NIST Interagency Report
NIST IR 8587 ipd

Protecting Tokens and Assertions from Forgery, Theft, and Misuse

Implementation Recommendations for Agencies and Cloud Service Providers

Initial Public Draft

Ryan Galluzzo
Andrew Regenscheid

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8587.ipd

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

NIST Interagency Report
NIST IR 8587 ipd

Protecting Tokens and Assertions from Forgery, Theft, and Misuse

Implementation Recommendations for Agencies and Cloud Service Providers

Initial Public Draft

Andrew Regenscheid
Computer Security Division
Information Technology Laboratory

Ryan Galluzzo
Applied Cybersecurity Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8587.ipd

December 2025

Published by:

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

Report written with:

NIST Technical Series Policies

Publication History

Approved by the NIST Editorial Review Board on YYYY-MM-DD
Supersedes NIST Series XXX (Month Year) DOI

How to Cite this NIST Technical Series Publication

Galluzzo R, Regenscheid A (2025) Protecting Tokens and Assertions from Forgery, Theft, and Misuse: Implementation Recommendations for Agencies and Cloud Service Providers. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency Report (IR) NIST IR 8587 ipd. https://doi.org/10.6028/NIST.IR.8587.ipd

Author ORCID iDs

Andrew Regenscheid: 0000-0002-3930-527X
Ryan Galluzzo: 0000-0003-0304-4239

Public Comment Period

December 22, 2025 – January 30, 2026

Submit Comments

iam@list.nist.gov
National Institute of Standards and Technology
Attn: Applied Cybersecurity Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 2000) Gaithersburg, MD 20899-2000

Additional Information

Additional information about this publication is available at https://csrc.nist.gov/pubs/ir/8587/ipd, including related content, potential updates, and document history.

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

Abstract

This report provides implementation guidance to help federal agencies and cloud service providers (CSPs) protect identity tokens and assertions from forgery, theft, and misuse. Building on updates to NIST SP 800-53 (Release 5.1.1), it outlines principles for CSPs and consuming agencies, details architectural considerations for identity providers and authorization servers, and recommends enhancements to key management, token verification, and life cycle controls. The report addresses threats demonstrated in recent high-profile attacks, emphasizes the importance of secure by design practices, configurability, interoperability, and continuous monitoring, and provides specific technical recommendations to safeguard single sign-on, federation, and application programming interface (API) access scenarios.

Keywords

access management; federation; key management; single sign-on.

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.

Note to Reviewers

As an initial public draft, this document intends to gain critical feedback from stakeholders across government and industry. Comments are welcome on all aspects of this document and specifically encouraged on the following areas:

1. **Signing Key Validity Periods.** This document provides recommendations for signing key validity periods used in conjunction with tokens and assertions. They are structured around key usage scenarios (e.g., whether the key is used for more than one tenant). NIST is interested in feedback on the length of validity periods, the structure of the scenarios, and any elements.
2. **Token Validity Periods.** This document sets a baseline validity period for tokens and assertions and allows for flexibility based on the availability of certain capabilities (e.g., revocation, compromise detection). NIST is interested in comments on token validity lengths and compensating controls that may impact them, particularly their availability, adoption, and use in government systems.
3. **Key Protection and Isolation.** NIST is interested in feedback on the clarity and suitability of key management definitions and whether they are appropriately mapped to Federal Information Security Modernization Act (FISMA) system categorization levels.
4. **Scoping.** This document recommends limiting the scope of trust in signing keys and requiring tokens to include explicit audience and scope restrictions to limit the impact of key compromise. NIST is interested in feedback on operational considerations, implementation challenges, and best practices that could strengthen these recommendations.
5. **Emerging Standards and Protocols.** This document references several new or emerging standards and protocols. NIST is particularly interested in the availability, maturity, and adoption rates of products that utilize these standards and protocols. Similarly, NIST is seeking input on additional standards and protocols that were not mentioned but could contribute to achieving security outcomes.

Reviewers can submit comments—including responses to the Note to Reviewer areas highlighted above—to iam@list.nist.gov throughout the public comment period.

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: iam@list.nist.gov

Table of Contents

List of Tables

List of Figures

Acknowledgments

The authors wish to acknowledge the collaboration between NIST, CISA, and NSA to make this document possible. We would also like to acknowledge those subject-matter experts who participated in the Joint Cyber Defense Collaborative (JCDC) Cloud Security Workshop in June of 2025 and provided invaluable insights that informed this document.

1. Introduction

Recent cybersecurity incidents at major cloud service providers (CSPs) have focused on the ability to steal, modify, or forge identity tokens and assertions used by enterprise single sign-on (SSO) and identity federation systems to gain access to sensitive applications, data, and communications. In response to this critical and emerging threat, NIST issued release 5.1.1 to Special Publication (SP) 800-53, *Security and Privacy Controls for Information Systems and Organizations* [1]. Released in November 2023,[1] the patch provided additional control and supporting control enhancements related to identity providers (IdPs), authorization servers, the protection of cryptographic keys, the verification of identity assertions and access tokens, and token management [2].

With the proliferation of cloud computing services and the distribution of government data and services to external infrastructure, platforms, and software, this threat becomes even more distinct. Agencies do not always have the visibility they need into the external services they procure to identify, respond to, and remediate emerging threats. As such, while this document provides controls and considerations that expand on SP 800-53, the recommendations contained herein are equally important to cloud services – whether commercially offered to agency customers or operated by government agencies.

IMMEDIATE CALL TO ACTION:

Federal agencies must understand the architectures, designs, and deployment models of their CSPs to configure services that are consistent with their risk posture and threat environment.

When providing services to federal agencies, CSPs need to deliver security mitigations that are configurable, transparent, and interoperable to empower cloud service consumers to implement risk-informed, threat-adaptive defenses across diverse environments.

1.1. Purpose and Scope

This publication is scoped to federal environments and has been developed pursuant to the authorities detailed in 15 U.S.C. §278g–3.[2] This document expands on the new control in SP 800-53 [1] and provides technical guidance to support the implementation of token and assertion protections and cryptographic mechanisms. Specifically, this document covers controls that protect identity and access management (IAM) systems that rely on digitally signed assertions and tokens when making access decisions. This often includes SSO, federation, and API access scenarios. Systems that do not rely on signed assertions and tokens

[1] Release 5.2.0 to SP 800-53 was subsequently published in August 2025.
[2] 15 U.S.C. §278g–3 directs NIST to develop standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets; and for information systems used or operated by an agency or by a contractor of an agency or other organization on behalf of an agency, other than national security systems.

(e.g., stateful management, API request authentication) are mentioned where relevant but out of scope for the controls covered.

1.2 Notations

This guideline uses the below typographical conventions in text to indicate the parameters for those entities that claim conformance. However, conformance with this guideline, remains voluntary for CSPs and federal agencies unless otherwise determined through policy, including OMB policies, or other binding agreements such as contracts or grants.

- Specific terms in **CAPITALS** represent normative requirements. When these same terms are not in **CAPITALS**, the term does not represent a normative requirement.
    - The terms “**MUST**” and “**MUST NOT**” indicate requirements to be followed in order to conform to the guidelines.
    - The terms “**SHOULD**” and “**SHOULD NOT**” indicate that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited.

1.3 Identity Management in Cloud and Hybrid Environments — Managing Responsibilities

In cloud environments, security and identity management are governed by a shared responsibility model that must clarify which aspects of identity and credential hardening are managed by the CSP and which are the responsibility of the cloud consumer (i.e., organization or end user). Understanding and implementing this division is essential for maintaining robust security and preventing gaps that adversaries could exploit.

CSP responsibilities often include securing the underlying infrastructure, maintaining core IAM services, managing token issuance and signing, providing secure secrets storage, ensuring infrastructure-level logging and monitoring, and maintaining compliance with regulatory frameworks. CSPs provide the configurable security controls and tools necessary for consumers to build secure applications and services.

Cloud consumer responsibilities often include securely configuring IAM policies, managing application-level secrets and credentials, enforcing strong authentication and access controls, conducting operational security activities (e.g., deployment via DevSecOps pipelines, access reviews, red team exercises), and responding to incidents. Consumers are also responsible for educating their users and teams and for maintaining logs and notification mechanisms.

Consumer configuration choices are driven by risk assessments of the cloud-hosted service’s sensitivity and the impacts of a loss of confidentiality, integrity, and/or availability. In nearly all cases and deployment scenarios, the CSP and the consumer will need to coordinate on logging, monitoring, incident response, and compliance reporting to ensure that incidents are managed effectively.

It is important for CSPs and consuming agencies to understand their respective roles and responsibilities for managing IAM controls to support effective collaboration and comprehensive security in cloud environments. Table 1 provides an example of how these responsibilities may be delineated in software-as-a-service offerings.

Table 1. Example responsibility areas

Responsibility Area	CSP	Consumer
Physical Security	●	
Infrastructure (Hardware/Networking)	●	
Core IAM Services	●	
Token Issuance/Signing	●	
Secrets Vaults/Hardware Security Module	●	
Infrastructure Logging/Monitoring	●	
IAM Policy Configuration		●
Application/User Access Control		●
Application Secrets Management		●
Multifactor Authentication (MFA)/User Authentication Setup		●
Session Management		●
Application Logging/Monitoring		●
User Education	●	●
Incident Response	●	●
Continuous Monitoring	●	●

1.4 Principles for Cloud Service Providers

The following principles are necessary to effectively support a secure relationship between consuming agencies and CSPs:

- **Secure development and design.** CSP systems are an integral component of agency services and increasingly essential to modern federal agency missions and service delivery. Not every CSP is a security company, but they still need to build security into the systems they design and deploy. Their practices should be consistent with CISA's Secure by Design principles [3], NIST's Secure Software Development Framework [4], and NIST's Engineering Trustworthy Secure Systems guidelines [5].
- **Transparency.** Effective risk management on both sides of the CSP-to-consumer relationship requires the CSP to be transparent about their deployed technology, the architecture underpinning the technology, and system-generated data. Particularly critical is the ability of the CSP to convey alignment with consumer security requirements (to include IA-13: Identity Providers and Authorization Servers from NIST

SP 800-53, Rev. 5.1.1) and establish effective communication channels to support logging and analyzing token- and assertion-related events.

