The Printer Working Group

August 16, 2019
Best Practice 5199.10-2019

IPP Authentication Methods (IPPAUTH)

Status: Approved

Abstract: This Best Practice document provides implementation guidance on how to best integrate various authentication mechanisms used over IPP's HTTP and HTTPS transports into IPP protocol exchanges when printer access or print feature policy require authorization.

This is a PWG Best Practice document. For the definition of "PWG Best Practices", see:

http://ftp.pwg.org/pub/pwg/general/pwg-process30.pdf

This document is available electronically at:

https://ftp.pwg.org/pub/pwg/ipp/informational/bp-ippauth10-20190816-5199.10.odt
https://ftp.pwg.org/pub/pwg/ipp/informational/bp-ippauth10-20190816-5199.10.pdf

Title: IPP Authentication Methods *(IPPAUTH)*

Table of Contents

List of Figures

List of Tables

1. Introduction

The Internet Printing Protocol (hereafter, IPP) uses HTTP as its underlying transport [STD92]. When an IPP Printer is configured to limit access to its services to only those Clients operated by an authorized User, it challenges the Client for authentication credentials using one of the HTTP or TLS authentication methods. User experience problems can occur if the Printer or associated authentication and authorization infrastructure assumes that all User Agents are web browsers, since IPP Clients are HTTP User Agents but do not implement many content technologies used in contemporary web browsers, and their use of HTTP is constrained.

This document surveys the HTTP authentication methods employed today that support and are supported by IPP, and outlines limits, constraints and conventions that ought to be considered by Client developers, Printer developers, and Infrastructure Administrators when implementing support for one of these HTTP authentication methods in IPP communications, to ensure a high quality printing user experience.

2. Terminology

2.1. Conformance Terminology

Capitalized terms, such as MUST, MUST NOT, RECOMMENDED, REQUIRED, SHOULD, SHOULD NOT, MAY, and OPTIONAL, have special meaning relating to conformance as defined in Key words for use in RFCs to Indicate Requirement Levels [BCP14]. The term CONDITIONALLY REQUIRED is additionally defined for a conformance requirement that applies when a specified condition is true.

2.2. Protocol Roles Terminology

This document defines the following protocol roles in order to specify unambiguous conformance requirements:

Client: Initiator of outgoing IPP session requests and sender of outgoing IPP operation requests (Hypertext Transfer Protocol -- HTTP/1.1 [RFC7230] User Agent).

Printer: Listener for incoming IPP session requests and receiver of incoming IPP operation requests (Hypertext Transfer Protocol -- HTTP/1.1 [RFC7230] Server) that represents one or more Physical Devices or a Logical Device.

2.3. Other Terms Used in This Document

Authentication: The corroboration that a peer entity in an association is the one claimed. ([ITUX.800] definition for “peer entity authentication”)

Authorization: The granting of rights, which includes the granting of access based on access rights. ([ITUX.800])

User: A person or automata using a Client to communicate with a Printer.

2.4. Acronyms and Organizations

IANA: Internet Assigned Numbers Authority, http://www.iana.org/

IETF: Internet Engineering Task Force, http://www.ietf.org/

ISO: International Organization for Standardization, http://www.iso.org/

PWG: Printer Working Group, http://www.pwg.org/

3. Requirements

3.1. Rationale

Given the following existing specifications:

1. [STD92] defines the core Internet Printing Protocol/1.1
2. [RFC7235] defines HTTP/1.1 authentication in terms of the architecture defined in "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing" [RFC7230], including the general framework previously described in "HTTP Authentication: Basic and Digest Access Authentication" [RFC2617] and the related fields and status codes previously defined in "Hypertext Transfer Protocol – HTTP/1.1" [RFC2616].
3. [RFC7617] defines the 'Basic' HTTP Authentication Scheme
4. [RFC7616] defines HTTP Digest Access Authentication
5. [RFC4559] defines SPNEGO-based Kerberos and NTLM HTTP Authentication
6. [RFC6749] defines the OAuth 2.0 Authorization Framework
7. [RFC8252] describes best practices for OAuth 2.0 for Native Apps
8. [RFC8446] defines version 1.3 of the Transport Layer Security (TLS) protocol.

And given the need for Clients and Printers to provide and support a positive user experience while supporting these HTTP authentication methods and in many cases not supporting the full functionality of a Web browser, this IPP Authentication Methods Best Practices document should:

- Describe each HTTP authentication system;
- Highlight details and consider pitfalls that can impact the IPP Client user experience.

3.2. Use Cases

3.2.1. Authentication Required for Authorized Printer Access

Andy is at work and wants to print from his laptop. He finds and selects a printer on his network. The IPP Client in his laptop checks to see if using the Printer will require authentication, so that the User's expectations can be appropriately managed. The Printer responds with an authentication challenge, and the Client presents a user interface appropriate for the HTTP authentication type in the challenge. Andy provides his credential information to the Client, and the Client submits that to the Printer. The Printer authenticates Andy's credentials and confirms Andy's account is authorized to print without restrictions, and specifies the features he is authorized to use. The laptop provides the usual print dialog user interface, allowing Andy to select among those authorized print options.

3.3. Exceptions

3.3.1. Authentication Failure Prevents Access To Printer

Lisa is visiting Andy's office and wants to print from her tablet. She uses her tablet to discover available printers, and selects one listed. The printer is configured to limit access to only authorized users.

The printer challenges the tablet for authentication, and the tablet presents an authentication dialog to Lisa. Lisa doesn't have an account, but enters her email address and guesses at a password anyway. The printer rejects these credentials, and sends another challenge. Her tablet shows the authentication dialog again. Lisa clicks “Cancel” and looks for a different printer.

3.3.2. Authorization Policy Limits Access To Print Features

Harry is an intern who works at Andy's office, and he wants to print some photos from his laptop. He uses his laptop to discover available printers, and selects one listed. The printer is configured to allow only authorized users to print in color, and interns are not authorized to use this feature. His laptop has a modern IPP Client that supports the IPP Get-User-Printer-Attributes operation, so features that he isn't allowed to use will not be listed in the print dialog.

When he selects the printer, the laptop sends the Get-User-Printer-Attributes IPP operation to request the list of authorized features available to Harry's account. The printer responds

to the laptop with an authentication challenge. The laptop has stored single sign-on credentials, so it uses those to avoid bothering its user with a distraction. The printer accepts these credentials, and provides the list of features his account is authorized to use. The laptop shows this set of features. Harry is disappointed that he cannot print in color, so he abandons trying to print the photos because he doesn't want black-and-white prints.

3.4. Out of Scope

The following are considered out of scope for this document:

1. Definition of new HTTP authentication methods.
2. Definition of how specific authorization mechanisms are used by an IPP Printer.
3. Definition of how authentication information is used for access control.
4. Definition of how Printers are configured to use a particular authorization mechanism.

3.5. Design Requirements

The design requirements for this IPP Authentication Methods Best Practice document are to:

1. Illustrate how each authentication method integrates into IPP interaction flows; and
2. Describe implementation considerations for authentication in general, including factors that influence user experience and technical protocol concerns.

4. Client Authentication Methods

Authentication is the process of establishing some level of trust that an entity is who or what they are claiming to be. A Printer uses the “authenticated identity” or the “most authenticated user” [STD92] to determine whether to authorize the requesting Client to access requested capabilities such as operations, resources, and attributes. The Internet Printing Protocol/1.1 [STD92] defines authorization roles for end users, operators, and administrators, but does not define how a Printer or an authorization mechanism maps those roles to authenticated users.

