Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 August 2026

P. Sawant
Apple Inc.
B. Brinckman
Cisco Systems

19 February 2026

Extensible Authentication Protocol (EAP) Using Privacy Pass Token
draft-ietf-emu-eap-ppt-02

Abstract

This document describes Extensible Authentication Protocol using Privacy Pass token (EAP-PPT) Version 1. The protocol specifies use of the Privacy Pass token for client authentication within EAP as defined in RFC3748. Privacy Pass is a privacy preserving authentication mechanism used for authorization, as defined in RFC9576. EAP-PPT must be performed only in a tunnel-based EAP method.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 23 August 2026.

Copyright Notice

Table of Contents

1. Introduction

This document specifies Extensible Authentication Protocol (EAP) method, EAP-PPT, which uses Privacy Pass token for EAP peer authentication; see [RFC9576] for more information about Privacy Pass. EAP-PPT MUST be used inside any tunnel-based EAP method that enables secure communication between a peer and a server by using Transport Layer Security (TLS) Protocol [RFC8446]. The tunnel-based EAP method MUST be a server authenticated TLS tunnel only.

Privacy Pass tokens are unlinkable authenticators that can be used to anonymously authorize a client [RFC9576]. Privacy Pass tokens are issued to peer by token issuers using an Issuance Protocol [RFC9578], and therefore, peer receives the token out of band of EAP-PPT. A client possessing such a token is able to prove that it was able to get it issued by a trusted issuer, without allowing the relying party redeeming the client's token (the origin) to link it with the issuance flow.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Much of the terminology in this document is defined in [RFC3748].

Additional terms are defined below:

NAI: Network Access Identifier [RFC7542]

EAP-PPT peer: This term is used for the entity acting as EAP peer. This term is identical to term Client defined in Section 2 of [RFC9576].

EAP-PPT server: This term is used for the entity acting as EAP server. This term is identical to term Server defined in Section 2 of [RFC9576].

Privacy Pass token: Unlinkable authenticator that can be used to anonymously authorize a client [RFC9577]. This is produced as an output of issuance protocol [RFC9578].

Token Challenge: An action by which a EAP-PPT Server requests EAP-PPT peer to present Privacy Pass Token for one of the presented challenges.

Token Redemption: An action by which a peer presents a Privacy Pass token to a EAP-PPT Server in EAP-PPT Protocol. See Section 2.2 of [RFC9577].

Identity provider: An entity that is responsible for authentication of end-user devices with the prupose of granting them access to a network resource.

3. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. Motivation

EAP is predominantly used for authentication of users and devices trying to join a network. Its security and extensibility capabilities makes it a popular choice in implementing secure network access. EAP is one of the most preferred authentication mechanisms used for secure wireless LAN access using [IEEE-802.11] standard, wired LAN access using [IEEE-802.1X] and Virtual Private Network (VPN) access. EAP is also used for secure network access for students and guests in academia, see [RFC7593].

One goal of privacy is to protect individual’s identity and personal information from eavesdropers, intermediaries and recipients in the network communication. An individual’s privacy may get compromized when network access is attempted using EAP as an authentication mechanism. The various privacy-specific threats are described in Section 5.2 of [RFC6973].

Typical approaches for authorizing clients, such as through the use of a permanent identity or service provider generated pseudo identity, are not privacy-friendly since they allow servers to track clients across sessions and interactions. This means service providers, identity providers, employers, or school/university administrators can track the individuals.

The goal of this specification is to protect an individual from the Section 5.2 of [RFC6973] in public and enterprise environments. EAP-PPT can be leveraged for authorization based on anonymous-credential authentication mechanisms. EAP-PPT takes a different approach: instead of carrying linkable state carrying information to servers, such as permanent identity or pseudonym, EAP peer presents Privacy Pass tokens that attest to this information. These tokens are anonymous in the sense that a given token cannot be linked to the protocol instance in which that token was initially issued.

[RFC9577] specifies the authentication scheme using Privacy Pass token over HTTP. [RFC9577] mainly serves use cases where access to restricted services require anonymous client authorization. Since [RFC9577] functions at the application layer of a networking stack, it justifies a need of a protocol that can offer the similar

functionality for the lower layers. EAP-PPT, performed inside a server-authenticated TLS tunnel offers anonymous network access to wired and wireless networks at those lower layers. Since EAP-PPT method provides unilateral authentication, it can be used together with responder authentication based on public key signatures in [RFC7296] protocol. [RFC7296] is a component of IPsec used for performing mutual authentication and establishing and maintaining Security Associations. [RFC7296] is widely used to implement remote access VPN service in public and enterprise environments, so EAP-PPT can be used to provide anonymous VPN services to clients.

In summary, EAP-PPT provides a solution for networks that wish to offer anonymous network access to users and devices, protecting their privacy, while still being able to authorize them based on the possession of valid token that proves that a trusted attestation was performed based on the policies they defined for network access.

5. Architecture Model

Figure 1 shows network architectural model for EAP-PPT.

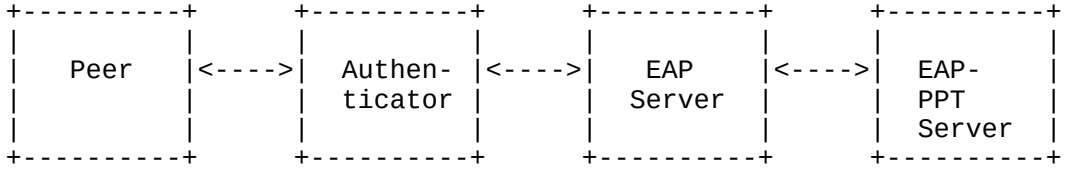

Figure 1: EAP-PPT Architectural Model

The entities depicted in Figure 1 are logical entities and may or may not correspond to separate network components. For example, the EAP Server and EAP-PPT Server might be a single entity; the authenticator and EAP Server might be a single entity; or the functions of the authenticator, EAP Server, and EAP-PPT Server might be combined into a single physical device. For example, typical [IEEE-802.11] deployments place the authenticator in an access point (AP) while a RADIUS Server may provide the Tunneled EAP Method and EAP-PPT method Server components. The above diagram illustrates the division of labor among entities in a logical manner and shows how a distributed system might be constructed; however, actual systems might be organized differently.

