session secret is *something you have*, and an additional authentication factor is required to continue the session.

7.2.1. Reauthentication from a Federation or Assertion

When using a federation protocol and Identity Provider (IdP) to authenticate at the RP as described in [SP800-63C], special considerations apply to session management and reauthentication. The federation protocol communicates an authentication event at the IdP to the RP using an assertion, and the RP then begins an authenticated session based on the successful validation of this assertion. Since the IdP and RP manage sessions separately from each other and the federation protocol does not connect the session management between the IdP and RP, the termination of the subscriber's sessions at an IdP and at an RP are independent of each other. Likewise, the subscriber's sessions at multiple different RPs are established and terminated independently of each other.

Consequently, when an RP session expires and the RP requires reauthentication, it is entirely possible that the session at the IdP has not expired and that a new assertion could be generated from this session at the IdP without explicitly reauthenticating the subscriber. The IdP can communicate the time and details of the authentication event to the RP, but it is up to the RP to determine if reauthentication requirements have been met. Section 5.3 of [SP800-63C] provides additional details and requirements for session management within a federation context.

8. Threats and Security Considerations

This section is informative.

8.1. Authenticator Threats

An attacker who can gain control of an authenticator will often be able to masquerade as the authenticator's owner. Threats to authenticators can be categorized based on attacks on the types of authentication factors that comprise the authenticator:

- *Something you know* may be disclosed to an attacker. The attacker might guess a memorized secret. Where the authenticator is a shared secret, the attacker could gain access to the CSP or verifier and obtain the secret value or perform a dictionary attack on a hash of that value. An attacker may observe the entry of a PIN or passcode, find a written record or journal entry of a PIN or passcode, or may install malicious software (e.g., a keyboard logger) to capture the secret. Additionally, an attacker may determine the secret through offline attacks on a password database maintained by the verifier.
- *Something you have* may be lost, damaged, stolen from the owner, or cloned by an attacker. For example, an attacker who gains access to the owner's computer might copy a software authenticator. A hardware authenticator might be stolen, tampered with, or duplicated. Out-of-band secrets may be intercepted by an attacker and used to authenticate their own session.
- *Something you are* may be replicated. For example, an attacker may obtain a copy of the subscriber's fingerprint and construct a replica.

This document assumes that the subscriber is not colluding with an attacker who is attempting to falsely authenticate to the verifier. With this assumption in mind, the threats to the authenticators used for digital authentication are listed in Table 3, along with some examples.

Table 3. Authenticator Threats

Authenticator Threat/Attack	Description	Examples
Assertion Manufacture or Modification	The attacker generates a false assertion	Compromised CSP asserts identity of a claimant who has not properly authenticated
	The attacker modifies an existing assertion	Compromised proxy that changes AAL of an authentication assertion
Theft	A physical authenticator is stolen by an Attacker.	A hardware cryptographic device is stolen.
		An OTP device is stolen.

		A look-up secret authenticator is stolen.
		A cell phone is stolen.
Duplication	The subscriber's authenticator has been copied with or without their knowledge.	Passwords written on paper are disclosed.
		Passwords stored in an electronic file are copied.
		Software PKI authenticator (private key) copied.
		Look-up secret authenticator copied.
		Counterfeit biometric authenticator manufactured.
Eavesdropping	The authenticator secret or authenticator output is revealed to the attacker as the subscriber is authenticating.	Memorized secrets are obtained by watching keyboard entry.
		Memorized secrets or authenticator outputs are intercepted by keystroke logging software.
		A PIN is captured from a PIN pad device.
		A hashed password is obtained and used by an attacker for another authentication (*pass-the-hash attack*).
	An out-of-band secret is intercepted by the attacker by compromising the communication channel.	An out-of-band secret is transmitted via unencrypted Wi-Fi and received by the attacker.
Offline Cracking	The authenticator is exposed using analytical methods outside the authentication mechanism.	A software PKI authenticator is subjected to dictionary attack to identify the correct password to use to decrypt the private key.
Side Channel Attack	The authenticator secret is exposed using physical characteristics of the authenticator.	A key is extracted by differential power analysis on a hardware cryptographic authenticator.

		A cryptographic authenticator secret is extracted by analysis of the response time of the authenticator over a number of attempts.
Phishing or Pharming	The authenticator output 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 memorized secret is revealed by a bank subscriber in response to an email inquiry from a phisher pretending to represent the bank.
		A memorized secret is revealed by the subscriber at a bogus verifier website reached through DNS spoofing.
Social Engineering	The attacker establishes a level of trust with a subscriber in order to convince the subscriber to reveal their authenticator secret or authenticator output.	A memorized secret is revealed by the subscriber to an officemate asking for the password on behalf of the subscriber's boss.
		A memorized secret is revealed by a subscriber in a telephone inquiry from an attacker masquerading as a system administrator.
		An out of band secret sent via SMS is received by an attacker who has convinced the mobile operator to redirect the victim's mobile phone to the attacker.
Online Guessing	The attacker connects to the verifier online and attempts to guess a valid authenticator output in the context of that verifier.	Online dictionary attacks are used to guess memorized secrets.

		Online guessing is used to guess authenticator outputs for an OTP device registered to a legitimate claimant.
Endpoint Compromise	Malicious code on the endpoint proxies remote access to a connected authenticator without the subscriber's consent.	A cryptographic authenticator connected to the endpoint is used to authenticate remote attackers.
	Malicious code on the endpoint causes authentication to other than the intended verifier.	Authentication is performed on behalf of an attacker rather than the subscriber.
		A malicious app on the endpoint reads an out-of-band secret sent via SMS and the attacker uses the secret to authenticate.
	Malicious code on the endpoint compromises a multi-factor software cryptographic authenticator.	Malicious code proxies authentication or exports authenticator keys from the endpoint.
Unauthorized Binding	An attacker is able to cause an authenticator under their control to be bound to a subscriber account.	An attacker intercepts an authenticator or provisioning key en route to the subscriber.

8.2. Threat Mitigation Strategies

Related mechanisms that assist in mitigating the threats identified above are summarized in Table 4.