A Printer specifies its supported authentication methods via several IPP attributes. The “uri-authentication-supported” attribute [STD92] indicates the authentication method used for a corresponding URI in “printer-uri-supported” [STD92]. The “xri-authentication” member attribute of “printer-xri-supported” [RFC3380] specifies the same corresponding values, if the Printer implements the “printer-xri-supported” attribute. Each of the authentication method keywords currently registered for “uri-authentication-supported” is described in its own subsection below. Some authentication methods may have additional IPP attributes associated with them. (Note: Configuration of the IPP Printer for use with a particular OAuth Authorization Server is out-of-scope for this document.)

One authentication & authorization system not described in this document is SAML (Security Assertion Markup Language) [SAMLCORE]. As of this writing, none of the standard SAML bindings to HTTP directly support IPP. OAuth 2.0 can indirectly support SAML via a SAML / OAuth 2.0 gateway. The gateway typically uses the SAML 2.0 assertion as an OAuth 2.0 Bearer token. Specific instructions for how to configure this depends on the SAML and OAuth 2.0 system implementations, and as with other infrastructure topics is beyond the scope of this document.

4.1. The 'none' IPP Authentication Method

The 'none' IPP Authentication Method [STD92] indicates that the receiving Printer provides no method to accept an asserted identity for the User operating the Client. The user name for the operation is assumed to be 'anonymous'. This authentication method is not recommended unless the Printer's operator intends to provide an anonymous print service.

Figure 4.1 illustrates how the 'none' authentication method integrates into an IPP operation request / response exchange.

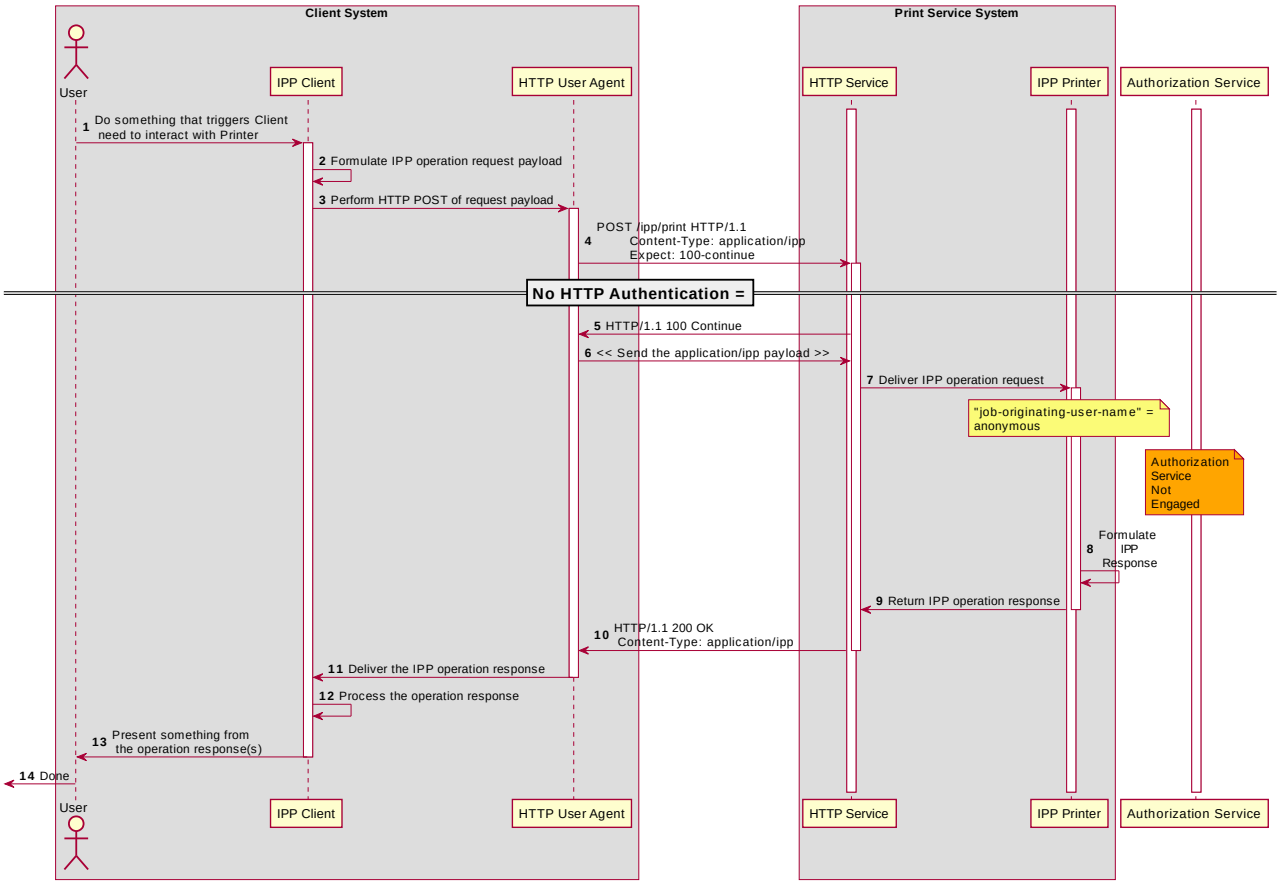

Figure 4.1: Sequence diagram for the 'none' IPP Authentication Method

4.2. The 'requesting-user-name' IPP Authentication Method

The 'requesting-user-name' IPP Authentication Method [STD92] indicates that the Client will provide the “requesting-user-name” operation attribute [STD92] in its IPP operation request. The Printer uses this unauthenticated name as the identity of the User operating the Client. This method is not recommended if job accounting or access authorization is important, since the Printer does not challenge the Client to prove the identity claimed in the “requesting-user-name”. Also, some Clients always send a fixed identity name (e.g. “mobile”) as a privacy defense when sending requests. As such, from a Printer's perspective, this method is increasingly undependable.

Figure 4.2 illustrates how the 'requesting-user-name' authentication method integrates into an IPP operation request / response exchange.