6. Protocol

6.1. Overview

A tunnel-based EAP method supports authentication in two phases after the initial EAP Identity request/response exchange. In the first phase, it uses a TLS [RFC8446] handshake to provide an authenticated key exchange and to establish a protected tunnel. The EAP peer and server that are configured to perform the peer authentication using EAP-PPT method, MUST establish the TLS tunnel without peer authentication. EAP server MUST NOT send CertificateRequest to the TLS client during the TLS handshake, when peer authentication is desired using EAP-PPT method. TLS-PSK cipher suites Section 9.2 of [RFC8446] MUST NOT be used.

The second phase of the authentication begins after the TLS tunnel is established. Any EAP method that fulfils the requirements specified in [RFC6678] is called tunnel-based EAP method.

A peer supporting EAP-PPT MUST NOT send its username or any other permanent identifiers in the first and subsequent EAP-Response/ Identity messages. EAP-Response/Identity message MUST contain only an anonymous NAI as per RFC 7542 Section 2.4 in order to route the authentication request to the right AAA system.

EAP-PPT authentication MUST be performed inside the server authenticated TLS tunnel established by the tunnel-based EAP method.

During the EAP-PPT authentication, the server challenges the peer to present a Privacy Pass token, and the Peer responds with a Privacy Pass token. Upon a successful verification of the token, the redemption of the token is deemed successful. EAP-PPT uses JavaScript Object Notation (JSON) [RFC8259] to encode the challenges, responses, results and errors. Encapsulation of EAP-PPT method can be supported by any tunnel- based EAP methods e.g. Protected EAP [PEAP], Tunneled Transport Layer Security EAP (TTLS) [RFC5281], EAP Flexible Authentication via Secure Tunneling (EAP-FAST) [RFC4851] and Tunnel Extensible Authentication Protocol (TEAP) [RFC7170].

Optionally, the Privacy Pass token MAY also carry extensions ([I-D.draft-ietf-privacypass-auth-scheme-extensions]) with additional metadata relevant to the EAP-PPT Server. An example of an extension that could be useful in EAP-PPT is token expiration, since tokens may be issued with a limited lifetime for security reasons. An expiration extension is described in [I-D.draft-hendrickson-privacypass-expiration-extension].

6.2. Successful Authentication

Figure 2 shows an example of basic, successful authentication exchange in EAP-PPT. At the minimum, EAP-PPT uses two roundtrips to authenticate and authorize the Peer. As in other EAP schemes, an identity request/response message pair is usually exchanged first. As specified in [RFC3748] the initial identity request is not required, and MAY be bypassed in cases where the EAP-PPT Server can presume the identity.

After obtaining the identity, the EAP-PPT Server constructs EAP-Request/PPT-Challenge message with a set of token Challenges and sends it to the EAP-PPT peer. EAP-Request/PPT-Challenge message encodes the set of token Challenges in JSON [RFC8259] format.

On receiving EAP-Request/PPT-Challenge message, the EAP-PPT peer looks at the each token Challenge and looks up the most suitable Privacy Pass token. If EAP-PPT Peer successfully finds the Privacy Pass token, it constructs EAP-Response/PPT-Challenge message containing the Privacy Pass token, and send it to the EAP-PPT server. EAP-Response/PPT-Challenge message encodes the response data in JSON [RFC8259] format.

The EAP-PPT server verifies the received Privacy Pass token in the EAP-Response/PPT-Challenge message. After a successful token Redemption, the EAP-PPT server sends EAP-Success.

EAP-PPT Server verifies the Privacy Pass token using a procedure called token Redemption Section 2.2 of [RFC9577].