- **Configurability.** Different cloud service models (e.g., infrastructure as a service, platform as a service, software as a service) have different capabilities, and security features need to be selectable and tunable by consumers to the greatest degree practical. This allows organizations to apply mitigations that match their risk tolerance and operational needs. Critical consensus and compliance-driven mitigations (e.g., those defined in this document and SP 800-53 more generally) should be available as defaults, with additional options for organizations with advanced requirements or specific configuration needs. Appropriate training, documentation, and support capabilities must accompany these configurations.
- **Interoperability.** Providing essential support for standards-based architectures allows seamless integration and management across hybrid and multi-cloud environments, enabling consumers to support redundancy and resiliency. Standards-based deployments reduce complexity introduced by bespoke integrations and facilitate consistent security postures across the consumer's entire enterprise. This facilitates interoperability between and amongst connected components without mandating or constraining a single approach to token or assertion management.

Implementing these principles promotes a healthy and secure relationship between the CSP and its consumers.

1.5 Principles for Consuming Agencies

Agencies that consume cloud services to support mission and service delivery have their own responsibilities with respect to secure implementation. Agencies must ensure that procured cloud environments and services satisfy FISMA security expectations for transaction risk and sensitivity. To effectively implement solutions that meet these responsibilities, agencies must apply the following principles to their engagement process with CSPs:

- **Risk assessment and control selection.** While CSPs may offer baseline solutions that align with FISMA categorization and FedRAMP impact levels, the consuming agency is responsible for completing a thorough risk assessment [6] in accordance with the federal Risk Management Framework (RMF) [7]. Agencies are also responsible for conducting a digital identity risk assessment (DIRA), as defined in SP 800-63-4, *Digital Identity Guidelines* [13]. This process helps define a baseline set of security controls, security outcomes, and assurance levels for any CSP-offered or in-house-developed token or assertion-based system.
- **Tailoring.** Control baselines serve as a starting point for selecting controls. Both the RMF and *Digital Identity Guidelines* enable agencies to further refine, select, and implement controls that are specific to their operational and threat environment and subsequently communicate these requirements to CSPs. This may also be informed by the availability of controls within selected CSP or on-premises environments.

- **Secure integration and configuration.** As indicated above, CSP services should be configurable. The consuming agency is responsible for configuring their cloud environments to meet their security categorization, control selection, tailoring outcomes, and defined assurance levels in coordination with their CSPs.

While cloud-hosted services introduce complexity to the risk management process, they also enable substantial mission and business capabilities that cannot be replicated with organic agency systems and environments. By acknowledging the shared responsibility for data protection and implementing these principles, both CSPs and agencies can more effectively manage and address the threats that they face while improving the way the government delivers digital capabilities to employees and members of the public. In particular, these processes are essential to supporting remote access scenarios, SSO, and the provisioning of modern API-based services.

1.6 Continuous Monitoring and Evaluation

No security controls — particularly identity management controls — should be “fire and forget.” Consistent with the RMF, both CSPs and consuming agencies are expected to implement continuous monitoring capabilities for the architectural components that they control. Consuming agencies coordinate with CSPs to continuously evaluate and monitor CSP environments, connections to on-premises or multi-cloud environments, and access events at all levels to identify potential vulnerabilities or incidents. In many real-world scenarios associated with token and assertion compromise, the only mechanisms for detection were the monitoring and analysis of account activity and the scrutiny of log data from CSP environments and services.

2. Overview of Assertions and Tokens

In federated and SSO environments, information about identity is often conveyed through either an identity assertion or an identity token. These statements contain authentication information and attributes that can be used by protected resources (i.e., relying parties [RPs]) to make access control decisions to determine whether a user or software is appropriately authenticated and authorized to access a specific application, service, or data. These assertions or identity tokens are digitally signed by the issuer (i.e., IdP) to provide the RPs with confidence that they came from a trusted source and that the data contained in the assertion or token has not been modified. When used correctly, these objects allow an enterprise to grant users access without having to constantly authenticate the user prior to accessing services – improving usability and efficiency for workforce tasks. These systems also provide substantial technical benefits, including the ability to centrally manage access control and authentication policies and provide a more consistent security posture across their environment.

However, the security of these systems relies on the enterprise's ability to safeguard the cryptographic keys used to sign assertions and issue identity tokens, as well as to appropriately verify those assertions and tokens when they are used to access RP applications. When keys are compromised (e.g., exfiltrated, copied) or verification is not done correctly, assertions and tokens can be used to enable unauthorized access and lateral movement through an organization, rapidly compromising multiple protected resources. The result of a failure of key management practices or verification is tantamount to giving attackers the "keys to the kingdom."

Several major public incidents have exploited weaknesses in assertion and token practices. For example, the SolarWinds compromise discovered in December 2020 was a complex cyber incident that involved numerous tactics, techniques, and procedures (TTPs) to compromise thousands of organizations — including federal agencies — that were using third-party security service provider SolarWinds. Once malware was delivered through a supply chain compromise, the attackers accessed protected resources by generating forged Security Assertion Markup Language (SAML) [8] assertions to bypass multifactor authentication (MFA) and other application-level protections. The attackers also compromised administrator accounts with privileged access to Active Directory Federation Services, exposed signing certificates, and minted valid but forged assertions to conduct a coordinated espionage campaign on high-value emails and other sources of intelligence [9].

In another SSO and federation attack, foreign actors accessed multiple agency email systems using forged tokens and assertions. Rather than focusing on elevating the privileges of administrator accounts to generate forged assertions, the attackers used a stolen commercial signing key that was inadvertently exposed. The compromised key should only have been valid for tokens issued in the affected vendors' commercial environments, not their enterprise or government systems. However, due to token validation failures, this stolen key was able to generate signed tokens that were used to access email servers and personal email accounts for high-ranking enterprise and federal officials, with over 60,000 emails being exfiltrated from a single agency [10].

Despite this, assertions and tokens remain critical components of a modern, efficient, and secure enterprise. When managed consistently with leading practices, they can enable secure access to enterprise services and data in a manner that supports the scale and distribution of critical modern IT infrastructures. This document provides additional considerations for agencies and CSPs to further improve and secure systems that rely on these access management constructs.

2.1. Terms and Concepts

Assertions and tokens can be used in several different architectural models for various purposes. The following sections provide a brief overview of their usage, features, and core concepts. However, implementations and deployment patterns will vary based on the enterprise that is deploying them, their objectives and outcomes, and the types of systems and access they need to support. In general, assertions and token-based schemes include the following elements,[3] regardless of what protocol or technology is being used:

- **End user.** The individual or entity seeking access to a protected resource. This end user may be a human or non-human user, depending on the use case.
- **Clients.** An application, browser, or other software that operates on behalf of the end user to request access to a specific resource.
- **Protected resources and resource servers.** Protected resources are the data, services, or applications that a client acting on behalf of a user is attempting to access. The resource server hosts the protected resource and is responsible for validating any identity or access tokens issued by the authorization server. Collectively, the protected resource and the resource server are referred to as the RP in federation or SSO scenarios.[4]
- **Authorization servers.** Authorization servers manage the authentication of the end user and generate or issue tokens with authentication data and user or client attributes. In SSO and federation scenarios, authorization servers are referred to as IdPs. They may also act as a token service that manages the exchange of identity tokens for access tokens or refreshes tokens for new access tokens.[5]

In different scenarios, these terms are aligned with different entities. For example, in SSO and federation, the IdP, RP, and access management system deploy these concepts to achieve the outcomes of the specific scenario in which tokens and assertions are being used. The subsequent sections of this document map these core concepts to specific roles in each scenario.

The standards and protocols used for tokens and assertions do not define specific implementation patterns. Similarly, this document will focus on controls and outcomes that **MUST** be achieved by implementations. These basic components may be distributed to different

[3] Different protocols and standards use slightly different terms for the different components. For the purposes of this document, these terms have been generalized to avoid using a single standard or protocol's specific taxonomy.
[4] This is known as a "client" in OAuth and a "service provider" in SAML.
[5] This is known as the "OpenID Provider" in OpenID Connect (OIDC).

physical or logical components in an architecture, reside in different domains or security boundaries (e.g., in federation scenarios), or be augmented by additional capabilities, such as fine-grained access control and authorization services used to layer additional decision-making capabilities into access determinations in front of the protected resource.

2.2. Types of Assertions and Tokens

Assertion and token-based access models shift authentication away from the protected resource. In place of direct authentication, they rely on the trusted nature of the assertion or token to make informed access decisions. Once a user or service is authenticated, tokens and assertions can be issued and validated to achieve several outcomes, including SSO, federation, and API access to protected resources. Three different types of tokens or assertions are typically used to achieve this:

1. **Identity tokens and assertions.** Identity tokens and assertions provide authentication and identity attributes related to a user. Protected resources evaluate these to make access decisions, and authorization servers can use them to issue access tokens for specific protected resources. They are typically used to support federation and SSO systems.
2. **Access tokens.** Access tokens contain authentication information and a limited set of attributes (e.g., unique user ID) to provide access to specific protected resources, applications, or services. They may be used on their own or in conjunction with identity tokens. They are also often used to support non-human access (e.g., to protect server-to-server communications or access to protected APIs).
3. **Refresh tokens.** Refresh tokens can be used with identity tokens or access tokens to enable extended sessions. They are exchanged by user agents or clients to receive new identity tokens or access tokens at the end of validity periods.