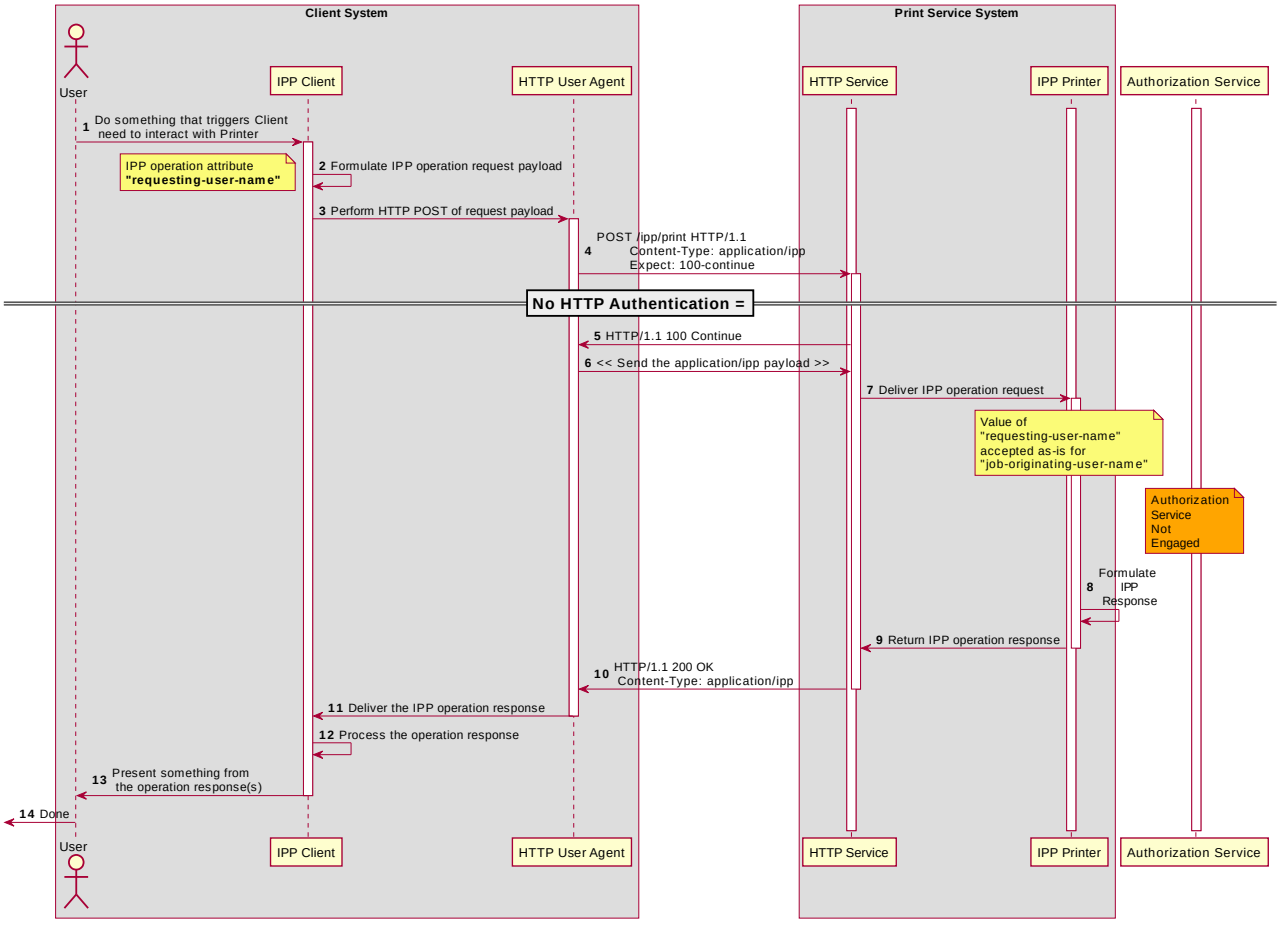

Figure 4.2: Sequence diagram for the 'requesting-user-name' IPP Authentication Method

4.3. The 'basic' IPP Authentication Method

The 'basic' IPP Authentication Method uses the HTTP Basic authentication scheme [RFC7617]. It is employed in IPP in much the same way as in conventional HTTP workflows using a Web browser. When the IPP Client receives an HTTP 401 Unauthorized response status and the “WWW-Authenticated” header in that response specifies 'Basic', a supporting Client will present UI asking the User to provide a user name and password. The Client will re-submit the IPP operation request to the HTTP Server providing access to the IPP Printer, including the “Authorization” HTTP header field with the provided credentials encoded in the format defined for the 'Basic' authentication method [RFC7617]. If the HTTP Server accepts that set of credentials, the IPP Printer authorizes access to the requested IPP operation and attributes for that account, and will respond accordingly.

Figure 4.3 illustrates how the 'basic' authentication method integrates into an IPP operation request / response exchange.

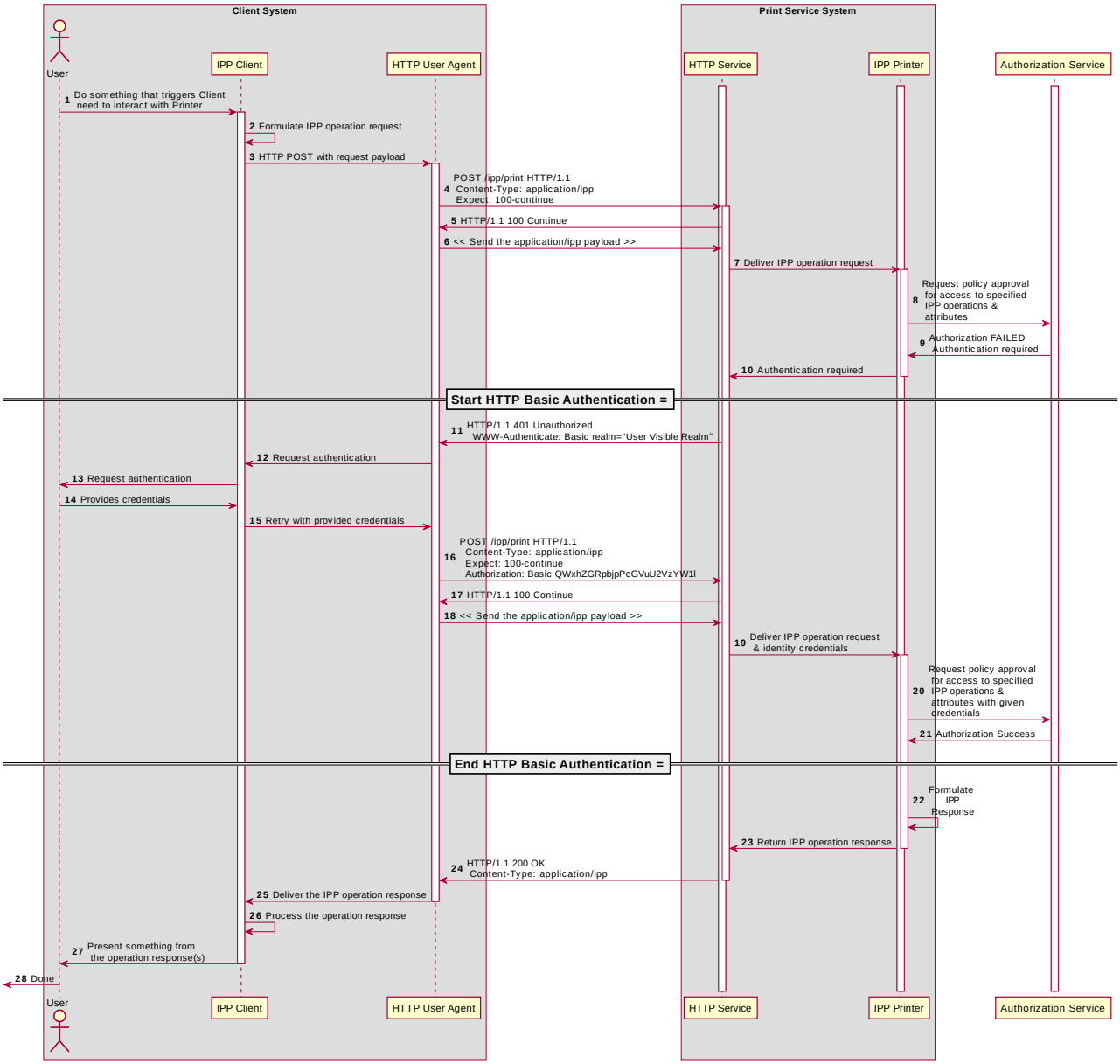

Figure 4.3: Sequence diagram for the 'basic' IPP Authentication Method

4.4. The 'digest' IPP Authentication Method

The 'digest' IPP Authentication method uses the HTTP Digest authentication scheme [RFC7616]. It is employed in IPP in much the same way as in conventional HTTP workflows using a Web browser. When the IPP Client receives an HTTP 401 Unauthorized response status and the “WWW-Authenticated” header in that response specifies 'Digest', a supporting Client will present UI asking the User to provide a user name and password. The Client will re-submit the IPP operation request to the HTTP Server providing access to the IPP Printer, including the “Authorization” HTTP header field with the provided credentials encoded in the format defined for the 'Digest' authentication method [RFC7616]. If the HTTP Server accepts that set of credentials, the IPP Printer authorizes access to the requested IPP operation and attributes for that account, and will respond accordingly.

