NIST Interagency Report
NIST IR 8587

Protecting Tokens and Assertions from Forgery, Theft, and Misuse

Implementation Recommendations for Agencies and Cloud Service Providers

Ryan Galluzzo
Andrew Regenscheid
Stephanie Nelson
Christine Lazcano

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

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

NIST Interagency Report
NIST IR 8587

Protecting Tokens and Assertions from Forgery, Theft, and Misuse

Implementation Recommendations for Agencies and Cloud Service Providers

Ryan Galluzzo
Applied Cybersecurity Division
Information Technology Laboratory

Andrew Regenscheid
Computer Security Division
Information Technology Laboratory

Stephanie Nelson
Accenture Federal
Supporting the Cybersecurity and Infrastructure Security Agency

Christine Lazcano
Cybersecurity Division
Cybersecurity and Infrastructure Security Agency

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

September 2026

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Arvind Raman, NIST Director and Under Secretary of Commerce for Standards and Technology

Report written with:

Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST or CISA, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

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

Publication History
Approved by the NIST Editorial Review Board on 2026-08-27

How to Cite this NIST Technical Series Publication
Galluzzo R, Regenscheid A, Nelson S, Lazcano C (2026) 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. https://doi.org/10.6028/NIST.IR.8587

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

Contact Information
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/final, 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 guidelines to help federal agencies and cloud service providers (CSPs) protect identity tokens, access 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 (SSO), federation, and application programming interface (API) access scenarios.

Keywords

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

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.

Patent Disclosure Notice

NOTICE: ITL has requested that holders of patent claims whose use may be required for compliance with the recommendations or requirements of this publication disclose such patent claims to ITL. However, holders of patents are not obligated to respond to ITL calls for patents and ITL has not undertaken a patent search in order to identify which, if any, patents may apply to this publication.

As of the date of publication and following call(s) for the identification of patent claims whose use may be required for compliance with the recommendations or requirements of this publication, no such patent claims have been identified to ITL.

No representation is made or implied by ITL that licenses are not required to avoid patent infringement in the use of this publication.

Table of Contents

1. Introduction 1
1.1. Purpose and Scope 1
1.1.1. Artificial Intelligence (AI) and AI Agents 2
1.2. Notations 2
1.3. Identity Management in Cloud and Hybrid Environments — Managing Responsibilities 2
1.4. Principles for Cloud Service Providers 4
1.5. Principles for Consuming Agencies 5
1.6. Continuous Monitoring and Evaluation 6
2. Overview of Assertions and Tokens 7
2.1. Terms and Concepts 8
2.2. Types of Assertions and Tokens 9
2.3. Uses of Tokens and Assertions 9
2.3.1. Token Management Methods 10
2.3.2. Single Sign-On and Federation 11
2.3.3. API Access Scenarios 12
3. Document Structure & Approach 14
4. IA-13: Identity Providers and Authorization Servers 15
4.1. Implementation Recommendations 15
4.1.1. Architecture and Design 15
4.1.2. Risk Management and Risk Assessment 18
4.1.3. Security Policy and Technical Documentation 18
4.1.4. Authorization Systems and Zero Trust Architectures 19
5. Control Enhancements 20
5.1. Control Enhancement 1: Protection of Cryptographic Keys 20
5.1.1. Additional Guidelines for the Protection of Cryptographic Keys 20
5.1.1.1. Generation 20
5.1.1.2. Distribution 21
5.1.1.3. Storage and Isolation for Keys and Cryptographic Functions 22
5.1.1.4. Key Usage Periods and Rotation 24
5.1.1.5. Key Revocation and Destruction 25
5.2. Control Enhancement 2: Verification of Identity Assertions and Access Tokens 25
5.2.1. Additional Guidelines for the Verification of Identity Assertions and Access Tokens 26
5.2.1.1. Assertion and Token Contents 26
5.2.1.2. Key Scoping and Usage 27

5.3. Token Management 28
5.3.1. Additional Guidelines for Token Management 28
5.3.1.1. Token Refresh and Validity Length 29
5.3.1.2. Token Revocation 30
5.3.1.3. Audience Restriction 31
5.3.1.4. Session Monitoring and Analysis 31
5.3.1.5. Token Protection 32
5.3.1.6. Workload Protections 32
5.3.1.7. Logging 33
6. Threats and Attacks 34
7. Additional Considerations 36
7.1. Secure Integration and Configuration Between CSPs and Consumers 36
7.2. Token and Assertion Presentation Methods 36
7.3. Token and Assertion Encryption 37
7.4. FAL3 Assertions 37
7.5. Device-Bound Session Credentials 38
7.6. Risk and Shared Signal Frameworks 38
7.7. Post-Quantum Cryptography Migration 38
References 40
Appendix A. List of Symbols, Abbreviations, and Acronyms 44
Appendix B. Glossary 46

List of Tables

Table 1. Example responsibility areas 3
Table 2. Stateful and stateless architectures 11
Table 3: Document Structure 14
Table 4. Design and architecture considerations 16
Table 5. Risk mitigation capabilities 29
Table 6. Suggested log data elements 33
Table 7. Threats to tokens and assertions 34

List of Figures

Fig. 1. Representative architecture: Token- and assertion-based systems 10

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 resulted from 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.

Federal agencies need to understand the architectures, designs, and deployment models of their CSPs to configure services that are consistent with their risk tolerance 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 guidelines 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 based on asymmetric cryptography when making access decisions. This often includes SSO, federation, application programming interface (API), and workload access scenarios. Systems that do not rely on asymmetric signed assertions and tokens (e.g., stateful management, schemes based on symmetric cryptography, API keys) are mentioned where relevant but out of scope for the controls covered.

[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, assets, and non-national security information systems used or operated by an agency, a contractor of an agency, or an organization on behalf of an agency.

1.1.1. Artificial Intelligence (AI) and AI Agents

Artificial intelligence (AI) systems — especially agentic AI systems (AI agents) — use signed tokens or assertions in many emerging IAM schemes. Organizations should apply these guidelines when agents use signed tokens to access systems, data, tools, or APIs. This document is not a comprehensive guide to addressing AI and AI agent access risks. Those risks create additional IAM challenges that require further guidelines and, in some cases, new or expanded standards and protocols. This document excludes those topics from scope. NIST and CISA recognize this gap and continue to develop practical guidelines as AI and agentic AI systems gain traction.[3]

1.2. Notations

This document uses the following typographical conventions in text to indicate the parameters for entities that claim conformance. However, conformance with these guidelines remains voluntary for CSPs and federal agencies unless otherwise determined through policy, including OMB policies, or other binding agreements (e.g., contracts, 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 to conform to these 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.

Agencies may choose to include these requirements in whole or in part based on risk management needs.

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 (e.g., authentication, federation, authorization), 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.

[3] More about NIST's AI Agent Security Initiative and Associated projects can be found at https://www.nist.gov/caisi/ai-agent-standards-initiative. NIST has also announced as possible AI Agent Identity Project at the NCCoE, which can be found here: https://www.nccoe.nist.gov/projects/software-and-ai-agent-identity-and-authorization.

