IPP Authentication Methods

(IPPAUTH)

Abstract: This best practice document provides implementation guidance on how to best integrate the various authentication mechanisms used over IPP's HTTP and HTTPS transports into IPP protocol exchanges and the design of authentication user experiences on IPP Client systems.

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


This document is available electronically at:


Copyright © 2017-2019 The Printer Working Group. All rights reserved.

Title: IPP Authentication Methods (IPPAUTH)
The material contained herein is not a license, either expressed or implied, to any IPR owned or controlled by any of the authors or developers of this material or the Printer Working Group. The material contained herein is provided on an “AS IS” basis and to the maximum extent permitted by applicable law, this material is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material and the Printer Working Group and its members hereby disclaim all warranties and conditions, either expressed, implied or statutory, including, but not limited to, any (if any) implied warranties that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.
1. Introduction

The Internet Printing Protocol (hereafter, IPP) uses HTTP as its underlying transport [RFC8010]. 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 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 common 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 implementers, Printer implementers, and Infrastructure Administrators when implementing support for one of these different 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

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/


3. Requirements

3.1. Rationale

Given the following existing specifications:

1. Internet Printing Protocol/1.1: Encoding and Transport [RFC8010] and Internet Printing Protocol/1.1: Model and Semantics [RFC8011] define the core Internet Printing Protocol/1.1 IETF STD 92

2. RFC 7617 defines the 'Basic' HTTP Authentication Scheme

3. RFC 7616 defines HTTP Digest Access Authentication

4. RFC 4559 defines SPNEGO-based Kerberos and NTLM HTTP Authentication

5. RFC 6749 defines the OAuth 2.0 Authorization Framework

6. RFC 8252 describes best practices for OAuth 2.0 for Native Apps

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 user experience provided by an IPP Client

3.2. Use Cases

3.2.1. Authenticated Printing

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 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 user interface elements corresponding to the HTTP authentication type. Andy enters his credential to prove access, and the Printer approves access. The laptop then provides the usual print user interface allowing Andy to select print options.

3.3. Exceptions

3.3.1. Authentication Failure Prevents Access

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. She 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 Failure Prevents Access

Harry is visiting Andy’s office and wants to print from his tablet. He uses his 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 Harry. He doesn’t have an account, but enters his email address and guesses at a password anyway. The printer rejects these credentials, and sends another challenge. His tablet shows the authentication dialog again. Harry clicks “Cancel” and looks for a different printer.

3.4. Out of Scope

The following are considered out of scope for this document:

1. Definition of new HTTP authentication methods
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” [RFC8011] to determine whether to allow the requesting Client access to capabilities such as operations, resources, and attributes. A Printer specifies its supported authentication methods via several IPP attributes. The “uri-authentication-supported” attribute [RFC8011] indicates the authentication method used for a corresponding URI in “printer-uri-supported” [RFC8011]. 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.

In cases where the Printer is not directly involved in the authentication process, such as when OAuth2 is used, or when the Printer depends on an external authentication service, the Printer might not be directly aware of the User's identity following authentication. In these cases, the Printer could still need to acquire the User's identity in order to accurately document the User's identity in the Job Object's Job Status attributes, or to support IPP operations such as Get-User-Printer-Attributes [IPPGUPA] that depend on the User's identity to provide meaningfully filtered operation responses.

One authentication system not described below is SAML (Security Assertion Markup Language)[SAMLCORE]. As of this writing, none of the standard SAML bindings to HTTP directly support IPP. SAML can indirectly support OAuth2 via a SAML / OAuth2 gateway. The bridge 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 OAuth2 system implementations, and is beyond the scope of this document.
4.1. The 'none' IPP Authentication Method

The 'none' IPP Authentication Method [RFC8011] very simply indicates that the receiving Printer is provided no method whatsoever to determine the identity of the User who is operating the Client that is making IPP operation requests. The user name for the operation is assumed to be 'anonymous'. This method is not recommended unless the Printer's operator has the objective of providing an anonymous print service. In most cases, the Client SHOULD provide the "requesting-user-name" operation attribute, as described in section 4.2.