Figure 4.4 illustrates how the 'digest' authentication method integrates into an IPP operation request / response exchange.

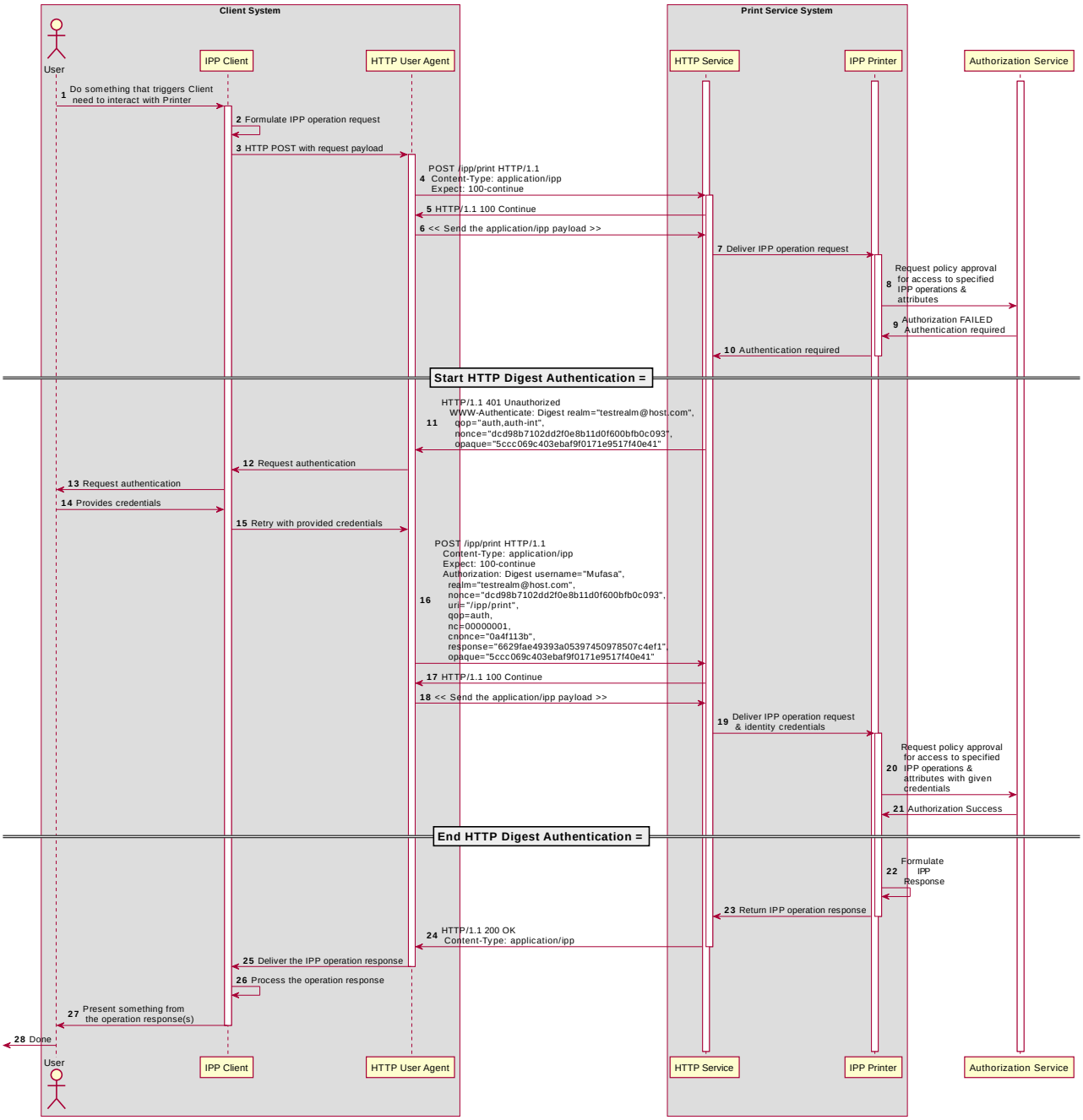

Figure 4.4: Sequence diagram for the 'digest' IPP Authentication Method

4.5. The 'negotiate' IPP Authentication Method

The 'negotiate' IPP Authentication method uses the HTTP Negotiate authentication scheme [RFC4559], which is used to support Kerberos and NTLM authentication methods with HTTP. It is employed in IPP in much the same way as in conventional HTTP workflows using a Web browser. When the IPP Client receives an HTTP 401 Unauthorized response status and the “WWW-Authenticated” header in that response specifies 'Negotiate', a supporting Client will present UI asking the User to provide a user name and password. The Client will re-submit the IPP operation request to the HTTP Server providing access to the IPP Printer, including the “Authorization” HTTP header field with the provided credentials encoded in the format defined for the 'Negotiate' authentication method [RFC4559]. If the HTTP Server accepts that set of credentials, the IPP Printer authorizes access to the requested IPP operation and attributes for that account, and will respond accordingly.

Figure 4.5 illustrates how the 'negotiate' authentication method integrates into an IPP operation request / response exchange.