These different types of tokens are often used in conjunction during actual implementations. For example, in an enterprise SSO scenario, an end user may authenticate to an enterprise identity management service, be issued an identity token, and use or exchange that token for resource-specific access tokens. Here again, it is worth noting that each vendor and ecosystem will have variations in how tokens and assertions are used, including the “hierarchy” of those tokens.

2.3. Uses of Tokens and Assertions

There are two primary uses for signed tokens and assertions: SSO/federation and API access. These may or may not be used in conjunction with each other. For example, a user or developer first authenticates to an IdP and is issued an identity token that is then used to get access tokens used to make authenticated API calls from their machine or system.

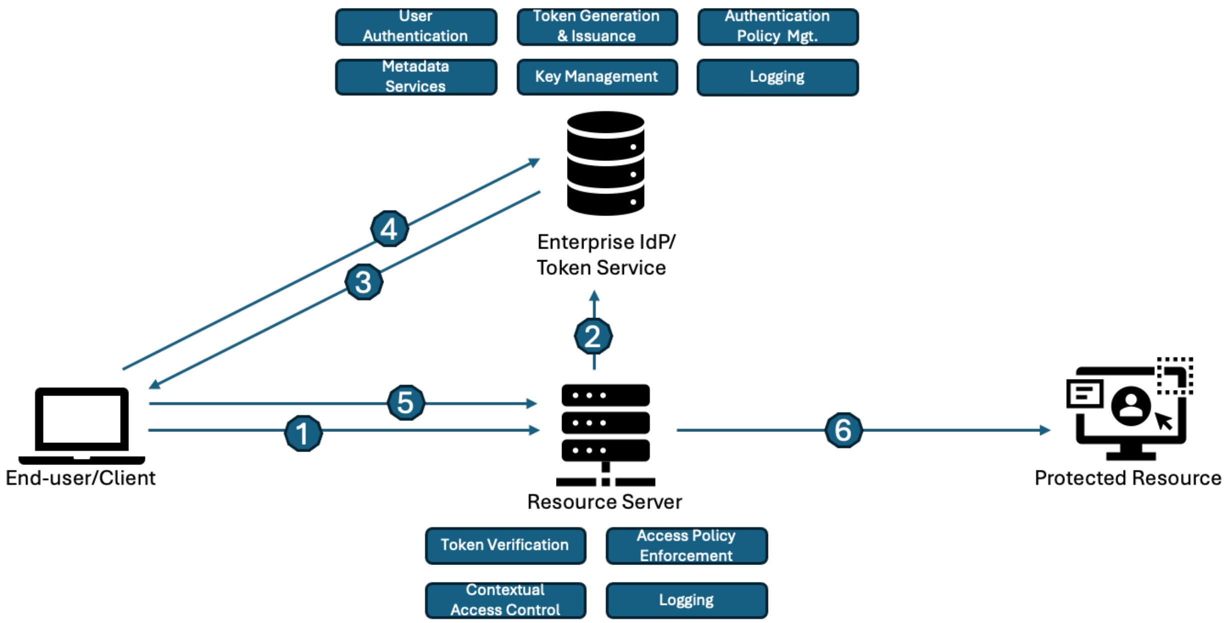

Fig. 1. Architecture of token-based systems

Figure 1 illustrates a typical process flow for a basic token architecture and its components. While the precise order and capabilities related to an assertion or token management process may vary, they generally follow a flow that includes:

1. The end user requests access (via a client) to protected resources through the resource server.
2. The resource server redirects the end user's request to the enterprise IdP or token service to authenticate the end user.
3. The enterprise IdP/token service evaluates the authentication policy for the protected resource and requests end-user authentication.
4. The end user authenticates to the enterprise IdP/token service, which issues an identity token, assertion, or access token to the end user or client.
5. The end user's client communicates the token/assertion to the resource server, which verifies the token and evaluates the access policy.
6. The resource server grants access to the protected resource.

In the context of non-human interactions, the methods of authentication may not interact with an end user but rather employ device or machine authentication mechanisms.

2.3.1. Single Sign-On and Federation

There are two primary methods for managing tokens in SSO and federation systems: stateless management and stateful management. Both methods grant a user or service access to a protected resource without having to authenticate them at the resource level for each access event. This is achieved by different means in each architecture. While signed tokens and assertions are a feature of stateless systems, many enterprises employ both models or even hybrid systems.

Table 2. Stateful and stateless architectures

Architecture	Description	Features
Stateful	The IdP maintains a server where applications can query the current state of each subscriber or system seeking access. This allows each application to confirm the authentication state of each user through a passed session identifier. The server maintains all necessary session information for each active authentication event, including the session status, authentication times, and revocation information. Session information is passed through session headers, cookies, or unsigned tokens (e.g., JavaScript Object Notation [JSON] Web Token [JWT]).	Typically considered more secure since the session information is actively maintained centrally and can be revoked as needed. Historically seen as less viable for large decentralized systems due to the performance of centralized session management. Also seen as less viable for API-based access scenarios. Not typically viable for cross-domain access or federation scenarios.
Stateless	All of the necessary information is provided to the RP to manage access decisions and the identity of the user requesting access. There is no “phoning home” to a central server to manage session data. Instead, it is all encapsulated in an assertion or token used by the RP to make access decisions. This data is typically represented by a signed SAML assertion or OpenID Connect (OIDC) JWT. Sessions are managed through validity times and refresh tokens that allow the session to be extended by requesting an updated token from IdP servers.	Typically seen as more effective for managing highly scaled and available services, particularly those dealing with large volumes of authenticated API calls. They also provide for cross-domain and federation capabilities. Can be implemented securely, though they sacrifice some security compared to stateful systems due to the limited revocability of assertions and tokens.

The controls and mitigations discussed in this document are ***specifically scoped to protecting stateless systems that rely on signed assertions and tokens***. However, when making determinations about SSO, federation and IAM implementations, CSPs and agencies need to evaluate which architecture works best for their scenario and risk environment.

2.3.1.1. Single Sign-On and Federation Features

SSO and federation revolve around a common set of roles and components to achieve desired outcomes. The following components are typical in an SSO or federation system (see Fig. 1):

- **Subscriber and subscriber account (end user and client).** The subscriber account represents the user or service that is attempting to gain access. It is typically maintained by an authoritative enterprise source, which stores the necessary attributes, entitlements, and authorizations required for applications to make access control decisions.

- **Identity provider (authorization server).** The IdP authenticates the user or service that is seeking to gain access to a protected resource. The IdP then issues an assertion, token, or other mechanism that conveys the authentication event, attributes, and other information needed for the protected resource to make an access decision.
- **Token service (authorization server).** The token service acts as an intermediary when identity tokens or assertions need to be exchanged for access tokens or when access tokens need to be refreshed to extend a session. Depending on the architecture or implementation of the SSO system, this may be a component of the IdP or operated independently of it.
- **Relying party (protected resource and resource servers).** The RP uses information provided by the IdP and made available through an assertion, token, or other mechanism to make access control decisions for protected resources. In most environments, the resource server is integrated with other capabilities to support more robust access policies, such as fine-grained and conditional access.

2.3.2. API Access Scenarios

In addition to their use in SSO and federation scenarios, access tokens are commonly used to support API access from applications or services. In practice, this looks similar to SSO use cases in which the user (i.e., the individual accessing the mobile application) is authenticated locally or with an enterprise IdP, and subsequent calls by the application to a hosted API service are authorized using signed access tokens.

2.3.2.1. API Access Features

Typical API access scenarios include:

- **Client.** An application that makes access requests on behalf of the end user. In some scenarios, this may also be a server or other software that makes access requests on behalf of an organization or other non-human entity.
- **Authorization server.** The authorization server verifies that the client – and potentially the user it represents – has been authenticated and then issues access tokens to authorize requests to protected API resources. Where this server resides in the organization’s architecture depends on the deployment model used, but the authorization server will often be integrated with other IAM systems and tools to enforce appropriate authentication and conditional access decisions before issuing valid tokens.
- **Protected resource and resource servers.** The protected resource (e.g., enterprise APIs or services) makes access decisions based on the information provided in access tokens.

In most environments, the resource server is integrated with other capabilities to support more robust access policies, such as fine-grained and conditional access.

Unlike web-based SSO scenarios, users will likely have much longer access windows to avoid constant authentication and password or MFA entry. In many instances, API access scenarios do not involve human users or interactive models of authentication at all. For example, a designated service account may receive an access token to call a cloud-hosted API to download data for a system or device. In these cases, a traditional “re-authentication” event is not always viable, and the validity of access tokens or associated refresh tokens will need to be determined based on other contextual information, such as device registration (e.g., is the server making the request registered as a trusted component), IP constraints (e.g., is the device coming from an approved network), transactional data (e.g., is the request exhibiting expected behavior), and, potentially, proof-of-possession techniques (e.g., device-specific certificates, message authentication techniques). Like other protected resources, API access typically sits behind resource servers that can act as a policy enforcement point to support the contextual security needed for access decisions.

3. IA-13: Identity Providers and Authorization Servers

IA-13 was added to the 5.1.1 revision of the SP 800-53 control catalogue [1] in direct response to several high-profile token-related security incidents. The following sections explore the IA-13 control and its enhancements in detail and provide additional implementation recommendations. The control is expected to be applied by CSPs that serve federal government consumers, implemented by federal agencies as part of their own token and assertion-based systems, and evaluated by consumers who seek to use CSP services.