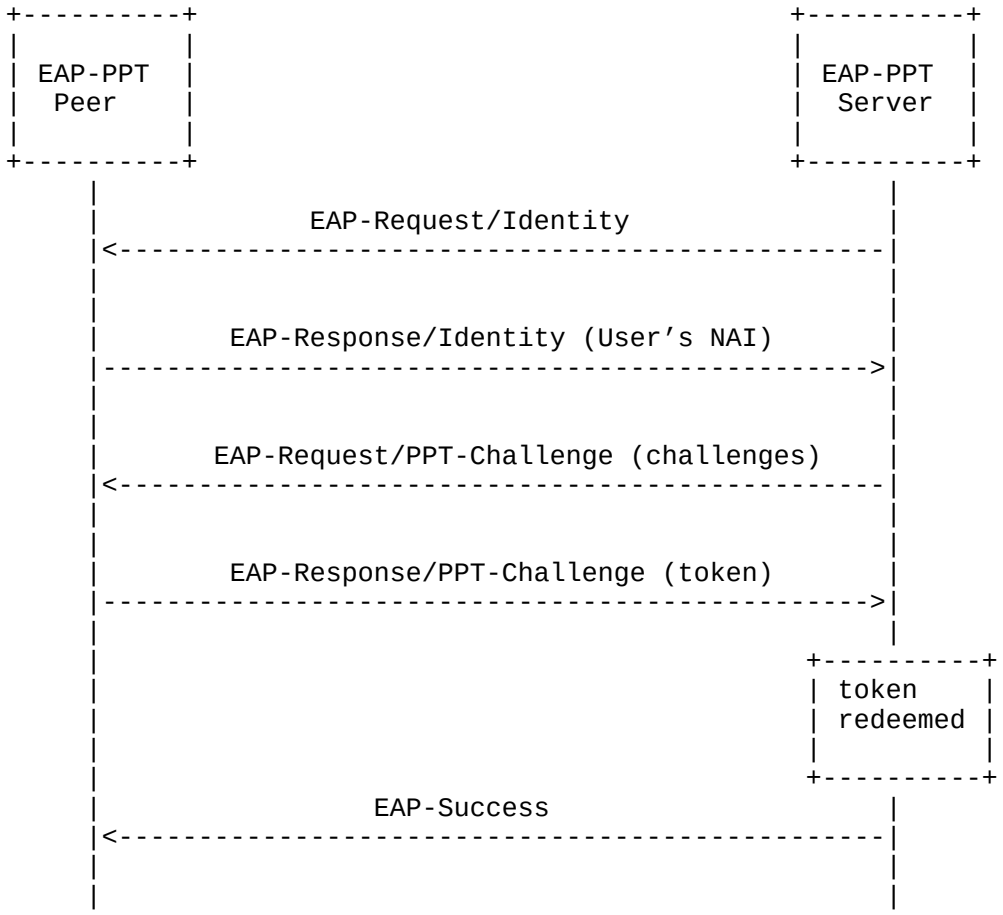

Figure 2: EAP-PPT Successful Authentication

6.3. Failed Authentication

Figure 3 shows how EAP-PPT server rejects the peer when token redemption fails. EAP-PPT Server sends EAP-Request/PPT-Error message containing the error information like error code and error description as described in Section 7.3.3.1. The error information is encoded in JSON [RFC8259] format. EAP-PPT peer responds to EAP-Request/PPT-Error with EAP-Response/PPT-Error without any data.

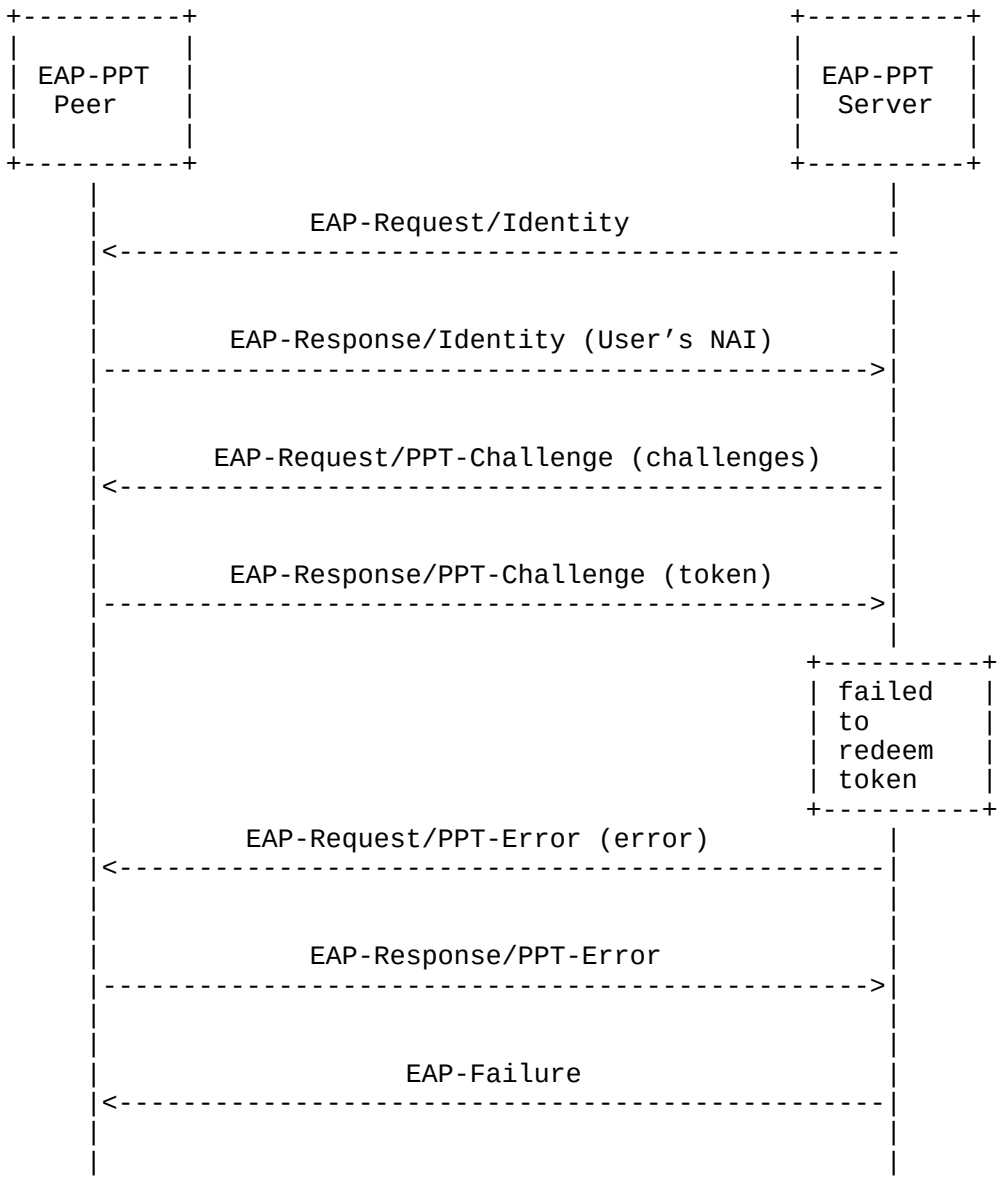

Figure 3: EAP-PPT Authentication Failure

6.4. Remediation

An EAP-PPT server MAY successfully validate a token, but fail to validate metadata carried in an extension. The EAP-PPT server MAY require different or more recently generated metadata, for example. In this case the EAP-PPT server MAY reject, or conditionally accept an EAP-PPT Authentication.