Table 4. Mitigating Authenticator Threats

Authenticator Threat/Attack	**Threat Mitigation Mechanisms**	**Normative References**
Theft	Use multi-factor authenticators that need to be activated through a memorized secret or biometric.	4.2.1, 4.3.1
	Use a combination of authenticators that includes a memorized secret or biometric.	4.2.1, 4.3.1

Duplication	Use authenticators from which it is difficult to extract and duplicate long-term authentication secrets.	4.2.2, 4.3.2, 5.1.7.1
Eavesdropping	Ensure the security of the endpoint, especially with respect to freedom from malware such as key loggers, prior to use.	4.2.2
	Avoid use of unauthenticated and unencrypted communication channels to send out-of-band authenticator secrets.	5.1.3.1
	Authenticate over authenticated protected channels (e.g., observe lock icon in browser window).	4.1.2, 4.2.2, 4.3.2
	Use authentication protocols that are resistant to replay attacks such as *pass-the-hash*.	5.2.8
	Use authentication endpoints that employ trusted input and trusted display capabilities.	5.1.6.1, 5.1.8.1
Offline Cracking	Use an authenticator with a high entropy authenticator secret.	5.1.2.1, 5.1.4.1, 5.1.5.1, 5.1.7.1, 5.1.9.1
	Store centrally verified memorized secrets in a salted, hashed form, including a keyed hash.	5.1.1.1.2, 5.2.7
Side Channel Attack	Use authenticator algorithms that are designed to maintain constant power consumption and timing regardless of secret values.	4.3.2
Phishing or Pharming	Use authenticators that provide phishing resistance.	5.2.5

Social Engineering	Avoid use of authenticators that present a risk of social engineering of third parties such as customer service agents.	6.1.2.1, 6.1.2.3
Online Guessing	Use authenticators that generate high entropy output.	5.1.2.1, 5.1.7.1, 5.1.9.1
	Use an authenticator that locks up after a number of repeated failed activation attempts.	5.2.2
Endpoint Compromise	Use hardware authenticators that require physical action by the subscriber.	5.2.9
	Maintain software-based keys in restricted-access storage.	5.1.3.1, 5.1.6.1, 5.1.8.1
Unauthorized Binding	Use AitM-resistant protocols for provisioning of authenticators and associated keys.	6.1

Several other strategies may be applied to mitigate the threats described in Table 3:

- *Multiple factors* make successful attacks more difficult to accomplish. If an attacker needs to both steal a cryptographic authenticator and guess a memorized secret, then the work to discover both factors may be too high.
- *Physical security mechanisms* may be employed to protect a stolen authenticator from duplication. Physical security mechanisms can provide tamper evidence, detection, and response.
- *Requiring the use of long memorized secrets* that don't appear in common dictionaries may force attackers to try every possible value.
- *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 subscribers understand 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 authentication process.
- *Out of band techniques* may be employed to verify proof of possession of registered devices (e.g., cell phones).

8.3. Authenticator Recovery

The weak point in many authentication mechanisms is the process followed when a subscriber loses control of one or more authenticators and needs to replace them. In many cases, the options remaining available to authenticate the subscriber are limited, and economic concerns (e.g., cost of maintaining call centers) motivate the use of inexpensive, and often less secure, backup authentication methods. To the extent that authenticator recovery is human-assisted, there is also the risk of social engineering attacks.

To maintain the integrity of the authentication factors, it is essential that it not be possible to leverage an authentication involving one factor to obtain an authenticator of a different factor. For example, a memorized secret must not be usable to obtain a new list of look-up secrets.

8.4. Session Attacks

The above discussion focuses on threats to the authentication event itself, but hijacking attacks on the session following an authentication event can have similar security impacts. The session management guidelines in Sec. 7 are essential to maintain session integrity against attacks, such as XSS. In addition, it is important to sanitize all information to be displayed [OWASP-XSS-prevention] to ensure that it does not contain executable content. These guidelines also recommend that session secrets be made inaccessible to mobile code in order to provide extra protection against exfiltration of session secrets.

Another post-authentication threat, cross-site request forgery (CSRF), takes advantage of users' tendency to have multiple sessions active at the same time. It is important to embed and verify a session identifier into web requests to prevent the ability for a valid URL or request to be unintentionally or maliciously activated.

9. Privacy Considerations

These privacy considerations supplement the guidance in Sec. 4. This section is informative.

9.1. Privacy Risk Assessment

Sections 4.1.5, 4.2.5, and 4.3.5 require the CSP to conduct a privacy risk assessment for records retention. Such a privacy risk assessment would include:

1. The likelihood that the records retention could create a problem for the subscriber, such as invasiveness or unauthorized access to the information.
2. The impact if such a problem did occur.

CSPs should be able to reasonably justify any response they take to identified privacy risks, including accepting the risk, mitigating the risk, and sharing the risk. The use of subscriber consent is a form of sharing the risk, and therefore appropriate for use only when a subscriber could reasonably be expected to have the capacity to assess and accept the shared risk.

9.2. Privacy Controls

Section 4.4 requires CSPs to employ appropriately tailored privacy controls. [SP800-53] provides a set of privacy controls for CSPs to consider when deploying authentication mechanisms. These controls cover notices, redress, and other important considerations for successful and trustworthy deployments.

9.3. Use Limitation

Section 4.4 requires CSPs to use measures to maintain the objectives of predictability (enabling reliable assumptions by individuals, owners, and operators about PII and its processing by an information system) and manageability (providing the capability for granular administration of PII, including alteration, deletion, and selective disclosure) commensurate with privacy risks that can arise from the processing of attributes for purposes other than identity proofing, authentication, authorization, or attribute assertion, related fraud mitigation, or to comply with law or legal process [NISTIR8062].

CSPs may have various business purposes for processing attributes, including providing non-identity services to subscribers. However, processing attributes for other purposes than those specified at collection can create privacy risks when individuals are not expecting or comfortable with the additional processing. CSPs can determine appropriate measures commensurate with the privacy risk arising from the additional processing. For example, absent applicable law, regulation or policy, it may not be necessary to get consent when processing attributes to provide non-identity services requested by subscribers, although notices may help subscribers maintain reliable assumptions about the processing (predictability). Other processing of attributes may carry different privacy risks that call for obtaining consent or allowing subscribers more control over the use or disclosure of specific attributes (manageability). Subscriber consent needs to be meaningful; therefore, as stated in Sec. 4.4, when CSPs use consent measures, acceptance by the subscriber of additional uses shall not be a condition of providing authentication services.

Consult the agency SAOP if there are questions about whether the proposed processing falls outside the scope of the permitted processing or the appropriate privacy risk mitigation measures.

9.4. Agency-Specific Privacy Compliance

Section 4.4 covers specific compliance obligations for federal CSPs. It is critical to involve the agency SAOP in the earliest stages of digital authentication system development in order to assess and mitigate privacy risks and advise the agency on compliance requirements, such as whether or not the collection of PII to issue or maintain authenticators triggers the *Privacy Act of 1974* [PrivacyAct] or the *E-Government Act of 2002* [E-Gov] requirement to conduct a PIA. For example, with respect to centralized maintenance of biometrics, it is likely that the Privacy Act requirements will be triggered and require coverage by either a new or existing Privacy Act system of records due to the collection and maintenance of PII and any other attributes necessary for authentication. The SAOP can similarly assist the agency in determining whether a PIA is required.

These considerations should not be read as a requirement to develop a Privacy Act SORN or PIA for authentication alone. In many cases it will make the most sense to draft a PIA and SORN that encompasses the entire digital identity process or include the digital authentication process as part of a larger programmatic PIA that discusses the online service or benefit that the agency is establishing.