> **Control Statement:** Employ IdPs and authorization servers to manage user, device, and non-person entity (NPE) identities, attributes, and access rights supporting authentication and authorization decisions in accordance with [Assignment: organization-defined identification and authentication policy] using [Assignment: organization-defined mechanisms].
>
> **Discussion:** IdPs, both internal and external to the organization, manage the user, device, and NPE authenticators and issue statements, often called identity assertions, attesting to identities of other systems or systems components. Authorization servers create and issue access tokens to identified and authenticated users and devices that can be used to gain access to system or information resources. For example, SSO provides IdP and authorization server functions. Authenticator management (to include credential management) is covered by IA-05.

3.1. Implementation Recommendations

The addition of IA-13 is intended to provide CSPs and agencies with a baseline control to support the effective management and evaluation of identity and access tokens (generated by organizational or external IdPs and authorization servers) and the access management services that evaluate those tokens when making access decisions. The following sections discuss actions and recommendations for implementing IA-13 and achieving the control’s intended outcomes.

3.1.1. Architecture and Design

When implementing this baseline control, organizations MUST evaluate, design, and document an architecture that will meet organizational needs and requirements. In current IAM infrastructures, the volume of applications, the automation of processes, and the move toward distributed, mobile-friendly environments make the use of assertions, tokens, or other abstracted authentication techniques almost unavoidable. Critical decisions include determining whether to use a stateful or stateless architecture, deciding on protocols for assertions and tokens (e.g., SAML versus OIDC), and establishing the mechanisms and hierarchies associated with identity and access tokens within the enterprise. There is no “one-size-fits-all” answer to this question, and agencies will often need to contend with legacy systems and cloud environments, which may all have varying impacts on architectural decisions. For example, some systems may not be capable of consuming assertions or tokens and may need to rely on agents, proxies, or translation services that introduce complexity and potential vulnerabilities to an overall system design. Additional considerations for system design and architecture are discussed in Table 3. This list should not be considered exhaustive; rather, it is a starting point for key design decisions that an agency needs to make.

Table 3. Design and architecture considerations

Topic	Considerations
Stateful vs. Stateless	There are benefits and risks with each of the core architectures associated with SSO and indirect authentication, as shown in Table 2. CSPs and agencies implementing identity management systems need to assess and implement technologies that support their customers and mission. Agencies that leverage CSP infrastructure need to understand its strengths and weaknesses and configure it appropriately to maximize the security of procured systems.
Signature Approach (Symmetric vs. Asymmetric Keys)	Tokens can be signed using either symmetric or asymmetric keys. In general, most SSO and token management systems implement asymmetric keys when signing assertions and tokens. However, some systems implement signing with symmetric keys, such as those that leverage hash-based message authentication code (HMAC) with JWT and SAML assertions. Generally, symmetric signing of assertions and tokens is not recommended. It introduces complexity for key rotation, hampers scalability, and increases key protection requirements. However, such schemes can be deployed in a manner that limits the blast radius of a key exposure since proper key management requires unique keys for each pair of authorization servers and resource servers. This means that the loss of a symmetric signing key does not compromise interactions outside of the scope of those specific interactions.
Protocol Selection	Consistent with the recommendations of CISA's *Hybrid Identity Solution Guidance,* modern open standards are preferred for managing assertions and tokens [11]. OIDC [25] used in conjunction with open authorization (OAuth) [12] provides a flexible, standards-based set of protocols that can support API and mobile-based services while also supporting traditional human-centric identity systems. While SAML is a secure and proven protocol, it is more constrained in its uses and best suited to web applications due to its limited ability to support mobile or API-based systems, which are integral to most modern enterprises. Legacy considerations may result in agencies and CSPs needing to deploy a combination of these protocols to fully cover their existing protected resource base.
Environments (Cloud, On-Premises, Hybrid)	The integration of cloud environments into the modern federal enterprise comes with many opportunities but also introduces complexity and potential security risks. Most users — human and machine alike — access protected resources in multiple environments to execute mission-critical activities. As such, agencies need to deploy IAM, SSO, and federation systems that can integrate with a host of external environments. This will inevitably impact the decisions that are made for token and assertion services, particularly when they support cross-boundary access.

Topic	Considerations
Mobile and Non-Human Identities	IAM systems must be able to handle human users as well as service accounts, software, and other non-human identities that access protected resources on a user's behalf. Agencies need to keep this in mind when selecting architectures, designs, and protocols. Stateful systems and older protocols (e.g., SAML) were not intended to support these forms of access, and the often limited computing capacity of such entities makes OIDC and OAuth — lighter-weight protocols designed with mobile access scenarios in mind — better suited to environments that deal with a more diverse set of access scenarios.
Legacy Applications	Most agencies have decades of legacy systems that introduce restrictions and constraints to any IAM deployment. Agencies will need to evaluate the state of legacy applications and understand the potential constraints they impose on new architectural plans and designs. Addressing these may require the introduction of additional system components to deal with translation, proxying, or authorization decisions.

3.1.2. Risk Assessment and Risk Management

Effective identity management requires controls and assurance levels commensurate with system risk. Frameworks such as the RMF and Federal Information Processing Standards (FIPS) 199/200 guide impact categorization (i.e., Low, Moderate, High) and inform control selection, while SP 800-63 *Digital Identity Guidelines* defines assurance levels for identity proofing (IAL), authentication (AAL), and federation, as well as a Digital Identity Risk Management (DIRM) model [9].

The DIRM helps organizations identify baseline assurance levels and, with tailoring, implement xAL sequences (e.g., IAL2/AAL2/FAL1) based on their mission, threat environment, and data sensitivity. DIRM is implemented by resource owners to determine access requirements and by service providers to design service assurance that supports consumers or clients. The requirements associated with federation assurance levels (FALs) establish the necessary conditions for generating, conveying, and processing identity assertions at prescribed levels.

Federal agencies **MUST** use these frameworks to determine appropriate baseline controls and tailor them as needed. CSPs **MUST** provide the necessary configurability and interoperability to support this process and provide mappings of their control and assurance functionality to allow agencies to meet baseline requirements.

3.1.3. Security Policy and Technical Documentation

To support the practical implementation of the architecture, agencies **MUST** define and document the security policies and technical requirements for token and assertion management, including both identity and access tokens. This **SHOULD** include addressing critical issues such as application access policies, token and assertion lifetimes, token and assertion validation process, key management practices, audit/logging (and associated data), and incident response processes. The specific protocols and contents of tokens and assertions **MUST** also be documented.

3.1.4. Authorization Systems and Zero Trust Architectures

Authorization systems consume identity and access tokens to make access control decisions. They sit at the heart of enterprise security and are the foundation of zero trust architectures. As such, the authorization infrastructure needs to be robustly designed and documented to support zero trust principles and outcomes. Consistent with Office of Management and Budget (OMB) Memo 22-09, *Moving the U.S. Government Toward Zero Trust Cybersecurity Principles* [19], agencies **MUST** implement assertions and tokens in a manner that enables more granular and dynamically defined permissions (e.g., attribute-based access control), enforces an effective validation process for the signatures that protect these assertions, and integrates with additional sources of authorization data. Authorization systems **SHOULD NOT** make access decisions based solely on the validity of a token but also based on the context in which it is presented.

4. Control Enhancements

Agencies can use control enhancements to further improve their implementation of a specific control. Like the base control, these enhancements can be mapped to the FISMA categorizations to support consistent application across the federal enterprise. The following sections discuss the specific control enhancements related to IA-13 in SP 800-53 and elaborate on the control text to provide additional recommendations.

4.1. Control Enhancement 1: Protection of Cryptographic Keys

> **Enhancement.** Cryptographic keys that protect access tokens are generated, managed, and protected from disclosure and misuse.
>
> **Discussion.** Identity assertions and access tokens are typically digitally signed. The private keys used to sign these assertions and tokens are protected commensurate with the impact of the system and information resources that can be accessed.

4.1.1. Additional Guidance for the Protection of Cryptographic Keys

The protection of cryptographic keys used to sign assertions and access tokens is critical to the security of SSO and federation systems. A compromised signing key allows attackers to forge their own tokens and assertions, which grants virtually unfettered access to connected systems and vastly increases their ability to move laterally within an enterprise.

4.1.1.1. Generation

Cryptographic keys used by IdPs and authorization servers to digitally sign and encrypt assertions or tokens, authenticate software clients and applications, or establish secure channels **MUST** be generated in accordance with the guidelines in SP 800-57p1r5, *Recommendations for Key Management: Part 1 – General* [20].

Key generation is the first phase of the key management life cycle and **MUST** be performed using cryptographically secure processes that ensure adequate entropy, algorithmic strength, and application suitability. Keys **MUST** be generated within cryptographic modules that are validated under FIPS 140 [21] and conform to NIST-approved algorithms, key sizes, and generation methods. Approved techniques will change over time as new algorithms are developed and as new threats or weaknesses are discovered. Currently approved techniques are summarized in SP 800-131Ar2, *Transitioning the Use of Cryptographic Algorithms and Key Lengths* [22], along with timelines for expected migrations to new techniques.

Organizations **MUST** ensure that cryptographic keys are associated with their intended purposes, systems, and devices. They **SHOULD** also maintain an inventory of the keys generated and used in their systems. This inventory **SHOULD** identify the type and purpose of each key, where it will be used and stored, and the expected lifetime of the key before it is replaced.

4.1.1.2. Distribution

Cryptographic keys need to be delivered securely to the systems and devices that are authorized to use them. Depending on the system and intended purpose, some cryptographic keys will be generated within the IdPs and authorization servers themselves. However, some cryptographic keys will be centrally generated and distributed to operational systems, such as when keys need to be provisioned onto multiple systems (e.g., to support load balancing or failover) or backed up to support incident response and recovery.