As shown in Figure 4, after successful token redemption, the EAP-PPT server MAY respond with a PPT error message containing error information like an error code and error description (see Section 7.3.3.1). to inform the EAP-PPT peer of the metadata validation issue. In this case, the EAP-PPT server MAY respond with an EAP-Failure or EAP-Success message, depending on the metadata specific policies set on the EAP-PPT server side. Since the peer proves the authenticity of issuance of token by providing cryptographically correct token, the EAP-PPT server MAY decide to authorize the Peer conditionally.

The EAP-PPT server MAY optionally also include a session-timeout value in the PPT-Error, informing the EAP-PPT peer how long the session will be permitted in order for the EAP-Peer to remediate and request a new token from its issuer. If the session-timeout attribute is included in the PPT-Error (see Section 7.3.3.1), the AAA server MUST also include a RADIUS Session-Timeout attribute (see Section 5.27 of [RFC2865]) with the same value in the Access-Accept RADIUS message to the authenticator (e.g., Network Access Server or IKEv2 Responder)). The EAP-PPT peer responds to the EAP-Request/PPT-Error with EAP-Response/PPT-Error without any data. The EAP-PPT peer MAY use the allotted session time to fetch a new token by contacting its attester. After getting a new token issued, the EAP-PPT peer may subsequently re-authenticate.

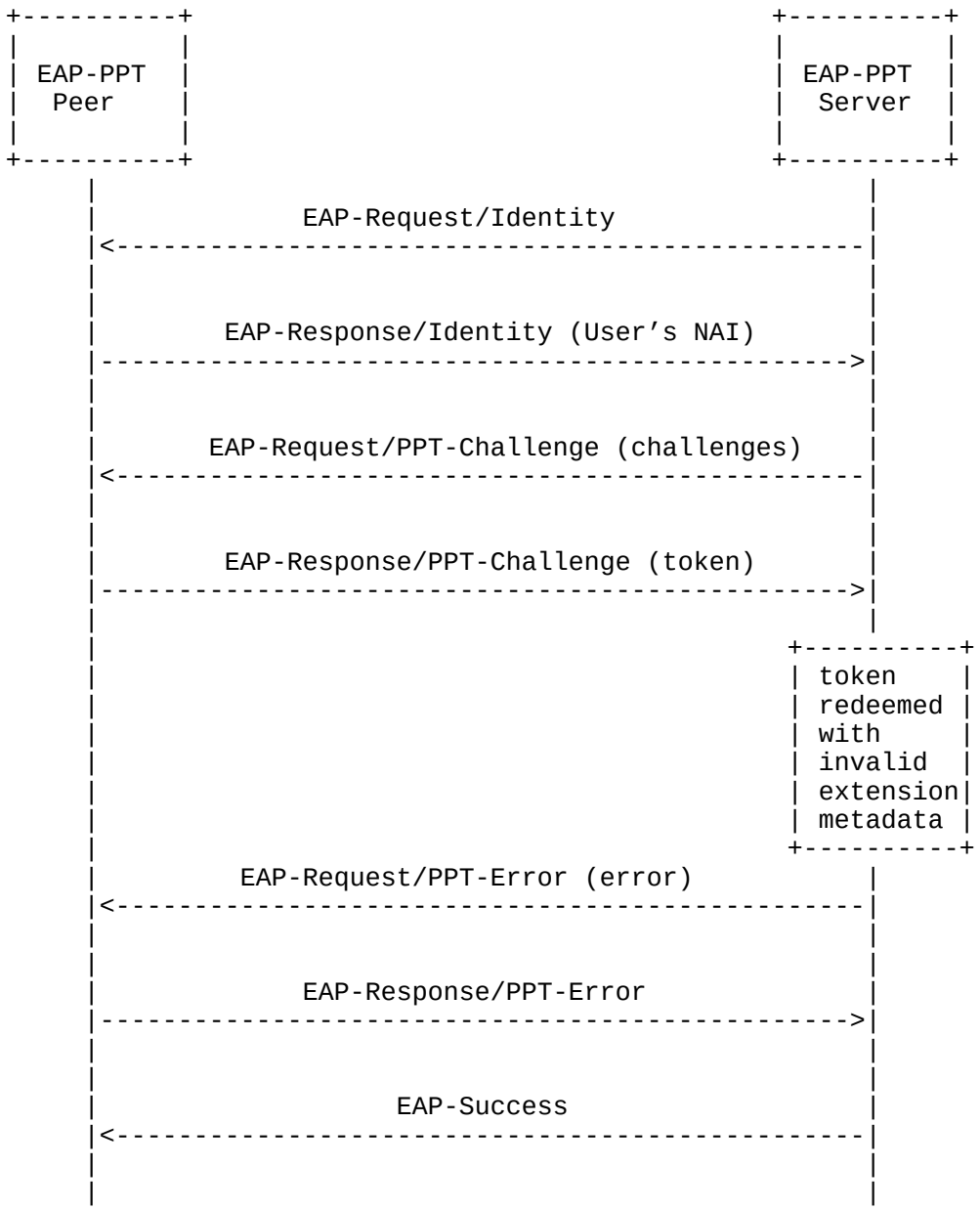

Figure 4: EAP-PPT Authentication Success with remediation

6.5. Privacy

The fundamental building block of privacy in EAP-PPT is use of Privacy Pass token, which is unlinkable authenticator, for authorization of the Peer. EAP-PPT peer selects an issuer to get a token issued from, using Issuance Protocol [RFC9578]. The Issuer generates a token response based on the token request, which is returned to the Client (generally via the Attester). Upon receiving the token response, the EAP-PPT peer computes a token from the token challenge and token response. This token can be validated by anyone with the per-Issuer key but cannot be linked to the content of the token request or token response.