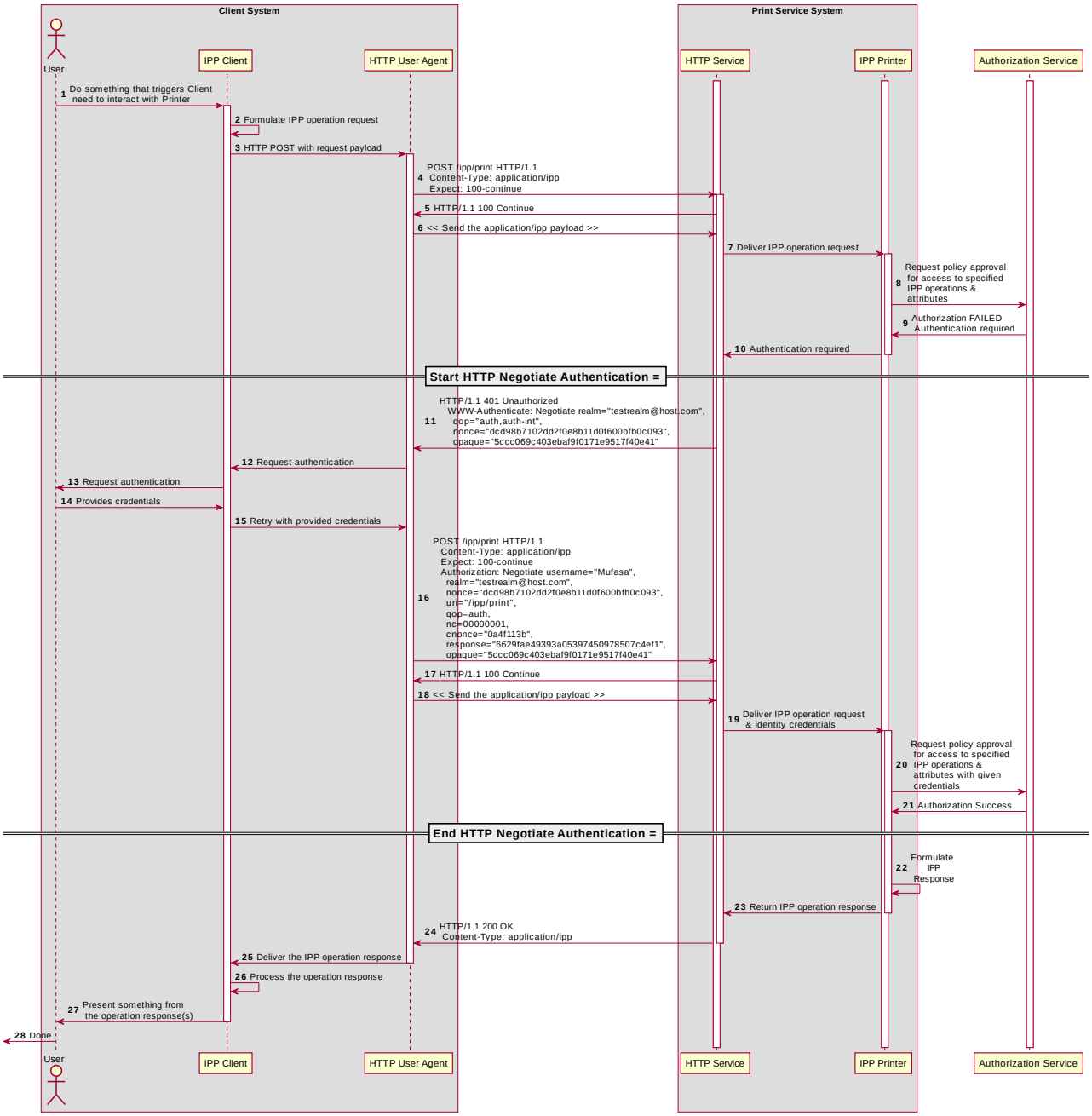

Figure 4.5: Sequence diagram for the 'negotiate' IPP Authentication Method

4.6. The 'oauth' IPP Authentication Method

The 'oauth' IPP Authentication method pertains to OAuth 2.0, which uses:

- The OAuth 2.0 authentication scheme [RFC6749], which defines the OAuth 2.0 system, authentication protocol framework, and OAuth 2.0 access tokens, which represents the scope, duration, and other attributes of an authorization grant;
- The OAuth 2.0 Bearer Token [RFC6750] which specifies the ways that an OAuth 2.0 access token can be encoded into general purpose HTTP requests and responses as an HTTP Bearer Token;
- The OAuth 2.0 Authentication Server Metadata [RFC8414] which provides the necessary metadata for interoperability;
- The OAuth 2.0 Dynamic Client Registration Protocol [RFC7591] which allows an IPP Client to register its local redirection URI with the Authorization Server; and
- OAuth 2.0 Token Introspection [RFC7662] which allows an IPP Printer to query information about a Bearer token provided by the IPP Client, including the list of granted scopes.

When the IPP Client receives an HTTP 401 Unauthorized response status, and the “WWW-Authenticated” header in that response specifies 'Bearer', a supporting Client will initiate the OAuth 2.0 flow by presenting a web view UI directed at the URL specified by the Printer's “oauth-authorization-server-uri” Printer Description attribute [PWG5100.18] and using the scope list specified by the Printer's "oauth-authorization-scope" Printer Description Attribute [IPP20190521]. Once the Client has acquired an OAuth 2.0 Access Token, it will encode that in the Bearer Token format and re-submit the IPP operation to the IPP Printer, including the “Authorization” HTTP header field with the provided credentials encoded in the OAuth 2.0 Bearer Token format [RFC6750]. If the HTTP Server accepts that set of credentials, the IPP Printer authorizes access to the requested IPP operation and attributes for that account, and will respond accordingly.

OAuth 2.0 is an authorization service framework that uses one or more authentication services, such as SAML 2.0 [SAMLCORE]. Figure 4.6 illustrates how the 'oauth' authentication method integrates into an IPP operation request / response exchange.

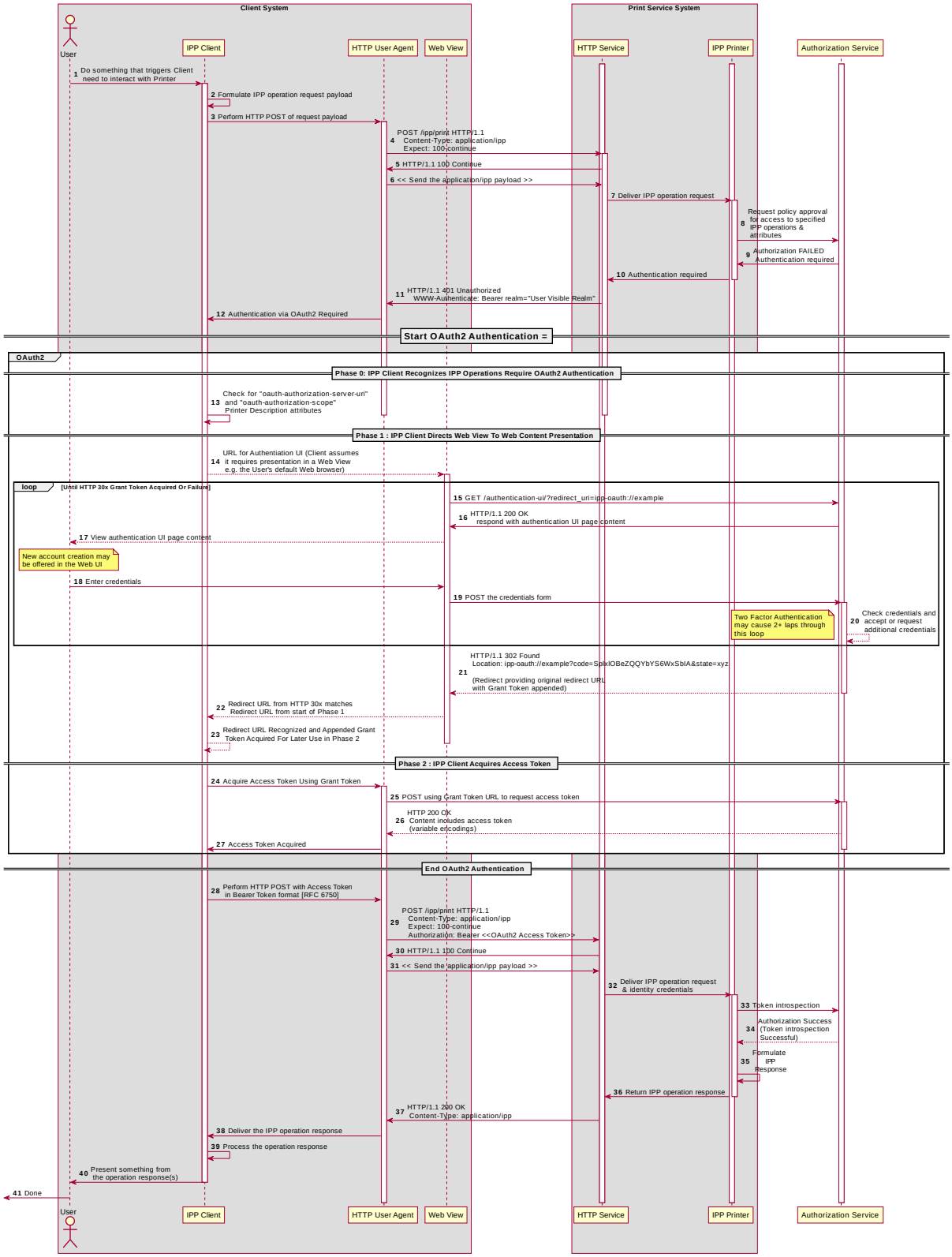

Figure 4.6: Sequence diagram for the 'oauth' IPP Authentication Method

4.7. The 'certificate' IPP Authentication Method

The 'certificate' IPP Authentication method uses X.509 certificate authentication via TLS [RFC8446]. This authentication method is initiated by the Printer when it sends a TLS Certificate Request message during the Transport Layer Security (TLS) handshake. The Client responds by sending a TLS Certificate message with the X.509 certificate identifying the User and/or Client. The Client then sends a TLS Certificate Verify message to prove to the Printer that the Client has the corresponding private key. If the Client has no X.509 certificate to provide to the Printer, it sends an empty TLS Certificate message.