Cryptographic keys that are distributed or copied between systems need to be securely distributed using authenticated protected channels to ensure their confidentiality and integrity. In particular, secret or private cryptographic keys **SHOULD** never be communicated or exported in plaintext. Automation in key generation and distribution processes can help ensure that these controls are consistently applied. When manual processes are used, organizations **SHOULD** use dual or multi-person controls to mitigate accidental or malicious breaches of key material, particularly for long-lived or high-value keys.

When cryptographic keys are backed up or archived, distribution procedures **MUST** maintain confidentiality and integrity protections that are equivalent to those applied during operational use. Backup copies of private or secret keys **SHOULD** be encrypted using a key-encryption key stored on a tamper-evident hardware cryptographic module. The distribution of archived keys (e.g., for validation of historical assertions or token records) **MUST** be tightly controlled and auditable with clear procedures for authorization, recovery, and eventual destruction. IdPs and authorization servers **MUST** document and regularly review the systems and personnel that are authorized to receive backup or archived keys, including those stored off-site or in cloud-based disaster recovery environments.

Public keys (e.g., the public component of token-signing keys) may be widely distributed but **MUST** still be published and transmitted in a manner that ensures their authenticity and integrity. For example, in federated identity systems, IdPs may publish signing keys in SAML metadata or in OIDC discovery documents that reference a JSON Web Key Set (JWKS) endpoint. These metadata documents or JWKS endpoints shall be served over HTTPS and, when feasible, additionally signed or integrity-protected using a trusted certification path or external signature. Recipients **MUST** validate the authenticity of the public key and verify that its algorithm and key size are consistent with expected usage. Metadata **SHOULD** include information such as key usage, algorithm, and unique key identifiers to facilitate correct matching during key rollover.

4.1.1.3. Storage and Isolation for Keys and Cryptographic Functions

The exfiltration of keys and the exposure of cryptographic functions are significant threats to any system that uses assertions or tokens to manage access. An attacker could use a stolen or copied key to mint and sign legitimate assertions and tokens that can facilitate access to a protected system for which they would be valid.

To prevent this, any system assessed at moderate or above **MUST** provide hardware-based mechanisms to store and use cryptographic keys for signing assertions and tokens. To appropriately protect and manage these keys and their associated cryptographic functions, agencies **MUST** use one of the following techniques:

- **Hardware isolation using Hardware Security Modules (HSMs).** An HSM is a physical computing device that provides tamper-evident and intrusion-resistant security features, supports management of digital keys and other secrets, and performs cryptographic operations. For all FISMA systems that are categorized as high, hardware isolation via HSMs **MUST** be used.
- **Hardware isolation using embedded processors.** In addition to HSMs, organizations can apply isolation by leveraging secure, segregated components on servers, such as Trusted Platform Modules (TPMs) or other secure enclaves. These mechanisms leverage hardware components to achieve tamper resistance and limited access, but they do not have as robust a set of security features as HSMs and are often less isolated from other components.
- **Confidential computing.** Hardware-enabled features can be used to store and use cryptographic keys in isolated computing environments that are not accessible to the operating system or applications running on the host system. These technologies often use a combination of virtualization and memory encryption techniques to protect secrets from exposure or compromise through the host. Such techniques can provide improved performance or reduced cost compared to dedicated hardware isolation, but they present a larger attack surface.

For systems rated low, agencies **MAY** implement any of the above techniques or **MAY** use software isolation. Software isolation achieves similar goals, though the keys are stored in a separate software location, making them more vulnerable to exfiltration or copying. Though software-based isolation supports greater scale and performance, making them a reasonable option for low-risk access scenarios, they are more exposed to compromise or accidental leakage than hardware-based approaches.

4.1.1.4. Key Usage Periods and Rotation

Key usage periods define the maximum duration that a cryptographic key may be actively used for operations such as signing, decrypting, or encrypting data. IdPs and authorization servers **SHOULD** establish key usage periods based on a risk-informed assessment that considers the sensitivity of the data and functions protected by that key, the scope of resources protected by the key, and the key’s exposure to potential compromise. Shorter cryptoperiods reduce the impact of key leakage or theft and promote resiliency within the system.

To enforce consistent application of these cryptoperiods and reduce the risk of human error, organizations **MUST** document approved cryptoperiods and **SHOULD** employ automated key rotation mechanisms. These mechanisms **SHOULD** schedule and execute key rollover in advance of expiration to provide overlapping acceptance of old and new keys and support interoperability across RPs and clients. Systems **SHOULD** maintain versioning metadata (e.g., key identifiers, not-before/expiration timestamps, key usage information) to allow recipients to validate tokens and assertions without disruption.

Different types of cryptographic keys used by CSPs and agencies warrant distinct rotation strategies and schedules. Assertion and token signing keys are high-risk and **SHOULD** be rotated frequently. Table 4 provides recommended maximum key usage periods for the keys used to sign identity tokens and access tokens.

Table 4. Key usage period recommendations

Scenario	Recommended Max Signing Key Usage Period
CSP hosted system with multi-tenant keys *Example: A CSP hosts a platform where token and assertion signing services use common key-management capabilities across their FISMA tenants.*	30 days
CSP hosted system with single-tenant keys *Example: A CSP uses unique key-management services and keys for each tenant in their environment, and keys issued for one tenant are not valid for any other.*	3 months
On-premises system *Example: An agency deploys an on-premises SSO or identity management system where keys are unique and key management does not need to function across system or trust boundaries.*	1 year

The security of identity assertions and access tokens depends on Transport Layer Security (TLS) and the web public key infrastructure (PKI) certificates. TLS certificates used for HTTPS endpoints follow separate certificate authority policies but **SHOULD** be managed via automated certificate management systems. Across all key types, cryptoperiod enforcement and automated rotation workflows help maintain a strong security posture and facilitate rapid response to emerging threats or vulnerabilities.

4.1.1.5. Key Revocation and Destruction

Organizations **SHOULD** employ technical and procedural mechanisms to signal that a cryptographic key should no longer be trusted, such as when that key is outdated, no longer being used, or suspected of compromise. This is typically achieved by removing these keys from published metadata (e.g., JWKS endpoints in OIDC, SAML metadata documents) so that RPs and clients no longer recognize them as valid.

Following removal from distribution, all instances of retired private or secret keys **MUST** be securely destroyed to prevent unauthorized recovery or reuse. This includes operational, backup, and archived copies. Destruction **SHOULD** be automated where possible, logged for accountability, and integrated into broader key life cycle processes. Systems may also monitor for references to retired key identifiers as a signal of misconfiguration or attempted misuse.

4.2. Control Enhancement 2: Verification of Identity Assertions and Access Tokens

Enhancement. The source and integrity of identity assertions and access tokens are verified before granting access to system and information resources.

Discussion. This includes verification of digital signatures protecting identity assertions and access tokens, as well as included metadata. Metadata includes information about the access request, such as information unique to the user, system or information resource being accessed, or the transaction itself (e.g., time). Protected system and information resources could include connected networks, applications, and APIs.

4.2.1. Additional Guidance for the Verification of Identity Assertions and Access Tokens

Enabling appropriate access controls and protections of data and resources requires both the RP and the token or assertion issuer to execute on a set of responsibilities. Issuers must properly format, structure, and sign assertions so that the RP can verify the information and make informed access decisions. RPs must evaluate each token at the lowest level of usage (e.g., domain, application, data) to ensure that the assertion or token represents a user or software with appropriate permissions for the services they are attempting to access. In this process, it is particularly important to verify that the digital signature protecting an assertion is signed by the correct key or keys and ensure that this meets the policies of the target resource.

4.2.1.1. Assertion and Token Contents

The content of assertions and tokens will vary based on the specific protocols being used and the access needs of an agency. It will also vary between identity tokens (which often carry more attributes and data) and access tokens. Agencies and CSPs **MUST** document each type of token and assertion accepted within their infrastructure, as well as the mandatory data elements within those tokens and assertions. This is critical to effectively verifying the assertion and token, as well as supporting access decisions and interoperability across a specific system. Consistent with SP 800-63C, *Federations and Assertions* [16], assertions and tokens **MUST** contain at least the following information:

1. **Issuer identifier:** An identifier for the issuer of the assertion or the identity token (e.g., the enterprise IdP).
2. **Subject identifier or client identifier:** An identifier for the user to which the assertion applies.
3. **Audience identifier:** An identifier for the party intended to consume the assertion or identity token (i.e., the RP or application).
4. **Issuance time:** A timestamp that indicates when the assertion or token was issued.
5. **Validity time window:** A period of time outside of which the assertion cannot be accepted as valid for the purposes of authenticating the user and starting an authenticated session.

6. **Assertion or token identifier or nonce:** A value that uniquely identifies this assertion and is used to prevent attackers from replaying prior assertions.
7. **Authentication time:** A timestamp that indicates when the last primary authentication event occurred (e.g., at an enterprise or external IdP).
8. **Signature:** A digital signature or message authentication code (MAC), including the verification key identifier, that covers the entire assertion.

4.2.1.2. Key Scoping and Usage

Since not all keys will be protected to the same level, keys for lower risks or different services may be compromised. For example, a CSP may have commercial signing keys that are protected differently than those used by their government clients. To prevent a compromised key from being used outside of its intended environment, digital signatures on assertions and identity tokens must be validated as being correct and appropriate for use within the target environment. This protects against both accidental access and intentional or malicious acts.