If the EAP-PPT peer has a token, it includes it in a response to a challenge from EAP-PPT server. This token SHOULD be sent only once in reaction to a challenge; peers SHOULD NOT send tokens more than once, even if they receive duplicate or redundant challenges.

The EAP-PPT server validates that the token was generated by the expected Issuer and has not already been redeemed for the corresponding token challenge. Mechanism to prevent double-spending of tokens is out of scope of EAP-PPT method.

Section 4 of [RFC9576] discusses deployement models in detail. It is RECOMMENDED to use a deployment model that guarantees EAP peer-server, Issuer-EAP peer, and Attester-EAP server unlinkability. Mechanisms for enforcing non-collusion are out of scope of EAP-PPT method.

EAP-PPT peer MAY opt for token Caching by getting multiple tokens issued from a single token challenge structure (Section 2.1.1 of [RFC9577]). This improves privacy by separating the time of token issuance from the time of token redemption. Optionally, the peer MAY use a vaiant of Privacy Pass Issuance ([I-D.draft-ietf-privacypass-batched-tokens]) to get more tokens issued and cached at a time.

EAP peer and server MUST send anonymous Network Access Identifiers (NAIs) (Section 2.4 of [RFC7542]) in the first and subsequent EAP-Response/Identity messages. EAP peer MUST NOT send its username (or any other permanent identifiers) in the Identity Response. Following [RFC7542], it is RECOMMENDED to omit the username (i.e., the NAI is @realm), but other constructions such as a fixed username (e.g., anonymous@realm) is allowed. Note that the NAI MUST be a UTF-8 string as defined by the grammar in Section 2.2 of [RFC7542].

During TLS hanshake in the first phase, EAP peer MUST send a Certificate message containing no certificates as described in Section 4.4.2 of [RFC8446], if CertificateRequest message is received. Many client certificates contain an identity such as an email address, and therefore, this document forbids client authentication during first phase.

It is desired to support fast reconnect (Section 7.2.1 of [RFC3748]) by shortening the TLS conversation using session resumption mechanism (Section 2.1.2 of [RFC5216]) during the first phase. EAP peer presents an identifier that was issued previously by the server, to attempt the session resumption. When a peer attempts to resume a TLS session using such an identifier it allows the EAP server to detect peer's revisit to the network. Similarly, use of Protected Access Credential (PAC) in EAP-FAST method (Section 3.2.2 of [RFC4851]) can potentially help the server determine peer's presence across session resumptions. This document recommends use of session resumption to be limited to the current association to the network. The EAP peer MUST perform full TLS handshake durist the first phase after every new association to the network. For example, an EAP peer can continue to resume TLS sessions during the re-authentications as long as the client device is associated to same access point of the secure wireless LAN [IEEE-802.11], so session resumption MUST be used only on the same Authenticator as for the original session.

6.6. Key Material Generation

The keys generated by this protocol, MSK and EMSK, are each in 64 octets in length. The protocol uses TLS exporter interface [RFC5705] to generate the key material. The output of the exporter is intended to be associated with the TLS session established in the first phase, a unique label string, and a context. Type is the value of the EAP Type field defined in Section 2 of [RFC3748], and it contributes to the context value. For EAP-PPT, the Type value is 0x39. Context value is constructed by concatenating Type value with Privacy Pass token value that was sent in EAP-Response/PPT-Challenge message. Key material MUST be generated by the EAP-PPT peer after receiving EAP Success from the EAP-PPT server.

```
Type = 0x39
Context = Type || token
Key_Material = TLS_Exporter("EXPORTER_EAP_PPT_Key_Material",
                            Context, 128)
```

The MSK and EMSK are dervied from the Key_Material as described in Section 7.10 of [RFC3748].

MSK = Key_Material(0, 63)
EMSK = Key_Material(64, 127)

TLS_Exporter function is defined in Section 4 of [RFC5705]

6.7. Channel Binding

[RFC6677] defines channel bindings for EAP which solve the "lying NAS" and the "lying provider" problems, using a process in which the EAP peer gives information about the characteristics of the service provided by the authenticator to the Authentication, Authorization, and Accounting (AAA) server protected within the EAP authentication method. This allows the server to verify the authenticator is providing information to the peer that is consistent with the information received from this authenticator as well as the information stored about this authenticator.

EAP-PPT server can optionally request channel binding information to the EAP- PPT peer after a successful redemption of the token sent in EAP-Response/PPT- Challenge mesage. EAP-PPT server uses EAP-Request/ PPT-Channel-Binding message to request the channel binding information to the peer. EAP-PPT server MUST send EAP-Request/PPT-Channel-Binding message after a successful redemption of the token and before sending EAP-Success message. EAP-PPT peer MUST send channel binding information in EAP-Response/PPT-Channel-Binding message in response to EAP-Request/PPT-Channel-Binding message. EAP-PPT MUST send the channel-binding information as defined in Section 5.3 of [RFC6677].

EAP-Request/PPT-Channel-Binding message is optional, and therefore EAP-PPT server may skip it when the EAP server has already received the information through EAP methods executed before EAP-PPT.