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 enabling logging in various services (when not on by default), configuring log duration and storage, and configuring alert and notification mechanisms based on logs.

Consumer configuration choices are driven by risk assessments of the cloud-hosted services and the impacts of a loss of confidentiality, integrity, and 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. Coordination between CSPs and consuming agencies will largely be programmatic and/or automated, but some level of manual coordination will often be necessary. Table 1 provides an example of responsibility areas in a software-as-a-service (SaaS) environment.

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	●	●
Token Revocation	●	●

Actual responsibility boundaries vary by service model, contract terms, and technical capabilities exposed by the CSPs. Coordination is often implemented through documented interfaces, telemetry, and self-service controls, not solely through manual interactions.

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. Security must be built into the multitude of systems and offerings that CSPs manage. Such security 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]. CSP systems must also be designed with resilience in mind, including support for continuity of operations, rapid recovery from incidents, and redundant systems and services.
- **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 (e.g., IA-13: Identity Providers and Authorization Servers from SP 800-53, Rev. 5.1.1 and NIST's Digital Identity Guidelines) and establish effective communication channels to support logging and analyzing token- and assertion-related events. The intent of this principle is to ensure that consumers have sufficient information about a CSP offering to make informed risk and configuration decisions. It is not intended to add further compliance, audit, or assessment expectations on the CSPs or consuming agencies.
- **Configurability.** Different cloud service models (e.g., infrastructure as a service, platform as a service, SaaS) have different capabilities, and security features need to be optimizable by consumers to the greatest degree practical. This allows organizations to apply mitigations that match their risk tolerance and operational needs, such as additional logging and monitoring or incorporating additional security offerings from the CSP. CSPs should also provide critical consensus and federal risk management mitigations (e.g., those defined in this document and SP 800-53 more generally) by default with additional options for organizations with advanced requirements or specific needs. Appropriate training, documentation, and support capabilities must accompany these configurations.
- **Interoperability.** Providing essential support for standards-based architectures facilitates integration and management across hybrid and multi-cloud environments, enabling consumers to support redundancy and resiliency. Identity, authorization, and federation standards — such as Open Identity Connect (OIDC), Open Authorization (Oauth), and Security Assertion Markup Language (SAML) — reduces the complexity

introduced by custom integrations and supports a consistent security posture across the consumer's entire enterprise. This approach enables interoperability among connected components without mandating a single method for 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 use cloud services to support mission and service delivery are responsible for secure implementation. Agencies must ensure that procured cloud environments and services satisfy Federal Information Security Modernization Act (FISMA) risk management practices for transaction risk and sensitivity as well as resiliency goals to maintain operations in the event of incidents and outages. 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 system categorization [23] and FedRAMP impact levels, the consuming agency is responsible for completing a thorough risk assessment [6] in accordance with the NIST 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* [8]. 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. These frameworks do not comprehensively cover all aspects of token and assertion usage, and agencies should supplement them with the guidelines in this document.
- **Tailoring.** Control baselines serve as a starting point for selecting controls; organizations further tailor the controls to manage risk in accordance with their risk management strategy and risk tolerance. 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. Agencies should tailor controls based on the capabilities within selected CSP or on-premises environments.
- **Secure integration and configuration.** As indicated above, CSP services should be configurable. In coordination with their CSPs, consuming agencies are responsible for configuring their cloud environments to align with their security categorization, control selection, tailoring outcomes, and defined assurance.

While cloud-hosted services introduce complexity to the risk management process, they also enable substantial mission and business capabilities that agencies cannot replicate with organic systems and environments. By recognizing the shared responsibility for applying these principles, CSPs and agencies can improve threat management and enhance digital services for employees and the public. 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 "set and forget." Consistent with the RMF, CSPs and consuming agencies implement continuous monitoring capabilities for the components that they control. Consuming agencies utilize CSPs' offerings to continuously evaluate and monitor CSP environments, implement connections to on-premises or multi-cloud environments, and access events through logs or metrics at all levels to identify potential vulnerabilities or incidents. This continuous monitoring should extend to the accounts, clients, and entities that are issuing, consuming, validating, or represented by tokens or assertions. Consuming agencies — with support from CSPs — need to regularly review access policies, entitlements, access grants, and account statuses (e.g., provisioned/deprovisioned) with appropriate support from CSPs to limit the misuse of tokens or assertions.

In many real-world compromises involving tokens or assertions, agencies can detect issues only by monitoring account activity and analyzing CSP log data. CSPs are responsible for promptly notifying the consuming agency when they identify, suspect, or reasonably believe that a security incident may have caused unauthorized access, disclosure, alteration, or loss of agency data. CSPs must notify agencies in accordance with legal, regulatory, and contractual requirements and include sufficient detail for the agency to assess impact, initiate incident response procedures, and fulfill any downstream reporting obligations. CSPs are also responsible for coordinating with the consuming agency to ensure timely, accurate, and compliant communication and support follow-on investigation, containment, and remediation.

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 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) using asymmetric cryptography to provide the RPs with assurance 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, thus 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 ability to safeguard the cryptographic keys used to sign assertions and issue 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 rapidly compromise multiple protected resources.

Several major public incidents have exploited weaknesses in assertion and token practices. For example, a supply chain intrusion discovered in December 2020 resulted from a complex cyber attack that involved numerous tactics, techniques, and procedures (TTPs) to compromise thousands of organizations — including federal agencies — that were using a third-party security service provider. Once malware was delivered through a supply chain compromise, the attackers accessed protected resources by generating forged Security Assertion Markup Language (SAML) [9] assertions to bypass multi-factor 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 [10].

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 consumer signing keys that were inadvertently exposed. The compromised keys should only have been valid for tokens issued in the affected vendor's commercial environments, not their enterprise or government systems. However, due to token validation failures, these stolen keys were 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 [11].

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,[4] 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.[5]
- **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.[6]

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 have many different implementation patterns. This document will not define specific implementation patterns but instead focus on controls and outcomes that **MUST** be achieved across all implementations. These basic elements may be distributed to different physical or logical components in an architecture, reside in different domains or security boundaries (e.g., in federation scenarios),

[4] 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.
[5] This is known as a "client" in OAuth and a "service provider" in SAML.
[6] This is known as the "OpenID Provider" in OpenID Connect (OIDC).

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 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 [46]. 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.

Figure 1 illustrates a typical process flow for a basic token architecture and its components.