In addition to the isolation of components that store signing keys and execute sensitive key-management functions, organizations need to consider the potential impacts of key validity for use across different domains or cloud tenants. For federal information systems, keys used to sign tokens and assertions need to be specific to the FISMA-approved tenants that they support. For example, a key that was used to sign a token or identity assertion in a CSP's commercial tenant should not be valid for signing assertions and tokens that were issued in a FEDRAMP-approved tenant or instance. Such validation failures can enable a compromised signing key collected through one tenant to compromise assertions and tokens in another. Additionally, CSPs and their federal customers need to determine, document, and limit key scopes and, if federation (i.e., sharing authentication or identity data between different security domains) is expected, appropriately capture these in trust agreements that define the expectations and security controls associated with assertion and token usage between the different domains. As such:

- Keys used to digitally sign assertions and tokens MUST be scoped at the lowest reasonable level (e.g., a single tenant, customer, set of customers) to reduce the impact of a key compromise incident.
- Keys generated for one defined segment MUST not be usable outside of that segment.
- The validity, source, and integrity of all identity assertions and tokens MUST be confirmed by authorization services before access is granted.
- All federation scenarios MUST be associated with a trust agreement that is consistent with the assessed FAL, as defined by a DIRA consistent with SP 800-63-4 and SP 800-63-4C.
- All federation requests MUST come through approved IdPs that are managed through an allowable mechanism consistent with SP 800-63C-4 and based on the assessed FAL of an online service or transaction.

- Policy enforcement points (e.g., at the application level) that rely on access tokens and identity assertions **MUST** confirm the validity, scope, source, and integrity of access tokens before granting access to resources.

4.3. Token Management

Enhancement. In accordance with [Assignment: organization-defined identification and authentication policy], assertions and access tokens are:

- Generated
- Issued
- Refreshed
- Revoked
- Time-restricted
- Audience-restricted

Discussion. An access token is a piece of data that represents the authorization granted to a user or NPE to access specific systems or information resources. Access tokens enable controlled access to services and resources. Properly managing the life cycle of access tokens, including their issuance, validation, and revocation, is crucial to maintaining the confidentiality of data and systems. Restricting token validity to a specific audience (e.g., an application or security domain) and restricting token validity lifetimes are important practices. Access tokens are revoked or invalidated if they are compromised, lost, or are no longer needed to mitigate the risks associated with stolen or misused tokens.

4.3.1. Additional Guidance for Token Management

In addition to the practices associated with protecting the keys used to sign assertions and tokens, the systems that manage and generate the actual assertions and tokens must be capable of providing additional security capabilities to improve the protection of assertions and tokens.

4.3.1.1. Token Refresh and Validity Length

Tokens need to be time-bound with short validity periods to minimize exposure if compromised. Automatic refresh mechanisms **SHOULD** be implemented with expiration policies tailored to token type, access scope, and usage context. The higher the risk associated with a protected resource, the shorter the validity period should be. Accordingly, access tokens and identity tokens **MUST** have defined, short lifetimes. Expired tokens **MUST** be rejected by authorization services and policy enforcement points. While a single, minimum threshold is very challenging to pin down due to the complex availability of different security features, it is generally recommended that access tokens and identity tokens **SHOULD** be valid for no more than one hour. However, given the potential variability of consuming agency use cases and risk tolerance, CSPs **SHOULD** make token validity length configurable by consumers and provide baseline validity periods based on the FISMA system categorization and authentication assurance level of applications.

The availability of certain features and capabilities can be used by CSPs and consuming agencies to adjust token validity periods. Table 5 discusses some of those considerations. The availability and effectiveness of these features **MUST** be documented and presented to consumers by CSPs so that effective decision-making can be made about validity periods.

Table 5. Risk mitigation capabilities

Capability	Description
Revocation	The ability for a token or assertion management system to revoke an existing token minimizes the risk associated with a compromised signing key or token/assertion from being reused for malicious purposes. Such a capability would need to operate together with compromise detection capabilities to be effective. Revocation is discussed in more detail in Sec. 4.3.1.2.
Compromise detection and session analysis	Compromise detection represents the ability to analyze and evaluate the context of a token or assertions used to determine whether it may have been compromised. Such a capability can inform decision-making at the resource server or token revocation actions. These systems can make use of numerous signals (e.g., geolocation, velocity) to inform decisions and actions. Session analysis, which supports compromise detection, is discussed more in Sec. 4.3.1.4.
Device registration or IP registration	Device registration allows access management systems to evaluate the trustworthiness of specific users as well as devices that are associated with the enterprise. Registration allows a specific device ID, IP address, or other identifier to be used to determine whether a user or API access request is coming from a trusted, registered device. It ensures that compromised or forged tokens cannot be used except in scenarios associated with specific devices or network endpoints.
Proof of possession and sender constraining	Proof-of-possession schemes, like device registration, seek to augment tokens and assertions with increased trust in the endpoints, clients, and devices that request and present tokens. These schemes implement additional means (e.g., device-specific public-private key pairs, mutual TLS binding) to verify the identity of an endpoint or user in a transaction before granting access. If strong sender binding is not achievable, IdPs and token services can consider channel-binding techniques that use TLS properties to constrain token usage.

Regardless of the validity period, implementing these capabilities improves the security of access decisions. CSPs **SHOULD** make them available, and consumers **SHOULD** implement them where possible.

Refresh token implementations allow service providers to interrupt token generation without incurring the cost of making every access token revocable. If identity tokens and access tokens are used in conjunction with refresh tokens, the refresh tokens may have a longer validity period than individual access or identity tokens.

When refresh tokens are used for human end users in interactive sessions, the lifetime of the refresh tokens **SHOULD NOT** exceed the reauthentication time frames, as defined in SP 800-63B-4, *Digital Identity Guidelines: Authentication and Authenticator Management* [15]. Consistent with tailoring practices in SP 800-63B-4, reauthentication time frames can be tailored based on the presence of session management compensating controls, such as sender constraining, proof-of-possession, or robust session analysis. Refresh tokens used for API access and non-human interactions **SHOULD** be accompanied by additional controls, including compromise detection and device registration. Where possible, revocation and proof-of-possession schemes **SHOULD** also be applied.

CSPs and agencies that use token and assertion-based systems **SHOULD** implement mechanisms for automatic token refresh, ensuring that the refresh tokens themselves are securely managed and subject to expiration and revocation policies. Conversely, for higher assurance processes, CSPs and agencies may choose not to implement refresh mechanisms and instead require the reauthentication and reissuance of identity or access tokens when they reach the end of their validity period or upon certain conditions, such as inactivity or session length restrictions. CSPs and agencies **SHOULD** also consider single-use or ephemeral access tokens for high-risk applications.

4.3.1.2. Token Revocation

Immediate and global token revocation is not always possible in stateless token implementations before the expiration of token validity periods. However, when short-lived access tokens are coupled with refresh tokens or reauthentication requirements, CSPs and agencies can substantially limit the time for which a compromised token is valid. Effective implementation of this means that IdPs and token services **MUST** have the ability to terminate sessions when a compromise is suspected. They **MUST** not accept refresh tokens or access tokens that are associated with an identity token or assertion that has been suspected of compromise without first reauthenticating the user. Further, IdPs and token services **MUST** ensure that once an identity token or assertion is revoked, this status propagates to any associated connected system.

A compromised identity token or assertion representing an end user often has many valid access tokens associated with it. Due to the distributed nature of stateless architectures, it becomes extremely complicated if not technically impossible to cut off all associated access tokens for a given user. To the extent possible, IdPs and token services **SHOULD** provide token and assertion revocation capabilities. These services **SHOULD** also consider implementing the Internet Engineering Task Force (IETF) Token Status List specification [17] and the Global Revocation specification [18]. Both are drafts but provide additional capabilities for more rapid revocation of access tokens. The extent and methods for revocation **MUST** be conveyed by CSPs to consuming agencies so that they understand and evaluate the risks associated with revocation events. Consuming agencies **MUST** assess the risks and impacts of revocation scenarios when integrating with CSPs.

4.3.1.3. Audience Restriction

Ensuring that tokens and assertions are only valid for their associated audience is critical to securing any indirect authentication system. As such, all tokens and assertions **MUST** include explicit audience fields, and all access control mechanisms **MUST** reject tokens with incorrect or missing audience restrictions. Access control mechanisms that receive tokens or assertions with missing or incorrect audience fields **SHOULD** immediately generate a security alert. Further, IdPs and token services **MUST** limit the scope of tokens and assertions to the minimum necessary for the intended operations.

4.3.1.4. Session Monitoring and Analysis

The continuous monitoring and analysis of token usage is essential for detecting compromise, enforcing contextual access controls, and supporting forensic investigations. To do so:

- IdPs, token services, and access management tools **MUST** implement capabilities to monitor token and assertion usage patterns. This can include evaluating geolocation, device data, and velocity anomalies. These capabilities **SHOULD** support the ability to correlate tokens and assertions (or the underlying identity) to actions taken across an enterprise or CSP services.
- It is also essential to integrate this information with other security mechanisms so that they can be correlated with event and threat data at different levels. Specifically, token and assertion usage data **MUST** be integrated with Security Information and Event Management (SIEM) software and — where appropriate — User and Entity Behavior Analytics (UEBA) systems and other security mechanisms.
- Logging is critical to both the detection and investigation of security incidents. All components **MUST** maintain tamper-resistant logs of token and assertion-related events and structure them to meet the requirements of SP 800-92, *Guide to Computer Security Log Management* [23], and OMB Memo 21-31, *Improving the Federal Government's Investigative and Remediation Capabilities Related to Cybersecurity Incidents* [24]. CSPs who offer their services to federal agencies **MUST** make information about their log data related to tokens and assertions available to federal consumers and **SHOULD** make this information available through programmatic mechanisms.

5. Threats and Attacks

While token and assertion-based systems (e.g., SSO, federation, API access) enable scaled modern infrastructure, they also require complex coordination between multiple infrastructure technologies and are contingent upon protecting the components that support trust between those different technologies. This includes the underlying cryptographic capabilities that support the integrity and validity of tokens and assertions, as well as the contents of those structures. This section summarizes the variety of threats and attacks that might target tokens and assertions and aligns these to the mitigations previously presented in this document. It is not an analysis of one specific event, nor is it a comprehensive set of protections against the multitude of threats to devices and endpoints involved in token and assertion-based systems (e.g., malware, compromised endpoints, insider threats).