Figure 4.1 illustrates how the 'none' authentication method integrates into an IPP operation request / response exchange. Other authentication methods will expand on this baseline request / response exchange.
4.2. The 'requesting-user-name' IPP Authentication Method

In the 'requesting-user-name' IPP Authentication Method [RFC8011], the Client MUST provides the "requesting-user-name" operation attribute [RFC8011] in its IPP operation request. The Printer uses this unauthenticated name as the identity of the actor operating the Client. This method is not recommended since there is no actual authentication performed as there is no credential provided to prove the identity claimed in the "requesting-user-name".

Figure 4.2 illustrates how the 'requesting-user-name' authentication method integrates into an IPP operation request / response exchange. This is basically identical to the 'none' method from a protocol perspective.
4.3. The 'basic' IPP Authentication Method

The 'basic' IPP Authentication Method uses HTTP Basic authentication scheme [RFC7617]. It is employed in IPP in much the same way that it is employed in conventional HTTP workflows using a Web browser. When the IPP Client encounters an HTTP 401 Unauthorized response, it evaluates whether it supports the authentication method identified by the value of the "WWW-Authenticated" header in the response. In this case, if it supports 'basic', it will present UI asking the User to provide username and password credentials that could be used to authenticate with the HTTP Server providing access to the IPP Printer. If the HTTP Server successfully authenticates that set of credentials, then the IPP operation request is passed on to the IPP Printer, which responds as usual.

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

![Sequence diagram for the 'basic' IPP Authentication Method](image-url)
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 that it is employed in conventional HTTP workflows using a Web browser; when the IPP Client encounters an HTTP 401 Unauthorized response, it evaluates whether it supports the authentication method identified by the value of the “WWW-Authenticated” header in the response. In this case, if it supports 'digest', it will present UI asking the User to provide username and password credentials that might be used to authenticate with the HTTP Server providing access to the IPP Printer. If the HTTP Server successfully authenticates that set of credentials, then the IPP operation request is passed on to the IPP Printer, which responds as usual.

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

Figure 4.5 illustrates how the 'negotiate' authentication method integrates into an IPP operation request / response exchange.
4.6. The 'oauth' IPP Authentication Method

The 'oauth' IPP Authentication method pertains to OAuth2, which uses:

- the OAuth2 authentication scheme [RFC6749], which provides...
- The OAuth2 Bearer Token [RFC6750] which provides...
- The OAuth2 Authentication Server Metadata [RFC8414] which provides the necessary metadata for interoperability.

OAuth 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, which depends on the Printer supporting the "oauth-authorization-server-uri" Printer Description attribute [PWG5100.18].
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. X.509 certificate authentication via TLS is initiated by the Printer by sending a Certificate Request message during the Transport Layer Security (TLS) [RFC5246] handshake. The Client then sends the X.509 certificate identifying the User and/or Client in a corresponding Certificate message, and a subsequent Certificate Verify message to prove to the Printer that the Client has the corresponding private key. If the Client has no configured X.509 certificate to provide, it sends an empty 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.

<table>
<thead>
<tr>
<th>Operation Status Code</th>
<th>Authentication Exception</th>
<th>Recommended Client Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>'client-error-not-authenticated'</td>
<td>Authentication required but no X.509 certificate supplied</td>
<td>Close the connection; select a certificate (with possible user interaction); retry connection with selected certificate</td>
</tr>
<tr>
<td>'client-error-not-authorized'</td>
<td>Access denied for the identity specified by the provided X.509 certificate; try again</td>
<td>Close the connection; select a different certificate (with possible user interaction); retry connection with selected certificate</td>
</tr>
<tr>
<td>'client-error-forbidden'</td>
<td>Access denied for the identity specified by the provided X.509 certificate; don't try again</td>
<td>Close the connection and present User with error dialog (&quot;Access denied&quot;)</td>
</tr>
</tbody>
</table>

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.
Do something that triggers Client
need to interact with Printer