Due to the many components of digital authentication, it is important for the SAOP to have an awareness and understanding of each individual component. For example, other privacy artifacts may be applicable to an agency offering or using federated CSP or RP services (e.g., Data Use Agreements, Computer Matching Agreements). The SAOP can assist the agency in determining what additional requirements apply. Moreover, a thorough understanding of the individual components of digital authentication will enable the SAOP to thoroughly assess and mitigate privacy risks either through compliance processes or by other means.

10. Usability Considerations

This section is informative.

Note: In this section, the term *users* means *claimants* or *subscribers.*

[ISO/IEC9241-11] defines usability as the "extent to which a system, product, or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." This definition focuses on users, their goals, and the context of use as key elements necessary for achieving effectiveness, efficiency, and satisfaction. A holistic approach that accounts for these key elements is necessary to achieve usability.

A user's goal for accessing an information system is to perform an intended task. Authentication is the function that enables this goal. However, from the user's perspective, authentication stands between them and their intended task. Effective design and implementation of authentication makes it easy to do the right thing, hard to do the wrong thing, and easy to recover when the wrong thing happens.

Organizations need to be cognizant of the overall implications of their stakeholders' entire digital authentication ecosystem. Users often employ multiple authenticators, each for a different RP. They then struggle to remember passwords, to recall which authenticator goes with which RP, and to carry multiple physical authentication devices. Evaluating the usability of authentication is critical, as poor usability often results in coping mechanisms and unintended workarounds that can ultimately degrade the effectiveness of security controls.

Integrating usability into the development process can lead to authentication solutions that are secure and usable while still addressing users' authentication needs and organizations' business goals.

The impact of usability across digital systems needs to be considered as part of the risk assessment when deciding on the appropriate AAL. Authenticators with a higher AAL sometimes offer better usability and should be allowed for use with lower AAL applications.

Leveraging federation for authentication can alleviate many of the usability issues, though such an approach has its own tradeoffs, as discussed in [SP800-63C].

This section provides general usability considerations and possible implementations, but does not recommend specific solutions. The implementations mentioned are examples to encourage innovative technological approaches to address specific usability needs. Further, usability considerations and their implementations are sensitive to many factors that prevent a one-size-fits-all solution. For example, a font size that works in the desktop computing environment may force text to scroll off of a small OTP device screen. Performing a usability evaluation on the selected authenticator is a critical component of implementation. It is important to conduct evaluations with representative users, realistic goals and tasks, and appropriate contexts of use.

Guidelines and considerations are described from the users' perspective.

Accessibility differs from usability and is out of scope for this document. Section 508 [Section508] was enacted to eliminate barriers in information technology and require federal government agencies to make their online public content accessible to people with disabilities. Refer to Section 508 law and standards for accessibility guidance.

10.1. Usability Considerations Common to Authenticators

When selecting and implementing an authentication system, consider usability across the entire lifecycle of the selected authenticators (e.g., typical use and intermittent events), while being mindful of the combination of users, their goals, and context of use.

A single authenticator type usually does not suffice for the entire user population. Therefore, whenever possible — based on AAL requirements — CSPs should support alternative authenticator types and allow users to choose based on their needs. Task immediacy, perceived cost benefit tradeoffs, and unfamiliarity with certain authenticators often impact choice. Users tend to choose options that incur the least burden or cost at that moment. For example, if a task requires immediate access to an information system, a user may prefer to create a new subscriber account and password rather than select an authenticator requiring more steps. Alternatively, users may choose a federated identity option — approved at the appropriate AAL — if they already have a subscriber account with an identity provider. Users may understand some authenticators better than others, and have different levels of trust based on their understanding and experience.

Positive user authentication experiences are integral to the success of an organization achieving desired business outcomes. Therefore, they should strive to consider authenticators from the users' perspective. The overarching authentication usability goal is to minimize user burden and authentication friction (e.g., the number of times a user has to authenticate, the steps involved, and the amount of information they have to track). Single sign-on exemplifies one such minimization strategy.

Usability considerations applicable to most authenticators are described below. Subsequent sections describe usability considerations specific to a particular authenticator.

Usability considerations for typical usage of all authenticators include:

- Provide information on the use and maintenance of the authenticator, e.g., what to do if the authenticator is lost or stolen, and instructions for use — especially if there are different requirements for first-time use or initialization.

- Authenticator availability should also be considered as users will need to remember to have their authenticator readily available. Consider the need for alternate authentication options to protect against loss, damage, or other negative impacts to the original authenticator.
- Whenever possible, based on AAL requirements, users should be provided with alternate authentication options. This allows users to choose an authenticator based on their context, goals, and tasks (e.g., the frequency and immediacy of the task). Alternate authentication options also help address availability issues that may occur with a particular authenticator.
- Characteristics of user-facing text:
  - Write user-facing text (e.g., instructions, prompts, notifications, error messages) in plain language for the intended audience. Avoid technical jargon and write for the audience's expected literacy level.
  - Consider the legibility of user-facing and user-entered text, including font style, size, color, and contrast with surrounding background. Illegible text contributes to user entry errors. To enhance legibility, consider the use of:
    * High contrast. The highest contrast is black on white.
    * Sans serif fonts for electronic displays. Serif fonts for printed materials.
    * Fonts that clearly distinguish between easily confusable characters (e.g., the capital letter "O" and the number "0").
    * A minimum font size of 12 points as long as the text fits for display on the device.
- User experience during authenticator entry:
  - Offer the option to display text during entry, as masked text entry is error-prone. Once a given character is displayed long enough for the user to see, it can be hidden. Consider the device when determining masking delay time, as it takes longer to enter memorized secrets on mobile devices (e.g., tablets and smartphones) than on traditional desktop computers. Ensure masking delay durations are consistent with user needs.
  - Ensure the time allowed for text entry is adequate (i.e., the entry screen does not time out prematurely). Ensure allowed text entry times are consistent with user needs.
  - Provide clear, meaningful and actionable feedback on entry errors to reduce user confusion and frustration. Significant usability implications arise when users do not know they have entered text incorrectly.

- Allow at least 10 entry attempts for authenticators requiring the entry of the authenticator output by the user. The longer and more complex the entry text, the greater the likelihood of user entry errors.
- Provide clear, meaningful feedback on the number of remaining allowed attempts. For rate limiting (i.e., throttling), inform users how long they have to wait until the next attempt to reduce confusion and frustration.

• Minimize the impact of form-factor constraints, such as limited touch and display areas on mobile devices:
  - Larger touch areas improve usability for text entry since typing on small devices is significantly more error prone and time consuming than typing on a full-size keyboard. The smaller the onscreen keyboard, the more difficult it is to type, due to the size of the input mechanism (e.g., a finger) relative to the size of the on-screen target.
  - Follow good user interface and information design for small displays.

Intermittent events include events such as reauthentication, subscriber account lock-out, expiration, revocation, damage, loss, theft, and non-functional software.