Table 6. Threats to tokens and assertions

Threats	Description	Mitigation
Assertion or token manufacture or modification	The attacker generates a false assertion or modifies an assertion or token. This is typically executed in conjunction with a signing key compromise.	• Cryptographically sign assertions and tokens. • Validate token and assertion signatures prior to access. • Protect signing keys through appropriate isolation techniques (e.g., hardware, virtualized, software) based on the risks associated with the system. • Revoke and rotate signing keys consistent with the level of risk associated with the system. • Implement fine-grained, conditional access policies that rely on additional signals beyond valid tokens and assertions.
Assertion or token redirect	The assertion/token can be used in unintended contexts.	• Ensure that all tokens or assertions have audiences. • Validate the audience prior to all access decisions. • Restrict the scope and validity of signing keys to only the environments in which they are intended to be used. • Validate that the signature is scoped for use in the target system. • Implement fine-grained, conditional access policies that rely on additional signals beyond valid tokens and assertions.
Assertion or token replay	The assertion can be used more than once with same RP.	• Ensure that all assertions and tokens are uniquely identifiable (i.e., token/assertion ID) to prevent reuse by an attacker. • Validate the uniqueness of the token/assertion ID prior to all access decisions. • Implement fine-grained, conditional access policies that rely on additional signals beyond valid tokens and assertions.

Threats	Description	Mitigation
Signing key compromise	The IdP or authorization server's signing key is exfiltrated or exposed.	• Protect signing keys through appropriate isolation techniques (e.g., hardware, virtualize, software) based on the risks associated with the system. • Revoke and rotate signing keys consistent with the level of risk associated with the system. • Automate key-management systems to the extent practical. • Implement fine-grained, conditional access policies that rely on additional signals beyond valid tokens and assertions.

6. Additional Considerations

This section describes additional considerations related to securing assertions, tokens, and the relationship between CSPs and consumers.

6.1. Secure Integration and Configuration Between CSPs and Consumers

Consumers should leverage secure, predefined configurations to enhance security postures for IAM and secrets management.

- **Templates and design patterns.** Organizations should leverage CSP predefined templates and/or validated patterns for IAM policies to streamline complexity and eliminate misconfigurations. This is crucial in hybrid multi-cloud environments where expertise might be lacking.
- **Interoperability profiles.** Simplify interoperability across multiple CSPs by leveraging community-maintained interoperability and security technical profiles for IAM services as well as organizationally operated user provisioning and access review capabilities.
- **Tailored mitigations.** Enable consumer organizations to configure cloud services to implement tailored mitigations based on their security, availability, and interoperability needs.
- **Endpoint restrictions.** Restrict API endpoints to known, trusted sources (e.g., virtual private networks, IP allowlists) to secure non-human identities and prevent unauthorized access.
- **Routine testing.** Conduct regular access reviews and red team exercises to proactively identify vulnerabilities, test incident response capabilities, and strengthen the overall security posture.
- **Continuous education.** Provide continuous education for development and platform teams to ensure they stay updated on the latest security threats, best practices, and secure coding techniques and to foster a culture of security-first thinking.
- **Automated compliance.** Leverage CSP-provided tools for continuous policy assessment and compliance to evaluate policies for misconfiguration or poor scope.

6.2. Token and Assertion Presentation Methods

Tokens and assertions can be communicated to resource servers and RPs through front-channel presentation or back-channel presentation.

When communicated via the front channel, the assertion or token is passed through from the IdP to the RP through a user agent (e.g., a browser) or client. This exposes the token or assertion to the end user's environment and increases the risk of leakage, interception, or manipulation. While techniques like token encryption or sender constraining can mitigate some risks, front-channel presentation is not recommended for high-risk use cases.

In contrast, back-channel presentation involves the resource server fetching the token directly from the authorization server rather than through the client seeking access to the protected resource. In these flows, the authorization server first mints the token and then provides the client with a temporary reference to that token. The client provides this reference to the resource server to retrieve the token from the authorization server. This method significantly reduces the token's exposure to leakage, interception, and injection and provides an opportunity to detect forged tokens that were not issued by the legitimate authorization server. Similar methods can be used to communicate assertions from IdPs to RPs over a back channel.

6.3. Token and Assertion Encryption

The need to encrypt the payload of tokens and identity assertions beyond expected transport-layer protections (e.g., TLS) depends on the sensitivity of the claims they contain and the exposure risk in their transmission and storage.

Tokens and assertions that contain personal information or privileged access indicators (e.g., roles, group membership, administrative flags) **SHOULD** be encrypted at the payload level between the issuer and the protected resource (or RP), especially when passed through potentially untrusted environments, such as browser redirects or mobile clients.

Even when personal information is not explicitly present, unencrypted tokens can still expose metadata (e.g., usernames, email addresses, issuer identifiers, scopes) that enable reconnaissance by adversaries, particularly in targeted attacks. For example, an unencrypted access token that includes an email address or role claim may help an attacker identify administrative accounts worth phishing or impersonating in a social engineering campaign.

Agencies and organizations **SHOULD** evaluate the sensitivity of the token or assertion payload as well as its exposure to leakage or interception through logs, browser or client storage, or redirects. The use of encryption mechanisms, such as JSON Web Encryption (JWE) for JWTs or SAML Encrypted Assertions, **SHOULD** be considered in contexts where exposure could facilitate privilege escalation, user targeting, or cross-tenant information leakage.

6.4. FAL3 Assertions

In some high-risk scenarios, RPs may choose not to rely on the indirect authentication of users from a federated partner or even an enterprise IdP and instead take advantage of assertions and tokens to provision users (i.e., IdP-asserted attributes) while independently authenticating them. SP 800-63C, *Federation and Assertions,* provides two primary means to achieve this at FAL3: holder-of-key (HoK) assertions and RP-bound authenticators.

- **Holder-of-key assertions.** An HoK assertion includes a unique identifier for an authenticator that can be verified independently by the RP, such as the public key of a certificate controlled by the subscriber (e.g., smart card authenticated mutual TLS).

- **RP-bound authenticators.** This type of authenticator is bound to the RP subscriber account and managed by the RP. It can be given to the subscriber by the RP or provided by the subscriber (e.g., an RP or a user-provided hardware security key).

Since FAL3 with HoK or bound authenticators presents a high-friction user experience, it is not expected to be deployed for most use cases, even in high-assurance scenarios. However, it remains a viable option for specific protected resources with elevated risk, or in cases where an RP trusts the accuracy of user attributes from an external domain but lacks confidence in IdP authentication because of the heightened risk of token or assertion theft or forgery. CSPs **SHOULD** make this available to consuming federal agencies if the agency risk assessment indicates a need for enhanced protections. This would require integration with agency certificate authorities or functionality to support bound authenticators.

6.5. Device-Bound Session Credentials

Device-bound session credentials (DBSC) [26] are an emerging technology that is designed to help protect against cookies and session hijacking threats. The protocol uses the secure enclave on devices to generate a JWT with a public-private key pair associated with a user's device. This is then shared with the browser to create a strong binding between that specific device and the session in which the user is interacting with the browser and a specific web application. DBSC aids substantially in ensuring that an ongoing, authenticated session between a device and a web application remains secure and has not shifted to a new, potentially compromised device or attacker who may have been able to compromise a session cookie or access token. It can be used to extend session times or validity periods for an access token used in browser-based workflows.

6.6. Risk Signal Frameworks

Risk signal frameworks consist of data structures and protocols that can enable the conveyance of event information between different participants in a federation or SSO scheme. Specific events (e.g., password reset, account compromise) can be sent between connected partners to limit the impact of a security event in a federated or SSO scheme. Protocols (e.g., OpenID Foundation's Shared Signal Framework [27]) create a means and method for improving the security of cross-domain scenarios that involve the use of tokens, assertions, and federated account information. As recommended in SP 800-63C, both agencies and CSPs should consider adopting this or similar models based on their available technologies. The value and risk mitigation capabilities of shared signals models will depend on the signals, timeliness, and accuracy of the information. Agencies and CSPs will need to evaluate the degree to which risk signals can integrate with revocation and verification capabilities.

References

[1] Joint Task Force (2020) Security and Privacy Controls for Information Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-53r5, Includes updates as of December 10, 2020. https://doi.org/10.6028/NIST.SP.800-53r5

[2] Joint Task Force (2025) Security and Privacy Controls for Information Systems and Organizations, 5.2.0 (National Institute of Standards and Technology, Gaithersburg, MD), Cybersecurity and Privacy Reference Tool. Available at https://csrc.nist.gov/projects/cprt/catalog#/cprt/framework/version/SP_800_53_5_2_0/home

[3] Cybersecurity and Infrastructure Security Agency (2023) Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software. (Cybersecurity and Infrastructure Security Agency, Washington, D.C.). Available at https://www.cisa.gov/resources-tools/resources/secure-by-design

[4] Souppaya MP, Scarfone KA, Dodson DF (2022) Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-218. https://doi.org/10.6028/NIST.SP.800-218

[5] Ross R, McEvilley M, Winstead M (2022) Engineering Trustworthy Secure Systems. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-160v1r1. https://doi.org/10.6028/NIST.SP.800-160v1r1

[6] Joint Task Force (2012) Guide for Conducting Risk Assessments. (National Institute of Standards and Technology, Gaithersburg, MD) NIST Special Publication (SP) NIST SP 800-30r1. https://doi.org/10.6028/NIST.SP.800-30r1

[7] Joint Task Force (2018) Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-37, Rev. 2. https://doi.org/10.6028/NIST.SP.800-37r2

[8] Ragouzis N, Hughes J, Philpott R, Maler E, Madsen P, Scavo T (2008) Security Assertion Markup Language (SAML) V2.0 Technical Overview. (Organization for Advancement of Structured Information Standards (OASIS) Open, Woburn, MA), SAML 2.0. Available at https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html

[9] Cybersecurity and Infrastructure Security Agency (2021) Advanced Persistent Threat Compromise of Government Agencies, Critical Infrastructure, and Private Sector Organizations (Cybersecurity and Infrastructure Security Agency, Washington, D.C.), Cybersecurity Advisory AA20-352A. Available at https://us-cert.cisa.gov/ncas/alerts/aa20-352a

[10] Microsoft (2023) Analysis of Storm-0558 techniques for unauthorized email access. (Microsoft, Redmond, WA). Available at https://www.microsoft.com/en-us/security/blog/2023/07/14/analysis-of-storm-0558-techniques-for-unauthorized-email-access/

[11] Cybersecurity and Infrastructure Security Agency (2024) Hybrid Identity Solutions Guidance. (Cybersecurity and Infrastructure Security Agency, Washington, D.C.).