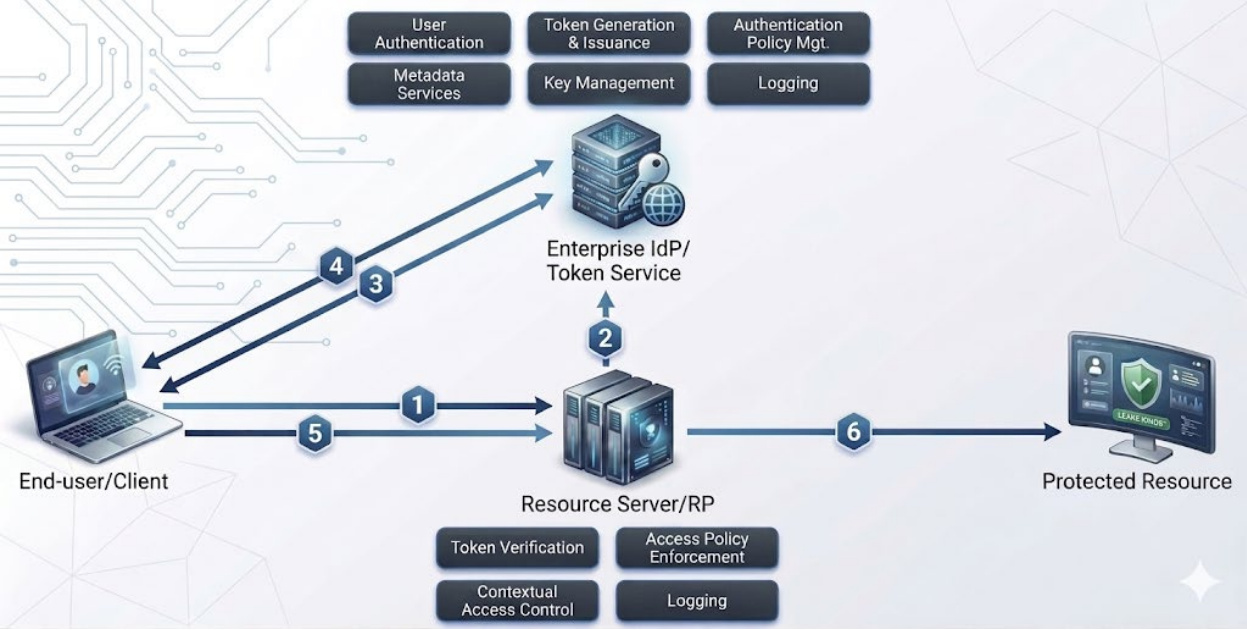

Fig. 1. Representative architecture: Token- and assertion-based systems

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 scenarios, the methods of authentication may not interact with an end user but rather employ device or machine authentication mechanisms (e.g., client-side certificates).

2.3.1. Token Management Methods

There are two primary methods for managing tokens and assertions in 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 and even hybrid systems.

The controls discussed in this document primarily focus on the use of asymmetrically signed tokens, which are more prevalent in and applicable to stateless models. However, most systems will use multiple kinds of tokens, so the controls in this document should be considered based on the implemented token types.

Table 2. Stateful and stateless architectures

Architecture	Description	Features & Considerations
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]) [47][50].	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 and resiliency issues, as stateful tokens require more infrastructure complexity. There are more challenges when utilizing stateful systems for API-based access scenarios. Not typically viable for cross-domain access or federation scenarios.
Stateless	All 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 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, as they are self-contained and include all the information the RP requires to authenticate the end user or client. Can be implemented securely but have limited revocability of assertions and tokens.

When making determinations about token and assertion management, CSPs and agencies need to evaluate which architecture works best for their scenario and risk environment.

2.3.2. Single Sign-On and Federation

SSO and federation use a common set of roles and components to achieve desired outcomes. In SSO and Federation scenarios, the end user requests access to a protected resource and is redirected to an enterprise or trusted IDP for authentication before being issued access tokens

that can be consumed by the protected resource. The following components are typical in an SSO or federation [28] 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 enterprise or trusted IdPs, which stores the 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, 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.

2.3.3. 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 like SSO use cases in which the user (i.e., the individual accessing the API) is authenticated locally or with an enterprise IdP, and subsequent calls to a hosted API service are authorized using signed access tokens.

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, workload, 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.

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), network address 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). Similar to 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. Document Structure and Approach

This document is built around IA-13: Identity Providers and Authorization servers, which was added to SP 800-53r5 in patch 5.1.1 in response to a series of high-visibility incidents that took advantage of vulnerabilities in the use of signed assertions and tokens. Each section restates the initial IA-13 related text (i.e., the control or a specific control enhancement) and provides more detailed requirements for achieving the specific outcomes related to its associated control text in SP 800-53. It is intended to provide a more detailed roadmap to secure the implementation of token and assertion-based systems for CSPs and agencies. Throughout the document, text from SP 800-53 is displayed with indentation and italicization to indicate the source material versus the expanded recommendations and considerations in this document.

As noted in Sec. 1.2, the controls identified in this document are voluntary for CSPs and federal agencies. At this time, IA-13 and its control enhancements have not been added to an SP 800-53 control baseline but may be in the future. Normative statements (e.g., **MUST**, **MAY**) are provided to create clear parameters for organizations that choose to claim conformance with this document. It also provides agencies seeking compliant solutions with a direct means to evaluate alignment with the controls defined in this document.

Table 3. Document structure

Section	Topic
Section 4	Covers the baseline IA-13 Control: IA-13: Identity Providers and Authorization servers
Section 5	Covers the control enhancements
Section 5.1	Covers Control Enhancement 1: Protection of Cryptographic Keys
Section 5.2	Covers Control Enhancement 2: Verification of Identity Assertions and Access Tokens

4. IA-13: Identity Providers and Authorization Servers

IA-13 was added to the 5.1.1 revision of the SP 800-53 control catalog [2] in direct response to several high-profile token-related security incidents. The control can 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.*

4.1. 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.

4.1.1. Architecture and Design

When implementing this 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 essential to supporting enterprise requirements. 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. There are benefits and risks to each of the core architectures for SSO and indirect authentication, as shown in Table 2. 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 4. Design and architecture considerations

Topic	Considerations
Stateful vs. Stateless	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. This document focuses on systems that leverage signed tokens and assertions, which tend to be more prevalent in stateless systems. However, these guidelines can be applied anywhere asymmetric signatures are used for token or assertion signing.
Signature Approach (Symmetric vs. Asymmetric Keys)	Tokens can be signed using symmetric or asymmetric keys. In general, most SSO and token management systems implement asymmetric keys when signing assertions and tokens. Asymmetric keys bring certain benefits over symmetric schemes, most notably easier key rotation, less complexity, and — as a result — greater scalability for distributed networks. 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. While implementers may choose to use symmetric key-management approaches, this document focuses on the use of asymmetrically signed tokens. Trust management is a critical component when implementing asymmetric signature approaches. Agencies need to understand the trust anchors, certificate management practices, and key hierarchies that they employ — or that are employed by CSPs — to provide the appropriate degree of confidence in token signing, signature verification, and token validation processes.