7. Message Format

7.1. Packet Format

EAP-PPT Packet Format is shown below.

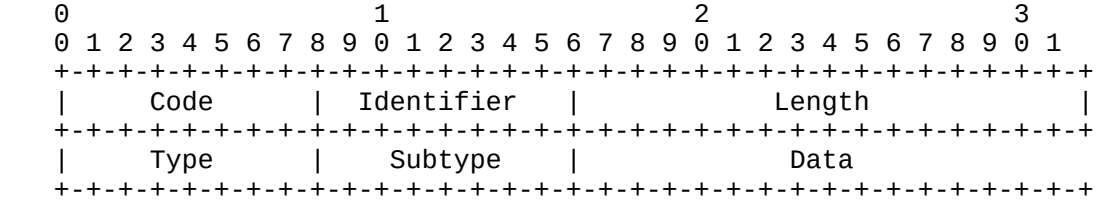

Figure 5: EAP-PPT Header

Code 1 for request, 2 for response.

Identifier The Identifier field is one octet and aids in matching responses with requests. The Identifier field MUST be changed for each request packet and MUST be echoed in each response packet.

Length The Length field is two octets and indicates the length of the EAP packet including the Code, Identifier, Length, Type, Subtype, and Data fields.

Type 57 (EAP-PPT)

Subtype Message subtypes as defined in Table 1

Data Data in JSON [RFC8259] format.

7.2. Subtypes

Subtype	Description
1	A PPT-Challenge request or PPT-Challenge response.
2	A PPT-Error request or PPT-Error response.
3	A PPT-Channel-Binding request or PPT-Channel-Binding response.

Table 1: EAP-PPT Subtypes

7.3. Messages

This section specifies the messages used in EAP-PPT.

7.3.1. EAP-Request/PPT-Challenge

The Server sends this message to the peer after successfully learning the identity of the peer. The purpose of this message is to present multiple token challenges to the peer and receive a Privacy Pass token for one of the challenges from the peer. This message is sent with subtype 1 (Table 1) and data is encoded in JSON [RFC8259] format as shown in Table 2 below -

Key	Type	Description
challenges	array	Array of one or more objects. Each element is an object that contains keys that are part the challenge. This is a required parameter.

Table 2: Token Challenges

Key	Type	Description
challenge	string	A string that contains a base64url token challenge value, encoded per [RFC4648]. This document follows the default padding behavior described in Section 3.2 of [RFC4648], so the base64url value MUST include padding. The token structure is defined in Section 2.2.1 of [RFC9577]. This key is based on the challenge parameter defined in Section 2.1.2 of [RFC9577]. This is a required parameter.
token-key	string	AA string that contains a base64url-encoded public key for use with the issuance protocol indicated by the challenge key. This key is based on the token-key parameter defined in Section 2.1.2 of [RFC9577]. This parameter MAY be omitted in deployments where peers are able to retrieve the Issuer key using an out-of-band mechanism.
extension-types	array	An array of ExtensionType that the EAP-PPT server is requesting the token to bind to. ExtensionType is defined in Section 3 of I-D.draft-ietf-privacypass-auth-scheme-extensions. This parameter is meaningful only if the Issuer, EAP-PPT peer and server have an out-of-band agreement to bind the extension to the token. This is an optional parameter.

Table 3: Token Challenge Keys

EAP-Request/PPT-Challenge Message carries JSON key "challenges" which is a JSON array of JSON objects. Each element in "challenges" is a JSON object that contains two keys i.e. "challenge" and "token-key", and optionally an array of "extension-types" as well, as shown in Table 3.

Example EAP-Request/PPT-Challenge Data -

```
{
    "challenges":
    [
        {
            "challenge": "AAIADmlzc3Vlci5leGFtcGxlIIo-g6M9mAB
            dLzC-9Bn6a_TNXGAF42sShbu0zNQPpLODAA5vcmlnaW4uZXh
            bXBsZQ==",
            "token-key": "MIIBUjA9BgkqhkiG9w0BAQowMKANMAsGCWC
            GSAFlAwQCAqEaMBgGCSqGSIb3DQEBCDALBglghkgBZQMEAgKi
            AwIBMAOCAQ8AMIIBCgKCAQEAyxrta2qV9bHOATpM_KsluUsuZ
            KIwNOQlCn6rQ8DfOowSmTrxKxEZCNS0cb7DHUtsmtnN2pBhKi
            7pA1I-beWiJNawLwnlw3TQz-Adj1KcUAp4ovZ5CPpoK1orQwy
            B6vGvcte155T8mKMTknaHl1fORTtSbvm_bOuZl5uEI7kPRGGi
            KvN6qwz1cz91l6vkTTHHMttooYHGy75gfYwOUuBlX9mZbcWE7
            KC-h6-814ozfRex26noKLvYHikTFxROf_ifVWGXCbCWy7nqR0
            zq0mTCBz_kl0DAHwDhCRBgZpg9IeX4PwhuLoI8h5zUPO9wDSo
            1Kpur1hLQPK0C2xNLfiJaXwIDAQAB"
        },
        {
            "challenge": "AAEADmlzc3Vlci5leGFtcGxlIIo-g6M9mA
            BdLzC-9Bn6a_TNXGAF42sShbu0zNQPpLODAA5vcmlnaW4uZXh
            hbXBsZQ==",
            "token-key": "67H-0zgxA2HAjQx1dpaWcSluBemaF9eSbf
            wopT-r1In6wPgryoYkmmaPOlv6s3TJ"
            "extension-types":
            [
                1,5,6
            ]
        }
    ]
}
```

Figure 6

7.3.2. EAP-Response/PPT-Challenge

The peer sends this message to the server in response to a valid EAP-Request/PPT-Challenge Message. Sending this Message indicates that the peer was able to look up a Privacy Pass token for one of the received challenges. This message is sent with subtype 1 (Table 1) and data is encoded in JSON [RFC8259] format as shown in Table 4.