1. Formulate IPP operation request
2. HTTP POST with request payload

3. Send cryptograpic info:
   - Random value
   - TLS/SSL Versions Supported
   - CipherSuites supported

4. Start HTTP TLS Authentication with Client Certificate

5. TLS “Server hello”
   Send cryptograpic info:
   - Random value
   - CipherSuites supported

6. TLS Server certificate

7. TLS Client certificate request

8. Present dialog with available X.509 certificates

9. Select and approve X.509 certificate

10. TLS “Server hello done”
    These additional steps in TLS negotiation
    provide client certificate authentication.

11. Validate server certificate;
    check cryptographic parameters

12. TLS Send client certificate (selected by User)

13. Validate client certificate

14. TLS Authentication Success

15. TLS Client key exchange

16. Generate the Master Secret and session keys
    from Pre-Master Secret

17. TLS “Change cipher spec” notifying change to using session key

18. TLS “Handshake end”

19. TLS “Server finished”

20. End HTTP TLS Authentication with Client Certificate

21. POST /ipp/print HTTP/1.1
    Content-Type: application/ipp
    Expect: 100-continue

22. HTTP/1.1 100 Continue

23. << Send the application/ipp payload >>

24. Deliver IPP operation request

25. Formulate IPP operation response

26. Return IPP operation response

27. HTTP/1.1 200 OK
    Content-Type: application/ipp

28. Deliver the IPP operation response

29. Process the operation response

30. Present something from the operation response(s)

31. Done

Figure 4.7: Sequence diagram for X.509 Certificate Authentication Via TLS
5. Implementation Recommendations

Provide possible technical solutions/approaches in this section. Include pros and cons for each technical solution or approach. Include references to specific protocols and/or data models when appropriate. Include mapping and gateway considerations when appropriate.

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.

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' [RFC8011].

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' [RFC8011].

5.1.4. OAuth2 Recommendations

The Client might support Resource Owner Grants (username and password) SHOULD otherwise follow the guidelines laid out in current OAuth2 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].

The OAuth2 authorization service might implement a complicated user presentation. If possible, the Client SHOULD select a presentation alternative that is the least complicated.
or the most similar to the user experience provided for older authentication methods (HTTP Basic or HTTP Digest) that might be more familiar to the user.

Printer Implementation Recommendations

5.1.5. General Recommendations

In some authentication topologies, the Printer is not directly involved in all phases of the authentication process. In these scenarios, the Printer could still need access to the User's identity for IPP level access authorization, Job accounting (e.g. the Job Object's Job Status attributes), or to support IPP operations such as Get-User-Printer-Attributes [IPPGUPA] that depend on the User's identity to provide meaningfully filtered operation responses. Distributed topologies SHOULD account for this need in their back-end integration with the Printer.

5.1.6. 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-authenticated' status code [RFC8011].

5.1.7. 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 [RFC8011].

5.1.8. 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.
5.1.9. OAuth2 Recommendations


An OAuth2 Authorization Service used in an IPP printing workflow SHOULD direct a Client to an authentication page that facilitates an appropriate presentation on even limited Client systems such as smartphones. In deployments where SAML [SAMLCORE] is used for network authentication and IPP authentication is needed, an OAuth2 / SAML gateway SHOULD be implemented to provide compatibility with IPP.

Internationalization Considerations

For interoperability and basic support for multiple languages, conforming implementations MUST 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
6. Security Considerations

6.1. Human-readable Strings

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


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

6.2. 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.

3. A Client SHOULD only respond to an authentication challenge over a secure connection (TLS) [RFC8010][RFC8011] unless TLS is not supported over that transport (e.g. IPP USB).

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 [RFC8010], 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 might 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 OAuth2 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 OAuth2 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, and specifying the different authentication methods in each of the corresponding values specified by its “uri-authentication-supported” attribute.

6.3. 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) [RFC8010][RFC8011] unless TLS is not supported over that transport (e.g. IPP USB).

3. A Printer MUST support User-provisioned X.509 certificates that persist across power cycles. These certificates MUST 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 MUST specify the alternative authentication schemes by listing the same URI multiple times in its “printer-uri-supported” attribute, and specifying a different authentication method for each corresponding value in its “uri-authentication-supported” attribute.