The Printer SHOULD allow both empty and valid X.509 certificates. The Printer SHOULD return the IPP status code listed in Table 4.1 when the corresponding authentication exception occurs. The Client SHOULD respond to the reported status code with the corresponding response listed in Table 4.1.

Operation Status Code	**Authentication Exception**	**Recommended Client Response**
'client-error-not-authenticated'	Authentication required but no X.509 certificate supplied	Close the connection; select a certificate (with possible user interaction); retry connection with selected certificate
'client-error-not-authorized'	Access denied for the identity specified by the provided X.509 certificate; try again	Close the connection; select a different certificate (with possible user interaction); retry connection with selected certificate
'client-error-forbidden'	Access denied for the identity specified by the provided X.509 certificate; don't try again	Close the connection and present User with error dialog (“Access denied”)

Table 4.1: IPP 'certificate' Authentication Method Error Condition Status Codes

Figure 4.7 illustrates how the TLS authentication method integrates into an IPP operation request / response exchange.

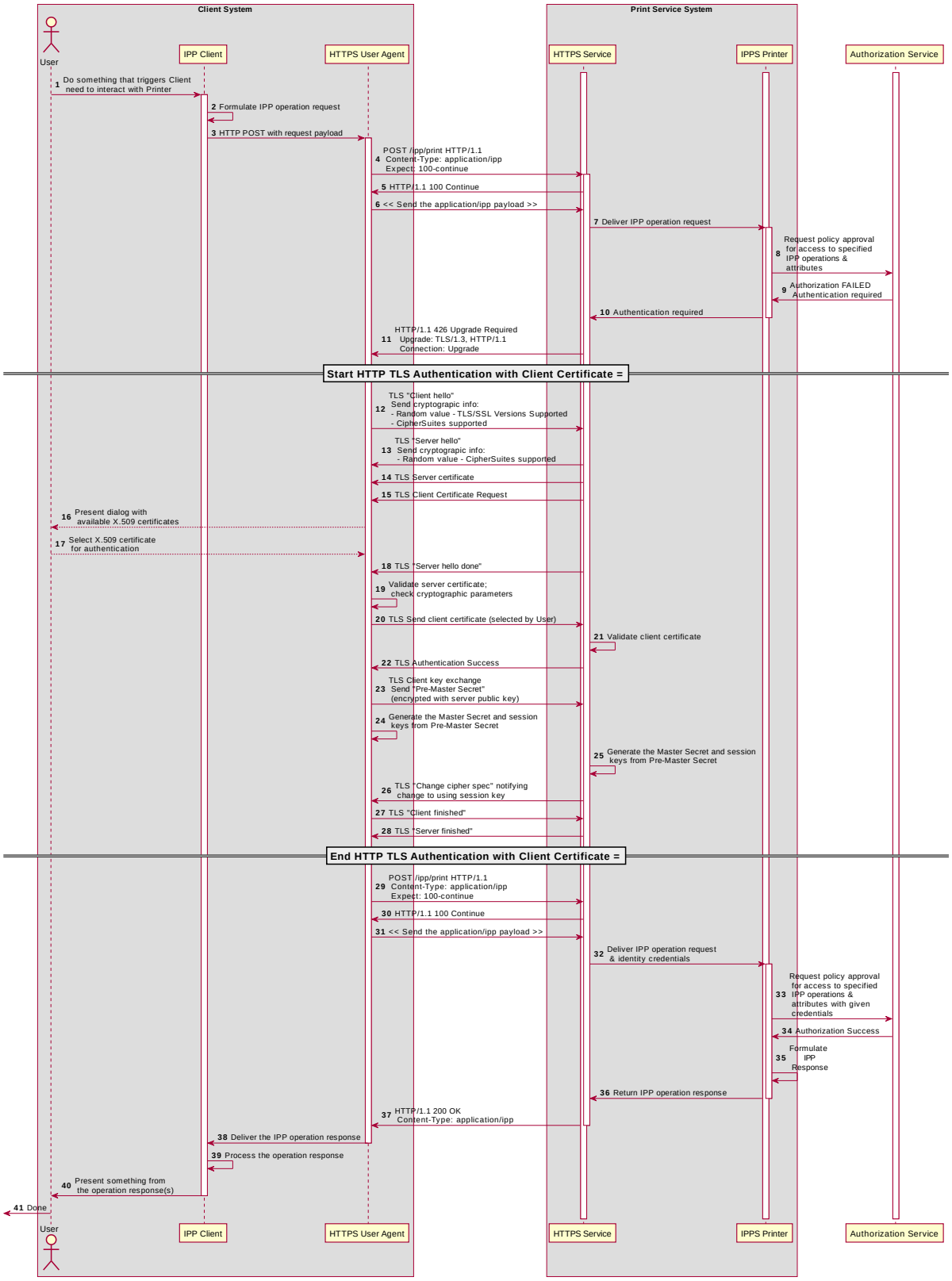

Figure 4.7: Sequence diagram for the 'certificate'' IPP Authentication Method

5. Implementation Recommendations

5.1. Client Implementation Recommendations

5.1.1. General Recommendations

A Client SHOULD limit the number of additional windows presented to the user during the course of an authentication workflow, to avoid causing a fragmented, disruptive user experience.

Since some tasks require multiple IPP operations, a Client SHOULD store non-persistent authentication credentials for reuse in later IPP operations for the duration of that task.

Client security considerations (section 7.2) should also be followed.

5.1.2. Handling Authentication Failure

A Client that encounters an authentication failure SHOULD offer the User another opportunity to provide valid authentication credentials and SHOULD abandon new attempts when the User rejects the offer for different credentials (e.g. by clicking on a “Cancel” button in an authentication dialog window). For HTTP authentication, the Client will receive an HTTP 401 Unauthorized response. For TLS authentication, the Client will receive an HTTP 200 OK with an IPP message body with status code 'client-error-not-authorized' [STD92].

5.1.3. Handling Authorization Failure

A Client that encounters an authorization failure SHOULD abandon communications with the target Printer because, while the credentials are recognized and authenticated, the identity corresponding to those valid credentials is not authorized to proceed. For HTTP authentication, the Client will receive an HTTP 403 Forbidden response. For TLS authentication, the Client will receive an HTTP 200 OK with an IPP message body with status code 'client-error-forbidden' [STD92].

5.1.4. OAuth 2.0 Recommendations

The Client that supports Resource Owner Grants (username and password) SHOULD otherwise follow the guidelines laid out in current OAuth 2.0 best practices including “Proof Key for Code Exchange by OAuth Public Clients” [RFC7636], “OAuth 2.0 for Native Apps” [RFC7636] and “OAuth 2.0 Security Best Current Practice” [OAUTH2SECBP].