Topic	Considerations
Protocol Selection	Consistent with the recommendations of CISA's *Hybrid Identity Solution Guidance,* modern open standards are preferred for managing assertions and tokens [12]. OIDC [13] used in conjunction with OAuth [14] — specifically, the iGov Profile [15] — 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)	Both on-premises and cloud environments contain associated risks. On-premises environments might employ longer key rotation time frames, require manual rotations, and contain weak local storage but provide complete control over key isolation and cryptographic operations. While the integration of cloud environments into the modern federal enterprise comes with many opportunities, it also introduces complexity and visibility challenges. 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.
Non-Human Identities and Workloads	Workloads need to be uniquely identified and managed. Many modern protocols designed to support workload access management (e.g., the Secure Production Identity Framework for Everyone [SPIFFE]) leverage signed tokens as a technical mechanism for authenticating the workload [44]. Known as SPIFFE Verifiable Identity Documents (SVIDs), these documents are used consistent with other access tokens discussed in these guidelines and require similar capabilities and protections [45]. Approaches to workload and non-human identities that leverage short-lived tokens provide greater protection over those that rely on static secrets. For workload scenarios, agencies need to assess the functionality of their development life cycle and secure software engineering practices to identify how their selected architectures and those of the CSPs they work with will support IAM and secure automation goals when making architectural and design determinations. Unlike user identities, it is more likely that the authoritative source of identity information and configurations for workloads will be the CSPs, making communication and documentation between the CSP and consumer even more important to deploying effective, tailored solutions.
Mobile Applications	Mobile architectures often favor stateless architectures due to devices' intermittent connectivity, availability within low-connectivity scenarios, and ability for RP to validate tokens locally without a callback. Protocols like OIDC and OAuth are designed to support native mobile application architectures, while protocols like SAML were built around web application workflows. Agencies need to determine which protocols best support their planned operating environments as well as their identified security and user experience objectives.

Topic	Considerations
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.

4.1.2. Risk Management and Risk Assessment

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 [23][24], while the Digital Identity Guidelines in SP 800-63 define assurance levels for identity proofing (IAL), authentication (AAL), and federation, as well as a Digital Identity Risk Management (DIRM) model [8].

The DIRM — which includes conducting a DIRA — helps organizations identify baseline assurance levels and tailor and 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.

Taken together, these risk management frameworks complement this document by providing CSPs and agencies with system categorizations; baseline controls for authentication, federation, and identity management; and a starting point for tailoring implementations. When coupled with this document, they also provide a path to more detailed configurations to address specific risks related to token and assertion management scenarios that are not fully defined in other guidelines.

Existing laws and programs define how to apply these risk management frameworks. FISMA sets requirements for federal agencies, and FedRAMP sets requirements for CSPs.

4.1.3. 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), revocation practices, session management, and incident response processes. CSPs and agencies **MUST** also document the specific protocols and contents of tokens and assertions. When deploying specific protocols for tokens, CSPs and agencies **SHOULD** consult with published best practices, such as IETF's Best Current Practice for OAuth 2.0 Security [43], OASIS's Security and Privacy

Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0 [21], and JSON Web Token Best Current Practice [20].

4.1.4. Authorization Systems and Zero Trust Architectures

Authorization systems (e.g., policy enforcement, policy decision points) consume assertions and tokens to make access control decisions. They sit at the heart of enterprise security and are the foundation of zero trust architectures. As such, 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* [31], 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 verification 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.

Token and assertion-based systems **MUST** be integrated within a full secure identity management life cycle to ensure that token and assertion validity remains aligned with the provisioning, deprovisioning, and entitlements associated with core human and non-human identities. Identity governance tools **SHOULD** also be used to further augment these capabilities by integrating organizational controls around the core identity represented by a token and allowing for the enforcement of least-privilege, zero standing privileges, just-in-time approvals, and the separation of duties based on the sensitivity of the data or service being accessed.

5. 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 system 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.

5.1. 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.*

5.1.1. Additional Guidelines 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 enables attackers to forge digitally signed tokens and assertions to gain unauthorized access to connected systems.

5.1.1.1. 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 (Part 1, Revision 5), *Recommendations for Key Management: Part 1 – General* [32].

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 [33] and conform to NIST-approved algorithms, key sizes, and generation methods. Approved techniques will evolve as new algorithms are developed and as new threats or weaknesses are discovered. SP 800-131Ar2, *Transitioning the Use of Cryptographic Algorithms and Key Lengths* [34], summarizes currently approved techniques and provides timelines for expected migrations to new techniques.

Organizations **MUST** associate each cryptographic key with its intended purpose, system, and device. 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.

5.1.1.2. 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 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 hardware cryptographic module or another hardware-protected, isolated key-management mechanism appropriate to the system impact level. The distribution of archived keys (e.g., for validating 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 when used to verify authentication or authorization decisions. This requirement applies to public-key distribution channels, metadata, trust stores, discovery endpoints, and other mechanisms that recipients rely on to validate tokens, assertions, or related security decisions. It does not require every replication or incidental copy of a public key (e.g., logging, telemetry, archival records) to provide the same publication or transmission assurances. For example, in federated identity systems, IdPs may publish signing keys in SAML metadata [9], OAuth metadata [49], or in OIDC discovery documents [13] that reference a JSON Web Key Set (JWKS) endpoint [48]. These metadata documents or JWKS endpoints **MUST** be served over HTTPS and, when feasible, additionally signed or integrity-protected using a trusted certification path or external signature.

Trust anchors and trust constraints that are used to validate published keys **MUST** be explicitly defined, controlled, and reviewed. Organizations **MUST NOT** rely on broad default platforms, domains, or enterprise trust stores unless the trusted issuers, certificate policies, name constraints, path constraints, key usages, and authorized purposes are appropriate for the identity system and are documented. Trust relationships **MUST** be scoped to the RP, federation, tenant, or application context in which the keys are intended to be used.

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.

When certificates are used to represent or bind signing, encryption, or verification keys, the IdPs, authorization servers, and RPs **MUST** document whether certificates are self-signed, issued by an enterprise PKI, provided by a cloud or external key-management service, or issued by another trusted authority. Recipients **MUST** verify that certificate fields (e.g., key usage, issuer, validity period, applicable policy constraints) are consistent with the intended use and application.

Certificates and PKI-based trust models are most useful when keys are generated, distributed, or rotated at scale, and RPs validate public keys through common trust anchors, certificate paths, and associated constraints. A centralized PKI may not be necessary or appropriate if recipients instead maintain explicit trust lists of authorized public keys or use protocol-specific discovery mechanisms to obtain current public keys. However, the mechanisms used to distribute trust lists or discover new public keys are critical to the security of token-based systems. Organizations **MUST** document how trusted public keys are added, updated, revoked, removed, and audited and **MUST** protect the authenticity, integrity, timeliness, and governance of those trust lists.

5.1.1.3. 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. If an attacker obtains a signing key or gains unauthorized access to a signing function, they could mint assertions or tokens that appear legitimate and use them to gain access to protected systems. To address these risks, systems **MUST** protect the keys used to sign assertions and tokens, both in storage and during cryptographic use. The appropriate protection mechanism depends on the impact level of the system, the sensitivity of the protected resources, and the operational environment.

For systems assessed at **moderate impact** or above, organizations **MUST** store signing keys using a hardware-based, hardware-backed, or otherwise isolated key storage mechanism. These mechanisms are intended to reduce the risk that signing keys can be extracted, copied, or exposed through compromise of the application, host operating system, or adjacent workloads. Signing keys and key material **MUST NOT** be persistently stored on the applications, servers, virtual machines, or container systems that use them. When key material must be made available to those systems, it **MUST** be retrieved only at system boot, launch, or runtime as necessary and protected from unauthorized access or disclosure while present.