Usability considerations for intermittent events across authenticator types include:

- To prevent users from needing to reauthenticate due to user inactivity, prompt users in order to trigger activity just before (e.g., 2 minutes) an inactivity timeout would otherwise occur.
- Prompt users with adequate time (e.g., 1 hour) to save their work before the fixed periodic reauthentication event required regardless of user activity.
- Clearly communicate how and where to acquire technical assistance. For example, provide users with information such as a link to an online self-service feature, chat sessions or a phone number for help desk support. Ideally, sufficient information can be provided to enable users to recover from intermittent events on their own without outside intervention.

10.2. Usability Considerations by Authenticator Type

In addition to the previously described general usability considerations applicable to most authenticators (Sec. 10.1), the following sections describe other usability considerations specific to particular authenticator types.

10.2.1. Memorized Secrets

Typical Usage

Users manually input the memorized secret (commonly referred to as a password or PIN).

Usability considerations for typical usage include:

- Memorability of the memorized secret
  - The likelihood of recall failure increases as there are more items for users to remember. With fewer memorized secrets, users can more easily recall the specific memorized secret needed for a particular RP.
  - The memory burden is greater for a less frequently used password.
- User experience during entry of the memorized secret
  - Support copy and paste functionality in fields for entering memorized secrets, including passphrases.

Intermittent Events

Usability considerations for intermittent events include:

- When users create and change memorized secrets:
  - Clearly communicate information on how to create and change memorized secrets.
  - Clearly communicate memorized secret requirements, as specified in Sec. 5.1.1.
  - Allow at least 64 characters in length to support the use of passphrases. Encourage users to make memorized secrets as lengthy as they want, using any characters they like (including spaces), thus aiding memorization.
  - Do not impose other composition rules (e.g. mixtures of different character types) on memorized secrets.
  - Do not require that memorized secrets be changed arbitrarily (e.g., periodically) unless there is a user request or evidence of authenticator compromise. (See Sec. 5.1.1 for additional information).
- Provide clear, meaningful and actionable feedback when chosen passwords are rejected (e.g., when it appears on a "blocklist" of unacceptable passwords or has been used previously).

10.2.2. Look-Up Secrets

Typical Usage

Users use the authenticator — printed or electronic — to look up the appropriate secret(s) needed to respond to a verifier's prompt. For example, a user may be asked to provide a specific subset of the numeric or character strings printed on a card in table format.

Usability considerations for typical usage include:

- User experience during entry of look-up secrets.

- Consider the prompts' complexity and size. The larger the subset of secrets a user is prompted to look up, the greater the usability implications. Both the cognitive workload and physical difficulty for entry should be taken into account when selecting the quantity and complexity of look-up secrets for authentication.

10.2.3. Out-of-Band

Typical Usage

Out-of-band authentication requires users have access to a primary and secondary communication channel.

Usability considerations for typical usage:

- Notify users of the receipt of a secret on a locked device. However, if the out-of-band device is locked, authentication to the device should be required to access the secret.
- Depending on the implementation, consider form-factor constraints as they are particularly problematic when users must enter text on mobile devices. Providing larger touch areas will improve usability for entering secrets on mobile devices.
- A better usability option is to offer features that do not require text entry on mobile devices (e.g., a single tap on the screen, or a copy feature so users can copy and paste out-of-band secrets). Providing users such features is particularly helpful when the primary and secondary channels are on the same device. For example, it is difficult for users to transfer the authentication secret on a smartphone because they must switch back and forth — potentially multiple times — between the out-of-band application and the primary channel.

10.2.4. Single-Factor OTP Device

Typical Usage

Users access the OTP generated by the single-factor OTP device. The authenticator output is typically displayed on the device and the user enters it for the verifier.

Usability considerations for typical usage include:

- Authenticator output allows at least one minute between changes, but ideally allows users the full two minutes as specified in Sec. 5.1.4.1. Users need adequate time to enter the authenticator output (including looking back and forth between the single-factor OTP device and the entry screen).
- Depending on the implementation, the following are additional usability considerations for implementers:

- If the single-factor OTP device supplies its output via an electronic interface (e.g, USB) this is preferable since users do not have to manually enter the authenticator output. However, if a physical input (e.g., pressing a button) is required to operate, the location of the USB ports could pose usability difficulties. For example, the USB ports of some computers are located on the back of the computer and will be difficult for users to reach.
- Limited availability of a direct computer interface such as a USB port could pose usability difficulties. For example, the number of USB ports on laptop computers is often very limited. This may force users to unplug other USB peripherals in order to use the single-factor OTP device.

10.2.5. Multi-Factor OTP Device

Typical Usage

Users access the OTP generated by the multi-factor OTP device through a second authentication factor. The OTP is typically displayed on the device and the user manually enters it for the verifier. The second authentication factor may be achieved through some kind of integral entry pad to enter a memorized secret, an integral biometric (e.g., fingerprint) reader, or a direct computer interface (e.g., USB port). Usability considerations for the additional factor apply as well — see Sec. 10.2.1 for memorized secrets and Sec. 10.4 for biometrics used in multi-factor authenticators.

Usability considerations for typical usage include:

- User experience during manual entry of the authenticator output.
  - For time-based OTP, provide a grace period in addition to the time during which the OTP is displayed. Users need adequate time to enter the authenticator output, including looking back and forth between the multi-factor OTP device and the entry screen.
  - Consider form-factor constraints if users must unlock the multi-factor OTP device via an integral entry pad or enter the authenticator output on mobile devices. Typing on small devices is significantly more error prone and time-consuming than typing on a traditional keyboard. The smaller the integral entry pad and onscreen keyboard, the more difficult it is to type. Providing larger touch areas improves usability for unlocking the multi-factor OTP device or entering the authenticator output on mobile devices.
  - Limited availability of a direct computer interface like a USB port could pose usability difficulties. For example, laptop computers often have a limited number of USB ports, which may force users to unplug other USB peripherals to use the multi-factor OTP device.

10.2.6. Single-Factor Cryptographic Software

Typical Usage

Users authenticate by proving possession and control of the cryptographic software key.

Usability considerations for typical usage include:

- Give cryptographic keys appropriately descriptive names that are meaningful to users since users have to recognize and recall which cryptographic key to use for which authentication task. This prevents users from having to deal with multiple similarly and ambiguously named cryptographic keys. Selecting from multiple cryptographic keys on smaller mobile devices may be particularly problematic if the names of the cryptographic keys are shortened due to reduced screen size.

10.2.7. Single-Factor Cryptographic Device

Typical Usage

Users authenticate by proving possession of the single-factor cryptographic device.

Usability considerations for typical usage include:

- Requiring a physical input (e.g., pressing a button) to operate the single-factor cryptographic device could pose usability difficulties. For example, some USB ports are located on the back of computers, making it difficult for users to reach.
- Limited availability of a direct computer interface like a USB port could pose usability difficulties. For example, laptop computers often have a limited number of USB ports, which may force users to unplug other USB peripherals to use the single-factor cryptographic device.