6. A Printer supporting OAuth2 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.
7. References

7.1. Normative References


7.2. Informative References


8. Authors' Addresses

Primary authors:

- Smith Kennedy
- HP Inc.
- 11311 Chinden Blvd.
- Boise ID 83714
- smith.kennedy@hp.com

- Michael Sweet
- Apple Inc.
9. Change History

9.1. January 17, 2019

Updated with live edits and feedback from the January 17 IPP WG meeting.

- Status changed to Stable in preparation for Changed all “might” to “could”
- Fixed all IETF RFC URLs to use the “https://tools.ietf.org/html/rfcXXX” format
- Changed the OAuth2 recommendations in sections 5.1.4 and 5.2.5 to simply point to best practice RFCs elsewhere.
- A few other minor editorial changes

9.2. January 16, 2019

Changed status to Prototype draft.

9.3. January 9, 2019

Added mention of “oauth-authorization-server-uri” and reference to 5100.18 in section 4.6 since it is mentioned in the sequence diagram.

9.4. January 7, 2019

- Minor editorial fixes to section 4.
- Editorial fixes to section 3.3.2

9.5. December 22, 2018

Updated with changes and feedback from review in November 2018 PWG F2F:
• Updated exception cases in section 3.3 to delineate authorization and authentication failure exception cases

• Restored all UML diagrams to their previous state, removing the authentication and authorization failure cases

• Rewrote recommendations in section 5.

9.6. November 9, 2018

Updated as per IPP WG review feedback from 2018-10-25:

• Added discussion of SAML 2.0 in appropriate locations in section 4 and 4.7, and added an informative reference to the OASIS SAML 2.0 specification.

• Added authorization and authentication failure and success cases to the sequence diagrams

• Fixed sub-section numbering for section 4

• Resolved all other issues from that review's meeting minutes

9.7. October 19, 2018

Added Printer guidance for how to specify support for multiple authentication methods for a particular URI, and how a Client might discover this and process it.

9.8. September 13, 2018

Updated with additional recommendations for Client and Printer on when (and when not) to rotate HTTP Digest parameters, to prevent operation failure.

9.9. September 5, 2018

Updated as per feedback from PWG August 2018 F2F:

• Updated file name and structure to make it a “best practices” document

• Moved all the authentication methods to a new section 4

9.10. June 29, 2018

Updated as per feedback from PWG May 2018 F2F:

• Added line numbers
• Resolved typos in diagrams in figures 3.5, 3.6, and the “new” 3.7 (TLS)
• Removed the second OAuth2 diagram
• Rewrote the TLS client authentication scheme description (contributed by Mike Sweet) and re-titled the section for its corresponding "uri-authentication-supported" keyword ('certificate')

9.11. May 10, 2018

Updated figures 6 and 7 (relating to OAuth2) to add a note indicating where the Printer might be able to acquire a user identifier suitable for making policy choices. Also made a few minor editorial updates.

9.12. April 30, 2018

Changed to Apache OpenOffice template. Added Mike Sweet as a co-author since he has contributed a great deal of content to the document. Resolved all “to-do” highlighted areas and resolved issues identified in the February 2018 vF2F minutes (https://ftp.pwg.org/pub/pwg/ipp/minutes/ippv2-f2f-minutes-20180207.pdf):

• Added sequence diagram for X.509 client authentication
• Added sequence diagram for hybrid 'oauth' / 'digest' authentication
• Many other changes

9.13. January 23, 2018

Updated as per email feedback and discussion:

• Fixed some editorial issues with naming HTTP Basic, HTTP Digest, and HTTP Negotiate, and some names of sections.
• Added mention of “printer-xri-supported”.
• Added additional references.
• Added additional sub-sections to capture Client and Printer recommendations for appropriate behavior when authentication is unsuccessful since the negative cases can vary widely.

Updated as per feedback from the November 2017 PWG vF2F and subsequent work with IPP WG members on specific details:

- Corrected OAuth2 sequence diagram to more correctly describe the sequence of operations and actors involved in an OAuth2 authenticated IPP Printer scenario.
- Added Implementation Recommendations that were revealed during the course of correcting the OAuth2 sequence diagram.

9.15. August 3, 2017

Initial revision.