Mechanisms for protected key storage include but are not limited to:

- **Hardware isolation using hardware security modules (HSMs).** HSMs are physical or cloud-accessible devices or services that protect key generation, storage, management, and cryptographic operations. HSMs provide tamper-evident or tamper-resistant security features and are designed to prevent export of protected private keys.

- **Hardware isolation using embedded processors.** Embedded hardware components, secure enclaves, TPMs, and other cryptographic processors are used to protect keys from direct exposure to the host operating system or application environment. These mechanisms can provide hardware-backed key storage and restricted key use, though their security properties vary by implementation.
- **Isolated key and secrets management systems**. Cloud-based key management services, cloud HSM services, managed key management systems, and centralized secrets management systems can be used to store and manage signing keys or key material when they provide appropriate separation from the application, server, virtual machine, or container environment. These systems provide strong access control, administrative separation, auditing, life cycle management, and protection against unauthorized disclosure or export.

For systems assessed as **high impact**, organizations **MUST** additionally protect the use of signing keys and associated cryptographic functions within an isolated execution environment. This requirement is intended to reduce the risk that a compromised host, application, administrator account, or adjacent workload can directly observe, extract, or misuse signing keys or cryptographic operations.

- **Hardware isolation using HSMs**. HSMs perform signing operations internally and do not expose private keys to the calling application or host operating system.
- **Hardware isolation using embedded processors**. Embedded security coprocessors perform cryptographic operations within their hardware-protected environment.
- **Isolated execution environments**. Trusted execution environments, confidential computing environments, virtualization-based security environments, isolated service partitions, or comparable mechanisms can be used to isolate cryptographic operations from the general-purpose application and operating system environment. These mechanisms protect key material while in use, restrict access to signing functions, and prevent unauthorized observation, extraction, or misuse of signing keys.
- **Remote cryptographic signing services**. Cloud HSM services, managed signing services, distributed cryptographic services, or comparable remote key operation systems can be used when they store signing keys and perform signing operations within a protected service boundary. These services are intended to prevent unauthorized key export, restrict signing operations to authorized entities, protect keys in storage and during use, and provide audit logs for key access and signing operations.

For systems assessed at **low impact**, organizations **MAY** implement any of the above techniques or **MAY** use software-based isolation. Software isolation should use appropriate mechanisms to restrict access to signing keys and cryptographic functions, protect key material from unauthorized disclosure or export, limit key use to authorized processes or services, and log key management and signing operations. Software-based controls may be acceptable for some lower-risk access scenarios but are more vulnerable to exfiltration, copying, and accidental leakage than hardware-based approaches.

5.1.1.4. 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 **MUST** establish and document key usage periods. The length of the key usage periods **MUST** be 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, the mechanisms used to protect the key in storage and during use, and the operational complexity of safely rotating the key across RPs, clients, infrastructure, and deployment environments. Shorter cryptoperiods can promote resiliency by requiring organizations to automate key rotation, reduce dependence on manual key management processes, and regularly exercise key rollover procedures.

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) so recipients can validate tokens and assertions without disruption. For federation protocols and integrations that do not provide automated key discovery or rollover mechanisms, including some SAML deployments, organizations **SHOULD** account for the additional coordination burden on IdPs, RPs, and other applications. Where practical, organizations **SHOULD** prefer architectures and protocols that support the automated publication, discovery, validation, and retirement of signing keys.

Different types of cryptographic keys used by CSPs and agencies warrant distinct rotation strategies and schedules. Due to the risk of compromise, assertion and token signing keys **SHOULD** be rotated frequently. Rotation schedules **SHOULD** balance the security benefits of shorter key usage periods with secure deployment practices, the scale of systems to which they must be distributed, the need for verification pauses and rollback capability, and the strength of the key protection mechanisms.

For large-scale hosted systems, key rotation generally requires the creation of new keys and metadata, exposure of new keys to all relevant systems, selection of new keys for signing, continued acceptance of prior keys until previously issued tokens expire, removal of prior keys from trust stores or discovery mechanisms, and deletion or deactivation of prior keys. These steps may need to be performed across multiple deployment environments with appropriate buffers to support secure deployment and incident detection.

The appropriate maximum signing key usage period depends on the risk and impact associated with the system and the potential scale of a key compromise event. Organizations should consider the impact level of the system, the potential scale of users and systems that could be affected if the key is compromised, and the operational ability to rotate the key safely without disrupting service. For high-impact systems, signing keys **SHOULD** have a maximum active usage period of no more than 90 days. For moderate and low impact systems, longer usage periods **MAY** be appropriate but **SHOULD** be less than one year. In all cases, organizations **SHOULD** use

shorter key usage periods if key protection mechanisms are weaker or the ability to rapidly detect and recover from compromise is limited.

Those recommendations apply to the maximum usage period when the keys are actively used to sign tokens but are not intended to cover discovery, publication, or acceptance periods. Systems **MAY** publish a new verification key before it is actively used for signing and **MAY** continue to publish or accept a previous verification key for a limited period after signing with that key has stopped to allow previously issued tokens or assertions to expire and to prevent service disruption. Organizations **MUST** document the rollover process, including key creation, publication or exposure, activation for signing, overlap, deactivation, removal from discovery or trust stores, revocation when needed, and destruction.

The security of identity assertions and access tokens depends on Transport Layer Security (TLS) and 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.

5.1.1.5. 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 appropriately cleared or erased to prevent unauthorized recovery or reuse. This includes operational, backup, and archived copies. This process **SHOULD** be automated where possible, logged for accountability, and integrated into broader key life cycle processes. For ephemeral environments, terminating the environment can be considered erasure of the keys. Systems **SHOULD** also monitor for references to retired key identifiers as a signal of misconfiguration or attempted misuse.

5.2. 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.*

5.2.1. Additional Guidelines 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.

5.2.1.1. 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 that is available to consuming agencies as well as the mandatory data elements within those tokens and assertions. At a minimum, this includes identity tokens and assertions that are used to convey identity data between CSPs and consuming agency systems (e.g., for federation or SSO), any tokens used to connect to federal agency-controlled endpoints or services, and any tokens that are configurable by the consuming agency.

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* [28]*,* 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. Session times may be extended beyond the initial access token validity time through the use of refresh tokens and subsequent access tokens (see Sec. 4.3.1.1).
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.

Authorization tokens based on OAuth **SHOULD** contain scopes, which provide information about the actions that an application or user can take related to a protected resource. They can be deployed to support consent by providing users with clear information on what a service is accessing on their behalf and to enable least privilege by providing a limited set of actions or authorizations for the subject of the token. This can also be further enhanced by rich authorization requests, which provide the ability to request and relay fine-grained authorization data through access tokens [26].