5.2. Printer Implementation Recommendations

5.2.1. General Recommendations

The Printer or the Job might also need to store a token or identifier (UUID, JWT, etc.) that represents the User's authenticated identity or authentication session, in cases where the Printer depends on an external authorization service for print policy evaluation. This token is considered by IPP to be an internal implementation detail, and the Printer never provides Clients access to these tokens via IPP, as discussed in RFC 8011 section 5.3.6 [STD92].

When handing an IPP Job Creation request, the Printer will also need to populate the Job's “job-originating-user-name” Job Status attribute. In cases where the Printer relies upon an external authentication service, it will need to acquire a meaningfully printable value from the authentication service.

Client security considerations (section 7.3) should also be followed.

5.2.2. Handling Authentication Failure

If a Printer receives an IPP operation request, challenges the Client for authentication using one of the methods described in this document, and the credentials are invalid, how the Printer reports the authentication failure depends on the authentication method. For HTTP authentication, the Printer returns an HTTP 401 Unauthorized response. For TLS authentication, the Printer returns an HTTP 200 OK with an IPP message body specifying a 'client-error-not-authorized' status code [STD92].

5.2.3. Handling Authorization Failure

If a Printer receives an IPP operation request, and the Client credentials have been authenticated, but the identity corresponding to the credentials is not authorized to use the Printer or the operations or attributes specified in the request, how the Printer reports the authorization failure depends on the authentication method. For HTTP authentication, the Printer returns an HTTP 403 Forbidden response. For TLS authentication, the Printer returns an HTTP 200 OK with an IPP message body specifying a 'client-error-forbidden' status code [STD92].

5.2.4. HTTP Digest Recommendations

A Printer SHOULD NOT invalidate any HTTP Digest parameters (nonce, etc.) in the middle of an IPP operation request. Especially in the case of operations that are streaming document data (Print-Job, Send-Document), the data stream might not be cacheable by the Client, and this can cause a significant burden to the Client, degrade the user experience, or cause the operation to fail. Once a Printer has received a Job Creation operation request or a Validate-Job operation request, it SHOULD NOT change the nonce

used for HTTP Digest authentication until the Job Submission operations for that Job have concluded.

5.2.5. OAuth 2.0 Recommendations

A Printer deployed in an OAuth 2.0 environment SHOULD follow current OAuth 2.0 best practices including “Proof Key for Code Exchange by OAuth Public Clients” [RFC7636], “OAuth 2.0 for Native Apps” [RFC7636] and “OAuth 2.0 Security Best Current Practice” [OAUTH2SECBP].

6. Internationalization Considerations

For interoperability and basic support for multiple languages, conforming implementations SHOULD support the Universal Character Set (UCS) Transformation Format -- 8 bit (UTF-8) [RFC3629] encoding of Unicode [UNICODE] [ISO10646] and the Unicode Format for Network Interchange [RFC5198].

Implementations of this specification SHOULD conform to the following standards on processing of human-readable Unicode text strings, see:

- Unicode Bidirectional Algorithm [UAX9] – left-to-right, right-to-left, and vertical
- Unicode Line Breaking Algorithm [UAX14] – character classes and wrapping
- Unicode Normalization Forms [UAX15] – especially NFC for [RFC5198]
- Unicode Text Segmentation [UAX29] – grapheme clusters, words, sentences
- Unicode Identifier and Pattern Syntax [UAX31] – identifier use and normalization
- Unicode Collation Algorithm [UTS10] – sorting
- Unicode Locale Data Markup Language [UTS35] – locale databases

Implementations of this specification are advised to also review the following informational documents on processing of human-readable Unicode text strings:

- Unicode Character Encoding Model [UTR17] – multi-layer character model
- Unicode in XML and other Markup Languages [UTR20] – XML usage
- Unicode Character Property Model [UTR23] – character properties
- Unicode Conformance Model [UTR33] – Unicode conformance basis

7. Security Considerations

7.1. Human-readable Strings

Implementations of this specification SHOULD conform to the following standard on processing of human-readable Unicode text strings, see:

- Unicode Security Mechanisms [UTS39] – detecting and avoiding security attacks

Implementations of this specification are advised to also review the following informational document on processing of human-readable Unicode text strings:

- Unicode Security FAQ [UNISECFAQ] – common Unicode security issues

7.2. Client Security Considerations

The following are the security recommendations for an IPP Client.

1. A Client SHOULD use the most secure authentication method supported by the Printer.

2. A Client SHOULD securely store at rest any personally identifiable information (PII) and authentication credentials such as passwords or session tokens.

3. A Client SHOULD only respond to an authentication challenge if it is conveyed over a secure connection [STD92] such as TLS, unless a secure connection is not supported over that transport (e.g. IPP USB [IPPUSB]).

4. A Client SHOULD validate the identity of the Printer by whatever means are available for that connection type. If the connection is secured via TLS [STD92], the Client SHOULD validate the server's TLS certificate, match it to the originating host, cross-check it to match the host name or IP address in the IPP URI for the target Printer, and otherwise follow industry best practices for validating the Printer's identity using X.509 certificates over TLS [RFC6125]. If the connection is not secured via TLS, other means could be necessary to validate the Printer's identity.

5. A Client SHOULD provide a means to allow the User to examine a Printer's provided identity.

6. A Client SHOULD provide one or more means of notification when it is engaging with a previously encountered Printer whose identity has changed.

7. A Client supporting OAuth 2.0 SHOULD conform to the recommendations in “Proof Key for Code Exchange by OAuth Public Clients” [RFC7636] and “OAuth 2 for Native Apps” [RFC8252] if the print system provides its own user interface

presentation and controls for handling the OAuth 2.0 authentication steps, to mitigate the risks described therein.

8. A Client SHOULD use the most secure authentication method available for a given Printer. In some cases, a Printer could support more than one authentication method for a particular URI. It can specify this by listing the same URI multiple times in its “printer-uri-supported” attribute [STD92], and specifying the different authentication methods in each of the corresponding values specified by its “uri-authentication-supported” attribute [STD92].

7.3. Printer Security Considerations

The following are the security recommendations for an IPP Printer.

1. A Printer SHOULD securely store at rest any personally identifiable information (PII) and authentication credentials such as passwords that are local to the Printer.

2. A Printer SHOULD only challenge a Client for authentication over a secure connection (TLS) [STD92] unless TLS is not supported over that transport (e.g. IPP USB [IPPUSB]).

3. A Printer SHOULD support Administrator-provisioned X.509 server certificates that persist across power cycles, and these certificates SHOULD NOT be automatically renewed or replaced.

4. A Printer SHOULD support self-generated self-signed X.509 certificates that persist across power cycles. The certificate SHOULD have a minimum default expiration of 5 years from the date of issuance / generation, SHOULD be automatically renewed (regenerated), using a new private key if the previous certificate has expired, SHOULD be generated using the mDNS, DHCP and/or manually-configured DNS hostname(s) and regenerated whenever these change, and SHOULD comply with the recommendations from the CA/Browser Forum [CABCORE] relating to, among other things, the set of cryptographic primitives, algorithms and key lengths to use to produce the certificate.

5. In cases where the Printer supports more than one authentication method for a particular URI, the Printer SHOULD specify the alternative authentication schemes by listing the same URI multiple times in its “printer-uri-supported” attribute [STD92] and specifying a different authentication method for each corresponding value in its “uri-authentication-supported” attribute [STD92].