10.2.8. Multi-Factor Cryptographic Software

Typical Usage

In order to authenticate, users prove possession and control of the cryptographic key stored on disk or some other "soft" media that requires activation. The activation is through the input of a second authentication factor, either a memorized secret or a biometric characteristic. Usability considerations for the additional factor apply as well — see Sec. 10.2.1 for memorized secrets and Sec. 10.4 for biometrics used in multi-factor authenticators.

Usability considerations for typical usage include:

- Give cryptographic keys appropriately descriptive names that are meaningful to users since users have to recognize and recall which cryptographic key to use for which authentication task. This prevents users from having to deal with multiple similarly and ambiguously named cryptographic keys. Selecting from multiple cryptographic keys on smaller mobile devices may be particularly problematic if the names of the cryptographic keys areas shortened due to reduced screen size.

10.2.9. Multi-Factor Cryptographic Device

Typical Usage

Users authenticate by proving possession of the multi-factor cryptographic device and control of the protected cryptographic key. The device is activated by a second authentication factor, either a memorized secret or a biometric. Usability considerations for the additional factor apply as well — see Sec. 10.2.1 for memorized secrets and Sec. 10.4 for biometrics used in multi-factor authenticators.

Usability considerations for typical usage include:

- Do not require users to keep multi-factor cryptographic devices connected following authentication. Users may forget to disconnect the multi-factor cryptographic device when they are done with it (e.g., forgetting a smartcard in the smartcard reader and walking away from the computer).
  - Users need to be informed regarding whether the multi-factor cryptographic device is required to stay connected or not.
- Give cryptographic keys appropriately descriptive names that are meaningful to users since users have to recognize and recall which cryptographic key to use for which authentication task. This prevents users being faced with multiple similarly and ambiguously named cryptographic keys. Selecting from multiple cryptographic keys on smaller mobile devices (such as smartphones) may be particularly problematic if the names of the cryptographic keys are shortened due to reduced screen size.
- Limited availability of a direct computer interface like a USB port could pose usability difficulties. For example, laptop computers often have a limited number of USB ports, which may force users to unplug other USB peripherals to use the multi-factor cryptographic device.

10.3. Summary of Usability Considerations

Figure 3 summarizes the usability considerations for typical usage and intermittent events for each authenticator type. Many of the usability considerations for typical usage apply to most of the authenticator types, as demonstrated in the rows. The table highlights common and divergent usability characteristics across the authenticator types. Each column allows readers to easily identify the usability attributes to address for each authenticator. Depending on users' goals and context of use, certain attributes may be valued over others. Whenever possible, provide alternative authenticator types and allow users to choose between them.

Multi-factor authenticators (e.g., multi-factor OTP devices, multi-factor cryptographic software, and multi-factor cryptographic devices) also inherit their secondary factor's usability considerations. As biometrics are only allowed as an activation factor in multi-factor authentication solutions, usability considerations for biometrics are not included in Figure 3 and are discussed in Sec. 10.4.

Usability Considerations	Memorized secrets	Look-up Secrets	Out of Band	Single Factor OTP Device	Multi-Factor OTP Device	Single Factor Cryptographic Software	Single Factor Cryptographic Device	Multi-Factor Cryptographic Software	Multi-Factor Cryptographic Device
Typical usage									
Authenticator availability – authenticators readily in user's possession	◆	◆	◆	◆	◆	◆	◆	◆	◆
Plain language for user facing text (e.g., instructions, prompts, notifications, error messages)	◆	◆	◆	◆	◆	◆	◆	◆	◆
Legibility of user facing text or text entered by users	◆	◆	◆	◆	◆	◆	◆	◆	◆
Unmasked text entry		◆	◆	◆	◆				
Support text entry – length of 64 characters, copy and paste	◆								
Delayed masking during text entry	◆								
Adequate time allowed for text entry	◆	◆	◆	◆	◆				
Entry errors – need clear and meaningful feedback	◆	◆	◆	◆	◆				
Minimum of 10 attempts allowed	◆	◆	◆	◆	◆				
Remaining allowed attempts – need clear and meaningful feedback	◆	◆	◆	◆	◆				
Form-factor constraints	◆	◆	◆	◆	◆	◆	◆	◆	◆
Location and availability of a direct computer interface such as a USB port				◆	◆		◆		◆
Physical input required (such as pressing a button)				◆			◆		
Cryptographic keys need for descriptive and meaningful names						◆		◆	◆
Complexity and size of the prompts		◆							
Authentication to secondary device to access the authentication secret			◆						
Continuous hardware connection not required									◆
Intermittent Events									
Reauthentication due to user inactivity	◆	◆	◆	◆	◆	◆	◆	◆	◆
Fixed periodic reauthentication	◆	◆	◆	◆	◆	◆	◆	◆	◆
Provisions for technical assistance	◆	◆	◆	◆	◆	◆	◆	◆	◆
Provisions to create and change memorized secrets	◆								

Figure 3. Usability Considerations Summary by Authenticator Type

10.4. Biometrics Usability Considerations

This section provides a high-level overview of general usability considerations for biometrics. A more detailed discussion of biometric usability can be found in *Usability & Biometrics, Ensuring Successful Biometric Systems* [UsabilityBiometrics].

Although there are other biometric modalities, the following three biometric modalities are more commonly used for authentication: fingerprint, face and iris.

Typical Usage

- For all modalities, user familiarity and practice with the device improves performance.
- Device affordances (i.e., properties of a device that allow a user to perform an action), feedback, and clear instructions are critical to a user's success with the biometric device. For example, provide clear instructions on the required actions for liveness detection.
- Ideally, users can select the modality they are most comfortable with for their second authentication factor. The user population may be more comfortable and familiar with — and accepting of — some biometric modalities than others.
- User experience with biometrics as an activation factor.
    - Provide clear, meaningful feedback on the number of remaining allowed attempts. For example, for rate limiting (i.e., throttling), inform users of the time period they have to wait until next attempt to reduce user confusion and frustration.
- Fingerprint Usability Considerations:
    - Users have to remember which finger(s) they used for initial enrollment.
    - The amount of moisture on the finger(s) affects the sensor's ability for successful capture.
    - Additional factors influencing fingerprint capture quality include age, gender, and occupation (e.g., users handling chemicals or working extensively with their hands may have degraded friction ridges).
- Face Usability Considerations:
    - Users have to remember whether they wore any artifacts (e.g., glasses) during enrollment because it affects facial recognition accuracy.
    - Differences in environmental lighting conditions can affect facial recognition accuracy.
    - Facial expressions affect facial recognition accuracy (e.g., smiling versus neutral expression).

    - Facial poses affect facial recognition accuracy (e.g., looking down or away from the camera).