5.2.1.2. Key Scoping and Usage

Not all keys will be protected to the same level. For example, a CSP may have commercial signing keys that are protected differently than those used by their government clients. Keys for lower risk systems or services may have lower security requirements and are subsequently more likely to be compromised. 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. Failure to manage key scopes can enable a compromised signing key collected through one tenant to compromise assertions and tokens in another. 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 used to digitally sign assertions **MUST** be scoped to a specific environment (e.g., development, testing, production).
- Keys used to sign identity tokens or assertions for non-FedRAMP, or federally authorized environments **MUST NOT** be valid for signing identity tokens and assertions in FedRAMP or federally authorized environments except in instances of federation with appropriate trust agreements.
- Keys **MUST** not be usable outside of their defined scope.
- Token and assertion signing keys **MUST NOT** be used for any other purpose.

- The scope, validity, source, and integrity of all identity assertions and tokens **MUST** be confirmed by the RP/resource server 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 and in accordance 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.

5.3. 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.*

5.3.1. Additional Guidelines 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.

5.3.1.1. 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 challenging to pin down due to the availability of different security features, it is 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.

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.
Advanced Authorization	The ability to rapidly push authorization updates to access control points and policy enforcement points at the protected resource, which can be used to terminate sessions associated with compromised tokens without requiring direct token revocation capabilities. As with revocation, this operates in conjunction with compromise detection, session analysis, and centralized policy distribution capabilities to propagate timely policies to protected resources.
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, token revocation actions, or risk signals that are distributed to connected relying parties. 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 Demonstrated Proof of Possession [DPoP], mutual TLS binding) to verify the identity of an endpoint or user in a transaction before granting access [16][17]. If strong sender binding is not achievable, IdPs and token services can consider channel-binding techniques that use TLS properties to constrain token usage.

The availability of certain features and capabilities can be used by CSPs and consuming agencies to adjust token validity periods. 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 4 discusses some of these considerations.

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. This validity period depends on the context in which the refresh token is used (e.g., mobile versus web-based applications), the type of user (e.g., human versus non-human), and the sensitivity of the protected resource being accessed. CSPs **SHOULD** make refresh token validity periods configurable by consuming agencies.

When refresh tokens are used for human end-users in interactive sessions, the lifetime of the refresh tokens **SHOULD NOT** exceed reauthentication time frames, as defined in SP 800-63B-4, *Digital Identity Guidelines: Authentication and Authenticator Management* [27]. 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, workloads, and non-human interactions **SHOULD** be as short as possible and 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. Authorization servers must protect refresh tokens from being replayed (e.g., through sender-constraining or rotation). 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.

5.3.1.2. 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** not accept refresh tokens that are associated with an identity token or assertion that has been suspected of compromise without first reauthenticating the user.

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. Further, IdPs and token services **SHOULD** provide a means to propagate revocation status to any associated RPs (e.g., token introspection endpoint, token status list, shared signaling). If this capability is available, connected RPs **MUST** reject tokens that have been revoked and terminate active sessions associated with revoked tokens.

The Open Identity Foundation's Shared Signals Framework (SSF) [38] and Continuous Access Evaluation Profile (CAEP) [22] provide standardized mechanisms for conveying security risk signals throughout federation partners and connected IdPs and RPs, which includes a session revocation signal. The SSF has seen significant adoption, and CSPs and consuming agencies **SHOULD** implement session revocation signals to provide greater session termination capabilities.

Additionally, the Internet Engineering Task Force (IETF) Token Status List specification (an active Internet-Draft adopted by the OAuth Working Group) [29] and the Global Token Revocation specification (an individual Internet-Draft in development) [30] provide additional capabilities for the more rapid revocation of access tokens and the propagation of current token statuses to connected RPs. CSPs and consumers should track these evolving standards.

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.

5.3.1.3. 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.

5.3.1.4. 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:

- CSPs and consumers **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.

- IdPs, token services, and access management tools should implement means of providing shared signals to enrich session management and access decision terminations. The Open ID Continuous Access Evaluation Profile supports numerous signals for communicating security event data between connected IdPs and RPs and across multi-cloud environments, and they should be implemented to improve overall session risk management capabilities.
- 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 provided in a format that is consumable by Security Information and Event Management (SIEM) software and — where appropriate — User and Entity Behavior Analytics (UEBA), Cloud-Native Application Protection Platforms (CNAPP), and other security mechanisms.

5.3.1.5. Token Protection

All tokens **MUST** be strictly protected from exposure in system logs, CI/CD pipelines, and related build or deployment artifacts and stored exclusively in approved secure storage mechanisms that meet or exceed the requirements defined for their applicable impact level. Any location that stores or transmits build output, execution traces, or debugging information is inherently high-risk and often widely accessible to developers, automation systems, or third-party integrations. When tokens appear in logs or pipeline artifacts, they can be inadvertently captured, forwarded, or retained far longer than intended, creating opportunities for unauthorized reuse. Pipelines **MUST** only retrieve secrets through secret management systems, inject them ephemerally at runtime, and ensure that they are never written to logs, console outputs, cache directories, or artifact stores. Any detected exposure of tokens should be treated as a security incident and remediated according to enterprise incident response procedures.

5.3.1.6. Workload Protections

Workload identities — including non-person entities, automated services, and machine-to-machine processes — **MUST** use tightly scoped, short-lived tokens that are issued exclusively through approved enterprise identity platforms and **SHOULD** use sender-constrained mechanisms (e.g., mTLS, DPoP [16][17]) whenever feasible. Incorporating SPIFFE-based workload identities strengthens this model by eliminating long-lived static credentials altogether. Instead, workloads obtain SPIFFE Verifiable Identity Documents (SVIDs) (e.g., SVID JWTs) dynamically from a trusted control plane [44][45]. This approach better aligns workload identity requirements with these guidelines and promotes tokens and workload credentials that are automatically rotated, cryptographically verifiable, and bound to the specific workload, reducing the risk of impersonation or stolen tokens being replayed.

5.3.1.7. Logging

Logging is critical for detecting and investigating 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* [35], and OMB Memo 26-14*, Ensuring Effective and Efficient Agency Logging and Network Visibility to Defend Against Evolving Cyber Threats* [36]. CSPs who offer their services to federal agencies **MUST** enable agencies to configure their services to meet the requirements of SP 800-92 and OMB Memo 26-14. This includes providing information about what logs are available, what data is logged, and what logs and log features are enabled by default versus what logs must be enabled by the consumer. Federal consuming agencies **MUST** evaluate the logging capabilities available to them and should enable logs that are "not on by default" to provide the most current and detailed information possible.

Tokens and assertions have unique data that is valuable to retain for logging purposes. Exactly what data might be retained will ultimately depend on the type of token or assertion being used and the logging services that are available. Table 5 provides a non-comprehensive set of recommendations for token and assertion data that may be included in logs. The tokens and assertions themselves **MUST NOT** be stored in logs. Additionally, PII or personal data contained in tokens **MUST NOT** be stored in logs.