6. A Printer supporting OAuth 2.0 SHOULD conform to the recommendations in “Proof Key for Code Exchange by OAuth Public Clients” [RFC7636] and “OAuth 2 for Native Apps” [RFC8252] to mitigate the risks described therein.

8. References

8.1. Normative References

[ISO10646] "Information technology -- Universal Coded Character Set (UCS)", ISO/IEC 10646:2011

[OAUTH2SECBP] T. Lodderstedt, J. Bradley, A. Labunets, D. Fett, “OAuth 2.0 Security Best Current Practice”, https://tools.ietf.org/html/draft-ietf-oauth-security-topics

[RFC3629] F. Yergeau, “UTF-8, a transformation format of ISO 10646”, RFC 3629, November 2003, https://tools.ietf.org/html/rfc3629

[RFC5198] J. Klensin, M. Padlipsky, “Unicode Format for Network Interchange”, RFC 5198, March 2008, https://tools.ietf.org/html/rfc5198

[RFC7636] N. Sakimura, Ed., J. Bradley, N. Agarwal, "Proof Key for Code Exchange by OAuth Public Clients", RFC 7636, September 2015, https://tools.ietf.org/html/rfc7636

[RFC8252] W. Denniss, J. Bradley, “OAuth 2.0 for Native Apps”, RFC 8252, October 2017, https://tools.ietf.org/html/rfc8252

[STD92] M. Sweet, I. McDonald, "Internet Printing Protocol/1.1", STD 92, June 2018, https://tools.ietf.org/html/std92

[UAX9] Unicode Consortium, “Unicode Bidirectional Algorithm”, UAX#9, May 2016, http://www.unicode.org/reports/tr9

[UAX14] Unicode Consortium, “Unicode Line Breaking Algorithm”, UAX#14, June 2016, http://www.unicode.org/reports/tr14

[UAX15] Unicode Consortium, “Normalization Forms”, UAX#15, February 2016, http://www.unicode.org/reports/tr15

[UAX29] Unicode Consortium, “Unicode Text Segmentation”, UAX#29, June 2016, http://www.unicode.org/reports/tr29

[UAX31] Unicode Consortium, “Unicode Identifier and Pattern Syntax”, UAX#31, May 2016, http://www.unicode.org/reports/tr31

[UNICODE] The Unicode Consortium, “Unicode® 10.0.0”, June 2017, http://unicode.org/versions/Unicode10.0.0/

[UTS10] Unicode Consortium, “Unicode Collation Algorithm”, UTS#10, May 2016, http://www.unicode.org/reports/tr10

[UTS35] Unicode Consortium, “Unicode Locale Data Markup Language”, UTS#35, October 2016, http://www.unicode.org/reports/tr35

[UTS39] Unicode Consortium, “Unicode Security Mechanisms”, UTS#39, June 2016, http://www.unicode.org/reports/tr39

8.2. Informative References

[CABCORE] CA/Browser Forum, “Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates”, Version 1.6.1, October 2018, https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-1.6.1.pdf

[IPP20190521] M. Sweet, “RFC: oauth-authorization-scope (1setOf name(MAX)) Printer Description attribute”, PWG IPP Email Registration, May 2019, https://ftp.pwg.org/pub/pwg/ipp/registrations/ippwg-oauth-authorization-scope-20190521.txt

[IPPUSB] S. Kennedy, A. Mitchell, “USB Print Interface Class IPP Protocol Specification”, December 2012, http://www.usb.org/developers/docs/devclass_docs/IPP.zip

[ITUX.800] ITU, "ITU-T Recommendation X.800 (03/91), Security architecture for Open Systems Interconnection for CCITT applications", March 1991, https://www.itu.int/rec/T-REC-X.800-199103-I

[PWG5100.18] M. Sweet, I. McDonald, “IPP Shared Infrastructure Extensions”, 5100.18-2015, June 2015, https://ftp.pwg.org/pub/pwg/candidates/cs-ippinfra10-20150619-5100.18.pdf

[RFC3380] T. Hastings, R. Herriot, C. Kugler, H. Lewis, “Internet Printing Protocol (IPP): Job and Printer Set Operations”, RFC 3380, September 2002, https://tools.ietf.org/html/rfc3380

[RFC4559] K. Jaganathan, L. Zhu, J. Brezak, “SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows”, RFC 4559, June 2006, https://tools.ietf.org/html/rfc4559

[RFC6125] P. Saint-Andre, J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011, https://tools.ietf.org/html/rfc6125

[RFC6749] D. Hardt, Ed., “The OAuth 2.0 Authorization Framework”, RFC 6749, October 2012, https://tools.ietf.org/html/rfc6749

[RFC6750] M. Jones, D. Hardt, “The OAuth 2.0 Authorization Framework: Bearer Token Usage”, RFC 6750, October 2012, https://tools.ietf.org/html/rfc6750

[RFC7230] R. Fielding, J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, June 2014, https://tools.ietf.org/html/rfc7230

[RFC7235] R. Fielding, J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Authentication", RFC 7235, June 2014, https://tools.ietf.org/html/rfc7235

[RFC7591] J. Richer, M. Jones, J. Bradley, M. Machulak, P. Hunt, “OAuth 2.0 Dynamic Client Registration Protocol”, RFC 7591, July 2015, https://tools.ietf.org/html/rfc7591

[RFC7616] R. Shekh-Yusef, D. Ahrens, S. Bremer, “HTTP Digest Access Authentication”, RFC 7616, September 2015, https://tools.ietf.org/html/rfc7617

[RFC7617] J. Reschke, “The 'Basic' HTTP Authentication Scheme”, RFC 7617, September 2015, https://tools.ietf.org/html/rfc7617

[RFC7662] J. Richer, Ed., “OAuth 2.0 Token Introspection”, RFC 7662, October 2015, https://tools.ietf.org/html/rfc7662

[RFC8414] M. Jones, N. Sakimura, J. Bradley, “OAuth 2.0 Authorization Server Metadata”, RFC 8414, June 2018, https://tools.ietf.org/html/rfc8414

[RFC8446] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3”, RFC 8446, August 2018, https://tools.ietf.org/html/rfc8446

[SAMLCORE] S. Cantor et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, 15 March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[UNISECFAQ] Unicode Consortium “Unicode Security FAQ”, November 2016, http://www.unicode.org/faq/security.html

[UTR17] Unicode Consortium “Unicode Character Encoding Model”, UTR#17, November 2008, http://www.unicode.org/reports/tr17

[UTR20] Unicode Consortium “Unicode in XML and other Markup Languages”, UTR#20, January 2013, http://www.unicode.org/reports/tr20

[UTR23] Unicode Consortium “Unicode Character Property Model”, UTR#23, May 2015, http://www.unicode.org/reports/tr23

[UTR33] Unicode Consortium “Unicode Conformance Model”, UTR#33, November 2008, http://www.unicode.org/reports/tr33

9. Authors' Addresses

Primary authors:

Smith Kennedy
HP Inc.
11311 Chinden Blvd.
Boise ID 83714

Michael Sweet
Apple Inc.
One Apple Park Way
MS 111-HOMC
Cupertino, CA 95014

Send comments to the PWG IPP Mailing List:

ipp@pwg.org (subscribers only)

To subscribe, see the PWG web page:

https://www.pwg.org/

Implementers of this specification document are encouraged to join the IPP Mailing List in order to participate in any discussions of clarification issues and review of registration proposals for additional attributes and values.

The authors would also like to thank the following individuals for their contributions to this standard:

Ira McDonald – High North, Inc.
William Wagner – TIC Inc.
Sean Kau – Google Inc.