Available at https://www.cisa.gov/resources-tools/services/hybrid-identity-solutions-guidance-hisg

[12] Hardt D (2012) The OAuth 2.0 Authorization Framework. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) 6749. https://doi.org/10.17487/RFC6749

[13] Temoshok D, Galluzzo R, LaSalle C, Lefkovitz N, Regenscheid A, Choong YY, Proud-Madruga D, Gupta S (2025) Digital Identity Guidelines. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-63-4. https://doi.org/10.6028/NIST.SP.800-63-4

[14] Temoshok D, Abruzzi C, Choong YY, Fenton JL, Galluzzo R, LaSalle C, Lefkovitz N, Regenscheid A, Vachino M (2025) Digital Identity Guidelines: Identity Proofing and Enrollment. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-63A-4. https://doi.org/10.6028/NIST.SP.800-63A-4

[15] Temoshok D, Fenton JL, Choong YY, Lefkovitz N, Regenscheid A, Galluzzo R, Richer JP (2025) Digital Identity Guidelines: Authentication and Authenticator Management. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-63B-4. https://doi.org/10.6028/NIST.SP.800-63B-4

[16] Temoshok D, Richer JP, Choong YY, Fenton JL, Lefkovitz N, Regenscheid A, Galluzzo R (2025) Digital Identity Guidelines: Federation and Assertions. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-63C-4. https://doi.org/10.6028/NIST.SP.800-63C-4

[17] Looker T, Bastian P, Bormann C (2025) Token Status List. (Internet Engineering Task Force(IETF)), IETF Internet Draft. Available at https://datatracker.ietf.org/doc/draft-ietf-oauth-status-list/

[18] Parecki A (2025) Global Token Revocation. (Internet Engineering Task Force(IETF)), IETF Internet Draft. Available at https://datatracker.ietf.org/doc/draft-parecki-oauth-global-token-revocation/

[19] Office of Management and Budget Memorandum M-22-09, Moving the U.S. Government Toward Zero Trust Cybersecurity Principles. January 2022. Available at https://www.whitehouse.gov/wp-content/uploads/2022/01/M-22-09.pdf

[20] Barker EB (2020) Recommendation for Key Management: Part 1 – General. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-57pt1r5. https://doi.org/10.6028/NIST.SP.800-57pt1r5

[21] National Institute of Standards and Technology (2019) Security Requirements for Cryptographic Modules. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 140-3. https://doi.org/10.6028/NIST.FIPS.140-3

[22] Barker E, Roginsky A. (2019) Transitioning the Use of Cryptographic Algorithms and Key Lengths. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-131Ar2. https://doi.org/10.6028/NIST.SP.800-131Ar2

[23] Kent K, Souppaya MP (2006) Guide to Computer Security Log Management. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-92. https://doi.org/10.6028/NIST.SP.800-92

[24] Office of Management and Budget Memorandum M-21-31, Improving the Federal Government’s Investigative and Remediation Capabilities Related to Cybersecurity Incidents. August 2021. Available at https://www.whitehouse.gov/wp-content/uploads/2021/08/M-21-31-Improving-the-Federal-Governments-Investigative-and-Remediation-Capabilities-Related-to-Cybersecurity-Incidents.pdf

[25] Sakimura N, Bradley J, Jones M, de Medeiros B, Mortimore C (2023) OpenID Connect Core 1.0 incorporating errata set 1 (OpenID Foundation, San Ramon, CA). Available at https://openid.net/specs/openid-connect-core-1_0.html

[26] W3C (2025) Device Bound Session Credentials. Available at https://github.com/w3c/webappsec-dbsc

[27] Tulshibagwale A, Cappalli T, Scurtescu M, Backman A, Bradley J, Miel S (2025) OpenID Shared Signals Framework Specification 1.0 (OpenID Foundation, San Ramon, CA). Available at https://openid.net/specs/openid-sharedsignals-framework-1_0.html

Appendix A. List of Symbols, Abbreviations, and Acronyms

AAL
Authentication Assurance Level

API
Application Programming Interface

CSP
Cloud Service Provider

FAL
Federation Assurance Level

HSM
Hardware Security Module

IaaS
Infrastructure as a Service

IAL
Identity Assurance Level

IAM
Identity and Access Management

IdP
Identity Provider

JWKS
JSON Web Key Set

JWT
JSON Web Token

NPE
Non-Person Entity

OAuth
Open Authorization

OIDC
OpenID Connect

PaaS
Platform as a Service

RMF
Risk Management Framework

RP
Relying Party

SaaS
Software as a Service

SAML

Security Assertion Markup Language

SSO
Single Sign-On

TTP
Tactics, Techniques, and Procedures

Appendix B. Glossary

assertion
A statement from an *IdP* to an *RP* that contains information about an authentication event for a subscriber. Assertions can also contain identity attributes for the subscriber in the form of attribute values, derived attribute values, and attribute bundles.

assertion reference
A data object that is created in conjunction with an assertion and used by the RP to retrieve an assertion over an authenticated protected channel.

asymmetric keys
Two related cryptographic keys comprised of a public key and a private key that are used to perform complementary operations, such as encryption and decryption or signature verification and generation.

authenticated protected channel
An encrypted communication channel that uses approved cryptography in which the connection initiator (client) has authenticated the recipient (server). Authenticated protected channels are encrypted to provide confidentiality and protection against active intermediaries and are frequently used in the user authentication process. Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) are examples of authenticated protected channels in which the certificate presented by the recipient is verified by the initiator. Unless otherwise specified, authenticated protected channels do not require the server to authenticate the client. Authentication of the server is often accomplished through a certificate chain that leads to a trusted root rather than individually with each server.

credential
An object or data structure that authoritatively binds an identity — via an identifier — and (optionally) additional attributes to at least one authenticator that is possessed and controlled by a subscriber. A credential is issued, stored, and maintained by the CSP. Copies of information from the credential can be possessed by the subscriber, typically in the form of one or more digital certificates that are often contained in an authenticator along with their associated *private keys*.

cryptographic key
A value used to control cryptographic operations, such as decryption, encryption, signature generation, or signature verification. See *asymmetric keys* or *symmetric key*.

federation
A process that allows for the conveyance of identity and authentication information across a set of networked systems.

identity provider (IdP)
The party in a federation transaction that creates an assertion for the subscriber and transmits the assertion to the RP.

message authentication code (MAC)
A cryptographic checksum on data that uses a symmetric key to detect both accidental and intentional modifications of the data. MACs provide authenticity and integrity protection but not non-repudiation protection.

private key
A cryptographic key used with a public-key cryptographic algorithm that is uniquely associated with an entity and is not made public. In an asymmetric-key (public-key) cryptosystem, the private key has a corresponding public key. Depending on the algorithm, the private key may be used to 1) compute the corresponding public key, 2) compute a digital signature that may be verified by the corresponding public key, 3) decrypt keys that were encrypted by the corresponding public key, or 4) compute a shared secret during a key-agreement transaction.

public key
A cryptographic key used with a public-key cryptographic algorithm that is uniquely associated with an entity and that may be made public. In an asymmetric-key (public-key) cryptosystem, the public key has a corresponding private key. The public key may be known by anyone and, depending on the algorithm, may be used to 1) verify a digital signature that was generated using the corresponding private key, 2) encrypt keys that can be decrypted using the corresponding private key, or 3) compute a shared secret during a key-agreement transaction.

public-key certificate
A digital document issued and digitally signed by the private key of a certificate authority that binds an identifier to a subscriber's public key. The certificate indicates that the subscriber identified in the certificate has sole control of and access to the private key.

public-key infrastructure (PKI)
A set of policies, processes, server platforms, software, and workstations used to administer certificates and public-private key pairs, including the ability to issue, maintain, and revoke public-key certificates.

session
A persistent interaction between a subscriber and an endpoint, either an RP or a credential service provider. A session begins with an authentication event and ends with a session termination event. A session is bound by the use of a session secret that the subscriber's software (e.g., browser, application, OS) can present to the RP to prove association of the session with the authentication event.

shared secret
A secret used in authentication that is known to the subscriber and the verifier.

signing key
The cryptographic key used to create a signature. In asymmetric cryptography, the signing key refers to the private key of the cryptographic key pair. In symmetric cryptography, the signing key is the *symmetric key*.

single sign-on (SSO)
An authentication process by which one account and its authenticators are used to access multiple applications in a seamless manner, generally implemented with a *federation* protocol.

symmetric key
A *cryptographic key* used to perform both the cryptographic operation and its inverse (e.g., to encrypt and decrypt or to create a *message authentication code* and verify the code).