Key	Type	Description
token	string	A string that contains base64url-encoded token structure value per Section 2.2.1 of [RFC9577]. This is a required parameter.
extensions	string	A string that contains base64url-encoded extension structure (Section 3 of .draft-ietf-privacypass-auth-scheme-extensions). This is an optional parameter.

Table 4: Token Challenge Response Keys

The peer MUST send empty token string when it fails to find a valid token for one of the received challenges. On receiving an empty token string in this message, the server MUST send EAP-Failure message to the peer.

Example EAP-Response/PPT-Challenge Data -

```
{
    "token": "AAEADmlzc4Vlci5leGFtcGxlIIo-g6M9mABdLzC-1Bn6a_TNX
    GAF52sShbu0zNQPpLODAA5vcmlnaW4uZXhhbXBsCB=="
}
```

Figure 7

7.3.3. EAP-Request/PPT-Error

The server sends this message to the peer when token redemption fails. The purpose of this message is to report redemption failure to the peer along with relevant information that may be useful to the peer. This message is sent with subtype 2 (Table 1) and data is encoded in JSON [RFC8259] format as shown in Table 5.

Key	Type	Description
code	number	An error code that describes the reason for the redemption failure. The range is 1-100. This key is required.
description	string	Human-readable UTF-8 text providing additional information, used to assist the user of the client device in understanding the error. This is an optional key.
session-timeout	number	Time in second after which the session is terminated by the authenticator. This is an optional key.

Table 5: Error Keys

Example EAP-Request/PPT-Error Data -

```
{
    "code": 1,
    "description": "invalid token format"
}
```

Figure 8

7.3.3.1. Error Codes

Code	Description
1	This code indicates a failure in validating the token data. This may occur due to incorrect formatting or encoding of the data.
2	This code indicates redemption failure. This is a fatal error, and only way for the peer to recover from this failure is to retry the EAP-PPT authentication with new token.
3	This code means the EAP-PPT server is unable to perform the token redemption at the moment. This can be used by the client to retry spending the token

	later.
4	This code indiates the server detected a double spend of the token. This is a fatal error, and only way for the peer to recover from this failure is to retry the EAP-PPT authentication with a new token.
5	This code indicates undefined failure. The client MAY choose to the spend the same oken later.
6	This code indicates token redemption success with an unexpected extension parameter value. This is a fatal error, and only way for the peer to recover from this failure is to retry the EAP-PPT authentication with a new token, binding to expected extension parameter value.
7	This code indicates token redemption success with an unexpected extension parameter value. However, the server side policy makes this a non-fatal error, and therefore, the peer is authorized unconditionally.
8	This code indicates token Redemption success with an unexpected extension parameter value. However, the server side policy makes this a non-fatal error, and therefore the peer is authorized conditionally. The condition here is - an authorization for limited time. The limited time authorization is indicated by sending session-timeout parameter along with the error code.
9-80	Reserved
81-100	Vendor Specific Errors.

Table 6: Error Codes

7.3.4. EAP-Response/PPT-Error

The peer sends this message as an acknowledgement to the server in response to a valid EAP-Request/PPT-Error Message. This message is sent with subtype 2 (Table 1) and it does not carry data.

7.3.5. EAP-Request/PPT-Channel-Binding

EAP-PPT server sends this message to the peer after a successful redemption of the token received in EAP-Response/PPT-Challenge message. The purpose of this message is to request channel binding information to the peer. This message is sent with subtype 3 (Table 1) and and it does not carry data.

7.3.6. EAP-Response/PPT-Channel-Binding

The peer sends this message as an in response to a EAP-Request/ PPT-Channel-Binding Message. This message is sent with subtype 3 (Table 1) and the data field contains channel-binding message as defined in Section 5.3 of [RFC6677]. EAP-PPT server MAY send EAP-Failure message if the channel-binding data is not found valid or satisfactory, depending on the server side policy.

8. Error Handling

8.1. Client Failure Scenarios

8.1.1. EAP-PPT peer found no valid token for token challenge

If on receipt of an EAP-Request/PPT-Challenge, the EAP-PPT peer cannot present a valid token matching for one of the received token challenges, then the EAP-PPT peer MUST respond with an empty token string in the EAP-Response/PPT-Challenge message. In this case, the EAP-PPT server MUST terminate the conversation by sending an EAP Failure packet.

8.1.2. EAP-PPT peer found no token with valid extension-types for token challenge

If on receipt of an EAP-Request/PPT-Challenge, the EAP-PPT peer cannot present a valid token bound to the extension-type(s) requested by the EAP-PPT server for one of the received token challenges, then the EAP-PPT peer MUST respond with an empty token string in the EAP-Response/PPT-Challenge message. In this case, the EAP-PPT server MUST terminate the conversation by sending an EAP Failure packet.

8.2. Server Failure Scenarios

8.2.1. EAP-PPT server found no valid token challenge for user NAI

If on receipt of a EAP Identity Response the EAP-PPT server does not have a token challenge for the user’s NAI, the EAP-PPT server MUST terminate the conversation by responding with an EAP Failure packet.