Table 6. Suggested log data elements

Element	Description
Token or Assertion Type	The type of token being represented in the logs
Issuer	Who issued the token
Scopes	The scopes or authorizations associated with the token or assertion
Client/User	An identifier or obfuscated identifier (e.g., hashed value) for the entity to which the token has been issued
Audience	An identifier or obfuscated identifier (e.g., hashed value) for the audience or resource server that was the intended recipient of the token or assertion
Events	Timestamped events related to the token or assertion, such as: • Issuance • Refresh • Acceptance/Rejection • Revocation

Additionally, CSPs **MUST** log data related to tokens and assertions that are available to federal consumers and **SHOULD** make this information available through programmatic mechanisms. CSPs and consuming agencies **SHOULD** consider implementing open-source logging schemas and frameworks (e.g., Open Cybersecurity Schema Framework, OpenTelemetry, Elastic Common Schema) to support more consistent, interoperable log data.

6. 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. Table 7 summarizes the variety of threats and attacks that might target tokens and assertions and aligns with 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 7. 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. 4.2.1 • Validate token and assertion signatures prior to access. 4.2.1.2 • Protect signing keys through appropriate isolation techniques (e.g., hardware, virtualized, software) based on the risks associated with the system. 4.1.1 • Implement sender-constrained tokens (e.g., DPoP, mTLS). 4.3.1.1 • Revoke and rotate signing keys consistent with the level of risk associated with the system. 4.1.1.4 • Implement fine-grained, conditional access policies that rely on additional signals beyond valid tokens and assertions. 3.1.4 • Maintain tamper-resistant logs of token and assertion-related events. 4.3.1.7
Assertion or Token Redirect	The assertion/token can be used in unintended contexts.	• Ensure that all tokens or assertions have audiences. 4.3.1.3 • Validate the audience prior to all access decisions. 4.3.1.3 • Restrict the scope and validity of signing keys to only the environments in which they are intended to be used. 4.2.1.2 • Validate that the signature is scoped for use in the target system. 4.2.1.2 • Implement sender-constrained tokens (e.g., DPoP, mTLS). 4.3.1.1 • Implement fine-grained, conditional access policies that rely on additional signals beyond valid tokens and assertions. 3.1.4 • Maintain tamper-resistant logs of token and assertion-related events. 4.3.1.7

Threats	Description	Mitigation
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. 4.2.1.1 • Enforce sender constraining or (for refresh tokens) rotation. 4.3.1.1 • Validate the uniqueness of the token/assertion ID prior to all access decisions. 4.2.1.2 • Implement fine-grained, conditional access policies that rely on additional signals beyond valid tokens and assertions. 3.1.4 • Maintain tamper-resistant logs of token and assertion-related events. 4.3.1.7
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. 4.1.1.3 • Revoke and rotate signing keys consistent with the level of risk associated with the system. 4.1.1.4, 4.1.1.5 • Automate key-management systems to the extent practical. 4.1.1.4 • Implement fine-grained, conditional access policies that rely on additional signals beyond valid tokens and assertions. 3.1.4
Token Theft	The attacker gains access to a legitimate assertion/token or refresh token.	• Revoke and rotate signing keys consistent with the level of risk associated with the system. 4.1.1.4, 4.1.1.5 • Validate that the signature is scoped for use in the target system. 4.2.1.2 • Implement sender-constrained tokens (e.g., DPoP, mTLS). 4.3.1.1 • Cryptographically sign assertions and tokens. 4.2.1 • Implement fine-grained, conditional access policies that rely on additional signals beyond valid tokens and assertions. 3.1.4 • Maintain tamper-resistant logs of token and assertion-related events. 4.3.1.7

7. Additional Considerations

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

7.1. 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.
- **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.

7.2. 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. For OAuth [43] deployments, current best-practice guidelines should be followed for Proof Key for Credential

Exchange (PKCE), exact redirect URI matching, issuer validation, and related front-channel protections.

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.

7.3. 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.

7.4. 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 if 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.

7.5. Device-Bound Session Credentials

Device-bound session credentials (DBSC) [37] 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.

7.6. Risk and Shared Signal Frameworks

Risk and shared 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 [38]) 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.

7.7. Post-Quantum Cryptography Migration

Token and assertion-based systems rely on public-key cryptography to sign and verify assertions, identity tokens, access tokens, metadata, certificates, and related trust-

management artifacts and to establish the protected channels used to distribute keys and convey sensitive security information. Public-key mechanisms currently used in these systems — including RSA, ECDSA, and ECDH — are vulnerable to attacks by a cryptographically relevant quantum computer. Migration to post-quantum cryptography (PQC) impacts both the systems that issue and sign tokens or assertions and the RPs, resource servers, gateways, clients, and other components that validate, consume, store, transmit, or route them. Standardized in FIPS 203, ML-KEM specifies a key-encapsulation mechanism to support quantum-resistant encryption. Standardized in FIPS 204, ML-DSA specifies a quantum-resistant digital signature algorithm. Compared to many of the public-key mechanisms used today, these algorithms have larger public keys, signatures, and ciphertexts that affect protocol design, token size, certificate and metadata handling, storage, transmission, and hardware-backed key protection.

PQC migration for identity systems will depend on the maturity of protocol standards, product support, and deployment infrastructures. The standards used by identity federation and access-token protocols (e.g., OIDC, OAuth, JOSE, COSE, SAML, X.509), related metadata, and discovery mechanisms are still being updated to support PQC algorithms. Legacy federation systems, including some SAML deployments, might not be updated to support PQC. Hardware can also be a limiting factor when keys are protected or used in HSMs, embedded cryptographic processors, secure elements, or other isolated execution environments, since those components may require firmware, product, validation, or architecture updates before they can generate, store, or use PQC keys.

Agencies and CSPs should inventory uses of public-key cryptography in their token and assertion ecosystems and assess whether existing token conveyance patterns remain viable as PQC algorithms are introduced. Many PQC public keys and signatures are significantly larger than the elliptic-curve and RSA keys and signatures commonly used today, which may create challenges for stateless access tokens, such as signed JWTs. Browser cookies are commonly constrained to approximately 4 kb per cookie, making them increasingly unsuitable for storing or conveying PQC-signed JWT access tokens. HTTP request headers are also commonly subject to implementation-specific limits around 8 kb, which may constrain the space available for claims, scopes, and other authorization data. Where these constraints are significant, organizations may need to use more compact token formats, back-channel presentations, reference tokens, token introspection, session-state mechanisms, or other architectures that avoid repeatedly carrying large, signed objects through constrained channels.