- Iris Usability Considerations:
    - Wearing colored contacts may affect the iris recognition accuracy.
    - Users who have had eye surgery may need to re-enroll post-surgery.
    - Differences in environmental lighting conditions can affect iris recognition accuracy, especially for certain iris colors.

Intermittent Events

As biometrics are only permitted as a second factor for multi-factor authentication, usability considerations for intermittent events with the primary factor still apply. Intermittent events with biometrics use include, but are not limited to, the following, which may affect recognition accuracy:

- If users injure their enrolled finger(s), fingerprint recognition may not work. Fingerprint authentication will be difficult for users with degraded fingerprints.
- The time elapsed between the time of facial recognition for authentication and the time of the initial enrollment can affect recognition accuracy as a user's face changes naturally over time. A user's weight change may also be a factor.
- Iris recognition may not work for people who had eye surgery, unless they re-enroll.

Across all biometric modalities, usability considerations for intermittent events include:

- An alternative authentication method must be available and functioning. In cases where biometrics do not work, allow users to use a memorized secret as an alternative second factor.
- Provisions for technical assistance:
    - Clearly communicate information on how and where to acquire technical assistance. For example, provide users information such as a link to an online self-service feature and a phone number for help desk support. Ideally, provide sufficient information to enable users to recover from intermittent events on their own without outside intervention.
    - Inform users of factors that may affect the sensitivity of the biometric sensor (e.g., cleanliness of the sensor).

11. Equity Considerations

This section is informative.

Accurate and equitable authentication service is an essential element of a digital identity system. While the accuracy aspects of authentication are largely the subject of the security requirements found elsewhere in this document, the ability for all subscribers to authenticate reliably is required to provide equitable access to government services as specified in Executive Order 13985, "Advancing Racial Equity and Support for Underserved Communities Through the Federal Government" [EO13985]. In assessing equity risks, a CSP should consider the overall user population served by its authentication service. Additionally, the CSP further identifies groups of users within the population whose shared characteristic(s) can cause them to be subject to inequitable access, treatment, or outcomes when using that service. The usability considerations provided in Sec. 10 should also be considered to help ensure the overall usability and equity for all persons using authentication services.

A primary aspect of equity is that the CSP needs to anticipate the needs of its subscriber population and offer authenticator options that are suitable for that population. Some examples of authenticator suitability problems are as follows:

- SMS-based out-of-band authentication may not be usable for subscribers in rural areas where mobile phone service is not available.
- OTP devices may be difficult for subscribers with vision difficulties to read.
- Out-of-band authentication secrets sent via a voice telephone call may be difficult for subscribers with hearing difficulties to understand.
- Facial matching algorithms may less effectively match facial characteristics of subscribers of some ethnicities.
- The cost of hardware-based authenticators may be beyond the means of some subscribers.
- Accurate manual entry of memorized secrets may be difficult for subscribers with some mobility and dexterity-related physical disabilities.
- The use of certain authenticator types may be challenging for subscribers with some disabilities such as intellectual, developmental, learning, and neurocognitive difficulties.

Normative requirements have been established requiring CSPs to mitigate the problems in this area that are expected to be most common. However, it is not feasible to anticipate all potential equity problems. Potential equity problems also will vary for different applications. Accordingly, CSPs need to provide mechanisms for subscribers to report inequitable authentication requirements and to advise them on potential alternative authentication strategies.

This guideline recommends the binding of additional authenticators to minimize the need for account recovery (see Sec. 6.1.2.3). However, a subscriber might find it difficult to purchase a second hardware-based authenticator as a backup. This inequity can be addressed by making inexpensive authenticators such as look-up secrets (see Sec. 5.1.2) available for use in the event of a primary authenticator failure or loss.

CSPs need to be responsive to subscribers that experience authentication challenges that cannot be solved using authenticators they currently support. This might involve supporting a new authenticator type or allowing federated authentication through a trusted service that meets the needs of the subscriber.

References

This section is informative.

General References

[Argon2] Biryukov, A., Dinu, D., Khovratovich, D., and S. Josefsson, "Argon2 Memory-Hard Function for Password Hashing and Proof-of-Work Applications", RFC 9106, DOI 10.17487/RFC9106, September 2021, https://www.rfc-editor.org/info/rfc9106.

[Blocklists] Habib, Hana, Jessica Colnago, William Melicher, Blase Ur, Sean Segreti, Lujo Bauer, Nicolas Christin, and Lorrie Cranor. "Password Creation in the Presence of Blacklists," 2017. Available at: https://www.ndss-symposium.org/wp-content/uploads/2017/09/usec2017_01_3_Habib_paper.pdf

[Composition] Komanduri, Saranga, Richard Shay, Patrick Gage Kelley, Michelle L Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. "Of Passwords and People: Measuring the Effect of Password-Composition Policies." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2595–2604. ACM, 2011. Available at: https://www.ece.cmu.edu/~lbauer/papers/2011/chi2011-passwords.pdf.

[E-Gov] ***E-Government Act*** **(includes FISMA)** (P.L. 107-347), December 2002, available at: https://www.gpo.gov/fdsys/pkg/PLAW-107publ347/pdf/PLAW-107publ347.pdf.

[EO13681] Executive Order 13681, *Improving the Security of Consumer Financial Transactions*, October 17, 2014, available at: https://www.federalregister.gov/d/2014-25439.

[EO13985] Executive Order 13985, *Advancing Racial Equity and Support for Underserved Communities Through the Federal Government*, January 25, 2021, available at: https://www.federalregister.gov/d/2021-01753.

[FEDRAMP] General Services Administration, *Federal Risk and Authorization Management Program*, available at: https://www.fedramp.gov/.

[M-22-09] OMB Memorandum M-22-09, *Moving the U.S. Government Toward Zero Trust Cybersecurity Principles*, January 26, 2022, available at: https://www.whitehouse.gov/wp-content/uploads/2022/01/M-22-09.pdf.

[NISTIR8062] NIST Internal Report 8062, *An Introduction to Privacy Engineering and Risk Management in Federal Systems*, January 2017, available at: https://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8062.pdf.

[UsabilityBiometrics] National Institute and Standards and Technology, *Usability & Biometrics, Ensuring Successful Biometric Systems*, June 11, 2008, available at:

https://www.nist.gov/customcf/get_pdf.cfm?pub_id=152184.

[OWASP-session] Open Web Application Security Project, *Session Management Cheat Sheet*, available at: https://www.owasp.org/index.php/Session_Management_Cheat_Sheet.

[OWASP-XSS-prevention] Open Web Application Security Project, *XSS (Cross Site Scripting) Prevention Cheat Sheet*, available at: https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet.

[Persistence] herley, cormac, and Paul van Oorschot. "A Research Agenda Acknowledging the Persistence of Passwords," IEEE Security&Privacy Magazine, 2012. Available at: https://research.microsoft.com/apps/pubs/default.aspx?id=154077.