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) Cybersecurity and Privacy Reference Tool. (National Institute of Standards and Technology, Gaithersburg, MD). 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) NIST 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) NIST SP 800-37r2. https://doi.org/10.6028/NIST.SP.800-37r2

[8] 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

[9] 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

[10] 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

[11] 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/

[12] 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

[13] 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

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

[15] Burgin K, Clancy T (2025) International Government Assurance Profile (iGov) for OAuth 2.0 – draft 09 (OpenID iGov Working Group, Internet Draft). Available at https://openid.net/specs/openid-igov-oauth2-1_0.html

[16] Campbell B, Bradley J, Sakamura N, Lodderstedt T, (2020) OAuth 2.0 Mutual-TLS Authentication and Certificate-Bound Access Tokens. (Internet Engineering Task Force (IETF)), IETF Request for Comment (RFC) RFC 8705. https://datatracker.ietf.org/doc/html/rfc8705

[17] Fett D, Campbell B, Bradley J, Lodderstedt T, Jones M, Waite D, (2023) OAuth 2.0 Demonstrating Proof-of-Possession (DPoP). (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) RFC 9449. Available at https://www.rfc-editor.org/rfc/rfc9449.html

[18] Burgin K and Clancy T, eds. (2025) International Government Assurance Profile (iGov) for OAuth 2.0. OpenID Foundation, OpenID iGov Working Group, draft 09. Available at https://openid.net/specs/openid-igov-oauth2-1_0.html

[19] Sheffer Y, Hardt D, and Jones MB (2026) JSON Web Token Best Current Practices. Internet Engineering Task Force (IETF), Internet-Draft draft-ietf-oauth-rfc8725bis-04. Work in progress. Available at https://datatracker.ietf.org/doc/draft-ietf-oauth-rfc8725bis/

[20] Würtele T, Hosseyni P, Luo K, and Fung A (2026) Updates to OAuth 2.0 Security Best Current Practice. Internet Engineering Task Force (IETF), Internet-Draft draft-ietf-oauth-security-topics-update-01. Work in progress. Available at https://datatracker.ietf.org/doc/draft-ietf-oauth-security-topics-update/

[21] Frederick H, Rob P, Eve M (2005) Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Open, saml-sec-consider-2.0-os. Available at https://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf

[22] Cappali T, Tulshigawale, A, (2025) OpenID Continuous Access Evaluation Profile 1.0 (Open ID Foundation, San Ramon, California). Available at https://openid.net/specs/openid-caep-1_0-final.html

[23] Federal Information Processing Standards Publication 199, Standards for Security Categorization of Federal Information and Information Systems, February 2004. https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.199.pdf

[24] National Institute of Standards and Technology (2006) Minimum Security Requirements for Federal Information and Information Systems. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publications (FIPS) NIST FIPS 200. https://doi.org/10.6028/NIST.FIPS.200

[25] 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

[26] Lodderstedt T, Richer J, Campbell B, (2023) OAuth 2.0 Rich Authorization Requests. (Internet Engineering Task Force (IETF)), IETF Request for Comments (RFC) RFC 9396. https://www.rfc-editor.org/rfc/rfc9396

[27] 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

[28] 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

[29] 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/

[30] 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/

[31] 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

[32] 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

[33] 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

[34] 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

[35] 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

[36] Office of Management and Budget Memorandum M-26-14, Ensuring Effective and Efficient Agency Logging and Network Visibility to Defend Against Evolving Cyber

Threats. May 2026. Available at https://www.whitehouse.gov/wp-content/uploads/2026/05/M-26-14-Ensuring-Effective-and-Efficient-Agency-Logging-and-Network-Visibility-to-Defend-Against-Evolving-Cyber-Threats.pdf

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

[38] 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

[39] Campbell B, Bradley J, Sakimura N, Lodderstedt T (2020) OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens. (Internet Engineering Task Force (IETF) IETF Internet Draft). Available at https://www.rfc-editor.org/rfc/rfc8705.pdf

[40] Fett D, Campbell B, Bradley J, Lodderstedt T, Jones M, Waite D (2023) OAuth 2.0 Demonstrating Proof of Possession (DPoP). (Internet Engineering Task Force (IETF), IETF Internet Draft). Available at https://datatracker.ietf.org/doc/html/rfc9449

[41] Sheffer Y, Hardt D, Jones M (2020) JSON Web Token Best Current Practices. (Internet Engineering Task Force (IETF), IETF Internet Draft). Available at https://datatracker.ietf.org/doc/html/rfc8725

[42] Wurtele T, Hosseyni P, Luo K, Fung A (2026) Updates to OAuth 2.0 Security Best Current Practice. (Internet Engineering Task Force (IETF), IETF Internet Draft). Available at https://datatracker.ietf.org/doc/draft-ietf-oauth-security-topics-update/

[43] Lodderstedt T, Bradley J, Labunets A, Fett D (2025) Best Current Practice for OAuth 2.0 Security. (Internet Engineering Task Force (IETF), IETF Internet Draft). Available at https://datatracker.ietf.org/doc/rfc9700/

[44] Spiffe (2026) X.509-SVID. (Spiffe, Internet Draft). Available at https://spiffe.io/docs/latest/spiffe-specs/x509-svid/

[45] Spiffe (2022) JWT-SVID. (Spiffe, Internet Draft). Available at https://spiffe.io/docs/latest/spiffe-specs/jwt-svid/

[46] Parecki A, McGuinness K, Campbell B (2026) Identity Assertion JWT Authorization Grant. (Internet Engineering Task Force (IETF), IETF Internet Draft). Available at https://datatracker.ietf.org/doc/draft-ietf-oauth-identity-assertion-authz-grant/

[47] Jones M, Bradley J, Sakimura N (2015) JSON Web Token (JWT). (Internet Engineering Task Force (IETF), IETF Internet Draft). Available at https://datatracker.ietf.org/doc/html/rfc7519

[48] Jones M (2015) JSON Web Key (JWK). (Internet Engineering Task Force (IETF), IETF Internet Draft). Available at https://datatracker.ietf.org/doc/html/rfc7517

[49] Jones M, Sakimura N, Bradley J (2018) OAuth 2.0 Authorization Server Metadata. (Internet Engineering Task Force (IETF), IETF Internet Draft). Available at https://datatracker.ietf.org/doc/html/rfc8414

[50] Bertocci V (2021) JSON Web Token (JWT) Profile for OAuth 2.0 Access Tokens. (Internet Engineering Task Force (IETF), IETF Internet Draft). Available at https://datatracker.ietf.org/doc/html/rfc9068

Appendix A. List of Symbols, Abbreviations, and Acronyms

AAL
Authentication Assurance Level

API
Application Programming Interface

CSP
Cloud Service Provider

DIRA

Digital Identity Risk Assessment

FAL
Federation Assurance Level

FISMA

Federal Information Security Modernization Act

HMAC

Hash-Based Message Authentication Code

HSM

Hardware Security Module

IaaS
Infrastructure as a Service

IAL
Identity Assurance Level

IAM
Identity and Access Management

IdP
Identity Provider

JSON

JavaScript Object Notation

JWKS

JSON Web Key Set

JWT

JSON Web Token

MFA
Multi-Factor Authentication

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

SPIFFE
Secure Production Identity Framework for Everyone

SSO
Single Sign-On

SVID
SPIFFE Verifiable Identity Documents

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.

authorization server
Authorization servers manage the authentication of the end user and generate or issue tokens with authentication data and user or client attributes.

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.

protected resource
The data, services, or applications that a client acting on behalf of a user is attempting to access.

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.

resource server
The resource server hosts the *protected resource* and is responsible for validating any identity or access tokens issued by the *authorization server*.

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 using 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).