[Policies] Weir, Matt, Sudhir Aggarwal, Michael Collins, and Henry Stern. "Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords." In Proceedings of the 17th ACM Conference on Computer and Communications Security, 162–175. CCS '10. New York, NY, USA: ACM, 2010. doi:10.1145/1866307.1866327.

[PrivacyAct] *Privacy Act of 1974* (P.L. 93-579), December 1974, available at: https://www.justice.gov/opcl/privacy-act-1974.

[PSL] *Public Suffix List* https://publicsuffix.org/list/

[Scrypt] Percival, C. and S. Josefsson, *The scrypt Password-Based Key Derivation Function*, RFC 7914, DOI 10.17487/RFC7914, August 2016, https://www.rfc-editor.org/info/rfc7914.

[Section508] Section 508 Law and Related Laws and Policies (January 30, 2017), available at: https://www.section508.gov/manage/laws-and-policies/.

[Shannon] Shannon, Claude E. "A Mathematical Theory of Communication," *Bell System Technical Journal*, v. 27, pp. 379-423, 623-656, July, October, 1948.

[Strength] Kelley, Patrick Gage, Saranga Komanduri, Michelle L Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. "Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms." In Security and Privacy (SP), 2012 IEEE Symposium On, 523–537. IEEE, 2012. Available at: https://ieeexplore.ieee.org/iel5/6233637/6234400/06234434.pdf.

[TOTP] M'Raihi, D., Machani, S., Pei, M., and J. Rydell, *TOTP: Time-Based One-Time Password Algorithm*, RFC 6238, DOI 10.17487/RFC6238, May 2011, https://www.rfc-editor.org/info/rfc6238.

Standards

[ISO/IEC9241-11] International Standards Organization, ISO/IEC 9241-11 *Ergonomic requirements for office work with visual display terminals (VDTs) — Part 11: Guidance on usability*, March 1998, available at: https://www.iso.org/standard/16883.html.

[ISO/IEC2382-37] International Standards Organization, *Information technology — Vocabulary — Part 37: Biometrics*, 2017, available at: https://standards.iso.org/ittf/PubliclyAvailableStandards/c066693 ISO IEC 2382-37 2017.zip.

[ISO/IEC10646] International Standards Organization, *Information technology — Universal coded character set (UCS)*, 2020, available at: https://www.iso.org/standard/76835.html.

[ISO/IEC24745] International Standards Organization, *Information technology — Security techniques — Biometric information protection*, 2011, available at: https://www.iso.org/iso/iso catalogue/catalogue tc/catalogue detail.htm?csnumber=52946.

[ISO/IEC30107-1] International Standards Organization, *Information technology — Biometric presentation attack detection — Part 1: Framework*, 2016, available at: https://standards.iso.org/ittf/PubliclyAvailableStandards/c053227 ISO IEC 30107-1 2016.zip.

[ISO/IEC30107-3] International Standards Organization, *Information technology — Biometric presentation attack detection — Part 3: Testing and reporting*, 2017.

[RFC20] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, https://www.rfc-editor.org/info/rfc20.

[UAX15] Unicode Consortium, *Unicode Normalization Forms*, Unicode Standard Annex 15, Version 9.0.0, February 2016, available at: https://www.unicode.org/reports/tr15/.

NIST Special Publications

NIST 800 Series Special Publications are available at https://csrc.nist.gov/publications/sp800. The following publications may be of particular interest to those implementing systems of applications requiring digital authentication.

[SP800-38B] NIST Special Publication 800-38B, *Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication*, October, 2016, https://dx.doi.org/10.6028/NIST.SP.800-38B.

[SP800-53] NIST Special Publication 800-53 Revision 5, *Security and Privacy Controls for Information Systems and Organizations*, September 2020 (updated December 10, 2020), https://dx.doi.org/10.6028/NIST.SP.800-53r5.

[SP800-63] NIST Special Publication 800-63-4, *Digital Identity Guidelines*, December 2022, https://doi.org/10.6028/NIST.SP.800-63-4.ipd.

[SP800-63A] NIST Special Publication 800-63B-4, *Digital Identity Guidelines:*

Enrollment and Identity Proofing, December 2022, https://doi.org/10.6028/NIST.SP.800-63a-4.ipd.

[SP800-63C] NIST Special Publication 800-63C-4, *Digital Identity Guidelines: Assertions and Federation*, November 2022, https://doi.org/10.6028/NIST.SP.800-63c-4.ipd.

[SP800-73] NIST Special Publication 800-73-4, *Interfaces for Personal Identity Verification*, February 2016, https://doi.org/10.6028/NIST.SP.800-73-4.

[SP800-90A] NIST Special Publication 800-90A Revision 1, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, June 2015, https://dx.doi.org/10.6028/NIST.SP.800-90Ar1.

[SP800-107] NIST Special Publication 800-107 Revision 1, *Recommendation for Applications Using Approved Hash Algorithms*, August 2012, https://dx.doi.org/10.6028/NIST.SP.800-107r1.

[SP800-131A] NIST Special Publication 800-131A Revision 2, *Transitioning the Use of Cryptographic Algorithms and Key Lengths*, March 2019, https://dx.doi.org/10.6028/NIST.SP.800-131Ar2

[SP800-132] NIST Special Publication 800-132, *Recommendation for Password-Based Key Derivation*, December 2010, https://dx.doi.org/10.6028/NIST.SP.800-132.

[SP800-185] NIST Special Publication 800-185, *SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash, and ParallelHash*, December 2016, https://doi.org/10.6028/NIST.SP. 800-185.

Federal Information Processing Standards

[FIPS140] Federal Information Processing Standard Publication 140-3, *Security Requirements for Cryptographic Modules*, March 22, 2019, https://doi.org/10.6028/NIST. FIPS.140-3.

[FIPS198] Federal Information Processing Standard Publication 198-1, *The Keyed-Hash Message Authentication Code (HMAC)*, July 2008, https://doi.org/10.6028/NIST.FIPS.198-1.

[FIPS201] Federal Information Processing Standard Publication 201-3, *Personal Identity Verification (PIV) of Federal Employees and Contractors*, January 2022, https: //dx.doi.org/10.6028/NIST.FIPS.201-3.

[FIPS202] Federal Information Processing Standard Publication 202, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, August 2015, https://dx.doi.org/10.6028/NIST.FIPS.202.

Appendix A. Strength of Memorized Secrets

This appendix is informative.

Throughout this appendix, the word "password" is used for ease of discussion. Where used, it should be interpreted to include passphrases and PINs as well as passwords.

A.1. Introduction

Despite widespread frustration with the use of passwords from both a usability and security standpoint, they remain a very widely used form of authentication [Persistence]. Humans, however, have only a limited ability to memorize complex, arbitrary secrets, so they often choose passwords that can be easily guessed. To address the resultant security concerns, online services have introduced rules in an effort to increase the complexity of these memorized secrets. The most notable form of these is composition rules, which require the user to choose passwords constructed using a mix of character types, such as at least one digit, uppercase letter, and symbol. However, analyses of breached password databases reveal that the benefit of such rules is not nearly as significant as initially thought [Policies], although the impact on usability and memorability is severe.

Complexity of user-chosen passwords has often been characterized using the information theory concept of entropy [Shannon]. While entropy can be readily calculated for data having deterministic distribution functions, estimating the entropy for user-chosen passwords is difficult and past efforts to do so have not been particularly accurate. For this reason, a different and somewhat simpler approach, based primarily on password length, is presented herein.

Many attacks associated with the use of passwords are not affected by password complexity and length. Keystroke logging, phishing, and social engineering attacks are equally effective on lengthy, complex passwords as simple ones. These attacks are outside the scope of this Appendix.

A.2. Length

Password length has been found to be a primary factor in characterizing password strength [Strength] [Composition]. Passwords that are too short yield to brute force attacks as well as to dictionary attacks using words and commonly chosen passwords.

The minimum password length that should be required depends to a large extent on the threat model being addressed. Online attacks where the attacker attempts to log in by guessing the password can be mitigated by limiting the rate of login attempts permitted. In order to prevent an attacker (or a persistent claimant with poor typing skills) from easily inflicting a denial-of-service attack on the subscriber by making many incorrect guesses, passwords need to be complex enough that rate limiting does not occur after a modest number of erroneous attempts, but does occur before there is a significant chance of a successful guess.

Offline attacks are sometimes possible when one or more hashed passwords is obtained by the attacker through a database breach. The ability of the attacker to determine one or more users' passwords depends on the way in which the password is stored. Commonly, passwords are salted with a random value and hashed, preferably using a computationally expensive algorithm. Even with such measures, the current ability of attackers to compute many billions of hashes per second with no rate limiting requires passwords intended to resist such attacks to be orders of magnitude more complex than those that are expected to resist only online attacks.

Users should be encouraged to make their passwords as lengthy as they want, within reason. Since the size of a hashed password is independent of its length, there is no reason not to permit the use of lengthy passwords (or pass phrases) if the user wishes. Extremely long passwords (perhaps megabytes in length) could conceivably require excessive processing time to hash, so it is reasonable to have some limit.

A.3. Complexity

As noted above, composition rules are commonly used in an attempt to increase the difficulty of guessing user-chosen passwords. Research has shown, however, that users respond in very predictable ways to the requirements imposed by composition rules [Policies]. For example, a user that might have chosen "password" as their password would be relatively likely to choose "Password1" if required to include an uppercase letter and a number, or "Password1!" if a symbol is also required.

Users also express frustration when attempts to create complex passwords are rejected by online services. Many services reject passwords with spaces and various special characters. In some cases, the special characters that are not accepted might be an effort to avoid attacks like SQL injection that depend on those characters. But a properly hashed password would not be sent intact to a database in any case, so such precautions are unnecessary. Users should also be able to include space characters to allow the use of phrases. Spaces themselves, however, add little to the complexity of passwords and may introduce usability issues (e.g., the undetected use of two spaces rather than one), so it may be beneficial to remove repeated spaces in typed passwords prior to verification.

Users' password choices are very predictable, so attackers are likely to guess passwords that have been successful in the past. These include dictionary words and passwords from previous breaches, such as the "Password1!" example above. For this reason, it is recommended that passwords chosen by users be compared against a blocklist of unacceptable passwords. This list should include passwords from previous breach corpuses, dictionary words, and specific words (such as the name of the service itself) that users are likely to choose. Since user choice of passwords will also be governed by a minimum length requirement, this dictionary need only include entries meeting that requirement. As noted in Sec. 5.1.1.2, it is not beneficial for the blocklist to be excessively large or comprehensive, since its primary purpose is to prevent the use of very common passwords that might be guessed in an online attack before throttling restrictions take effect. An excessively large blocklist is likely to frustrate users that attempt to choose a memorable password.

Highly complex memorized secrets introduce a new potential vulnerability: they are less likely to be memorable, and it is more likely that they will be written down or stored electronically in an unsafe manner. While these practices are not necessarily vulnerable, statistically some methods of recording such secrets will be. This is an additional motivation not to require excessively long or complex memorized secrets.

A.4. Central vs. Local Verification

While passwords that are used as a separate authentication factor are generally verified centrally by the CSP's verifier, those that are used as an activation factor for a multi-factor authenticator are either verified locally or are used to derive the authenticator output, which will be incorrect if the wrong activation factor is used. Both of these situations are referred to as "local verification".

The attack surface and vulnerabilities for central and local verification are very different from each other. Accordingly, the requirements for memorized secrets verified centrally is different from those verified locally. Centrally verified secrets require the verifier, which is an online resource, to store salted and iteratively hashed verification secrets for all subscribers' passwords. Although the salting and hashing process increases the computational effort to determine the passwords from the hashes, the verifier is an attractive target for attackers, particularly those who are interested in compromising an arbitrary subscriber rather than a specific one.

Local verifiers do not have the same concerns with attacks at scale on a central online verifier, but depend to a greater extent on the physical security of the authenticator and the integrity of its associated endpoint. To the extent that the authenticator stores the activation factor, that factor must be protected against physical and side-channel (e.g., power and timing analysis) attacks on the authenticator. When the activation factor is entered through the associated endpoint, the endpoint needs to be free of malware, such as key-logging software, if the password is to be protected. Since these threats are less dependant on the length and complexity of the password, those requirements are relaxed for local verification.

Online password-guessing attacks are a similar threat for centrally and locally verified passwords. Throttling, which is the primary defense against online attacks, can be particularly challenging for local verifiers because of the limited ability of some authenticators to securely store information about unsuccessful attempts. Throttling can be performed by either keeping a count of invalid attempts in the authenticator, or by generating an authenticator output that is rejected by the CSP verifier, which does the throttling. In this case it is important that the invalid outputs not be obvious to the attacker, who could otherwise make offline attempts until a valid-looking output appears.

A.5. Summary

Length and complexity requirements beyond those recommended here significantly increase the difficulty of memorized secrets and increase user frustration. As a result, users often work around these restrictions in a way that is counterproductive. Furthermore, other mitigations such as blocklists, secure hashed storage, and rate limiting are more effective at preventing modern brute-force attacks. Therefore, no additional complexity requirements are imposed.

Appendix B. Change Log

This appendix is informative. It provides an overview of the changes to SP 800-63B since its initial release.

- Section 5.2.3 — Updated biometric performance requirements and metrics and included discussion of equity impacts.
- Section 5.2.5 — Added definition and updated requirements for phishing resistant authenticators.
- Section 5.2.11 — Established separate requirements for locally verified memorized secrets known as *activation secrets*.
- Section 5.2.12 — Added requirements for authenticators that are connected via wireless technologies such as NFC and Bluetooth.