8.2.2. EAP-PPT server is unable to validate token data

If on receipt of an EAP-Response/PPT-Challenge, the EAP-PPT server is unable to validate the token data presented by the EAP-PPT peer (due to incorrect data, formatting or encoding), the EAP-PPT server MUST respond with an EAP-Request/PPT-Error with error code 1 (see Section 7.3.3.1). The EAP-PPT peer MUST subsequently acknowledge the error with an EAP-Response/PPT-Error message, after which the EAP-PPT server MUST respond with EAP Failure as shown in Figure 3.

8.2.3. EAP-PPT server token redemption failure

If on receipt of an EAP-Response/PPT-Challenge, the EAP-PPT server token redemption fails, the EAP-PPT server MUST respond with an EAP-Request/PPT-Error with error code 2 (see Section 7.3.3.1). The EAP-PPT peer MUST subsequently acknowledge the error with an EAP-Response/PPT-Error, after which the EAP-PPT server MUST respond with EAP Failure as shown in Figure 3. The EAP-PPT peer MUST NOT use this token in subsequent authentication.

8.2.4. EAP-PPT server temporary failure

If the EAP-PPT server is (temporarily) unable to perform token redemption, and it receives an EAP-Response/PPT-Challenge, the EAP-PPT server MUST respond with an EAP-Request/PPT-Error with error code 3 (see Section 7.3.3.1). The EAP-PPT peer MUST subsequently acknowledge the the error with an EAP-Response/PPT-Error message, after which the EAP-PPT server MUST respond with EAP Failure as shown in Figure 3. The EAP-PPT peer MAY use this token in subsequent authentication.

8.2.5. EAP-PPT server detected double spend

The EAP-PPT server MAY implement double spend detection, to ensure a token is only used once. If the EAP-PPT server implementing double spend detection detects double spend of a token sent in an an EAP-Response/PPT-Challenge, the EAP-PPT server MUST respond with an EAP-Request/PPT-Error with error code 4 (see Section 7.3.3.1). The EAP-PPT peer MUST subsequently acknowledge the error with an EAP-Response/PPT-Error message, after which the EAP-PPT server MUST respond with EAP Failure as shown in Figure 3. The EAP-PPT peer MUST NOT use this token in subsequent authentication.

8.2.6. EAP-PPT server undefined failure

If the EAP-PPT server is experiencing an undefined failure, when receiving an EAP-Response/PPT-Challenge, the EAP-PPT server MUST respond with an EAP-Request/PPT-Error with error code 5 (see Section 7.3.3.1). The EAP-PPT peer MUST subsequently acknowledge the error with an EAP-Response/PPT-Error message, after which the EAP-PPT server MUST respond with EAP Failure as shown in Figure 3. The EAP-PPT peer MAY use this token in subsequent authentication.

8.2.7. EAP-PPT server token redemption success with unexpected extension value

If on receipt of an EAP-Response/PPT-Challenge, the EAP-PPT server finds an unexpected extension parameter value, the EAP-PPT server MAY deem this to be a fatal error. In this case the EAP-PPT server MAY respond with an EAP-Request/PPT-Error with error code 6 (see Section 7.3.3.1). The EAP-PPT peer MUST subsequently acknowledge the error with an EAP-Response/PPT-Error message, after which the EAP-PPT server MUST respond with EAP Failure as shown in Figure 3. The EAP-PPT peer MUST NOT use this token in subsequent authentication.

8.3. Conditional Acceptance Scenarios

8.3.1. EAP-PPT server redemption, unexpected extension value, unconditional access

If on receipt of a EAP-Response/PPT-Challenge, the EAP-PPT server token redemption succeeds, but the EAP-PPT server finds an unexpected extension parameter value, The EAP-PPT server MAY deem this to be a recoverable error and allow the session to proceed unconditionally. In this case, the EAP-PPT server MAY respond with an EAP-Request/PPT-Error with error code 7 (see Section 7.3.3.1). The EAP-PPT peer MUST subsequently acknowledge the error with an EAP-Response/PPT-Error message, after which the EAP-PPT server MUST respond with EAP Success as shown in Figure 2.

8.3.2. EAP-PPT server redemption, unexpected extension value, conditional access

If on receipt of a EAP-Response/PPT-Challenge, the EAP-PPT server token redemption succeeds, but the EAP-PPT server finds an unexpected extension parameter value, The EAP-PPT server MAY deem this to be a recoverable error and allow the session to proceed conditionally. In this case the EAP-PPT server MAY respond with an EAP-Request/PPT-Error with error code 8 (see Section 7.3.3.1). The EAP-PPT server MUST send a session-timeout value in the response message. The EAP-PPT peer MUST subsequently acknowledge the error with an EAP-

Response/PPT-Error message, after which the EAP-PPT server MUST respond with EAP Success as shown in Figure 2. The EAP Server MUST include a session-timeout attribute in the RADIUS Access-Accept packet to the Authenticator, so it can terminate the session when the session-timeout condition is no longer met.

An example of such condition is when the peer needs to remediate its device to be compliant with the network access policy, or if the peer needs to get a new token issued from the Issuer with expected extension parameter value. The length of rgw session timer should in principle be as short, as possible but long enough for the device to reach compliance. For example for token issueance, if there is no user interaction required for issuance, a session timer of 1 minute should be sufficient. For remediation where user interaction is required, the session timeout could be more like 5 to 10 minutes.

9. Deployment Considerations

EAP-PPT can be leveraged in a number of use cases and deployment models. This section covers generic deployment recommendations to ensure end-to-end privacy and unlinkability of tokens. This section also describes some specific expected deployment models in which EAP-PPT can be leveraged.

Although this section covers deployment of Origin, Issuer and Attester as it relates to the EAP-PPT server, specifics on how to deploy Issuer and Attester are not described here but can be found in Section 4 of [RFC9576].

9.1. Recommendations for preserving privacy

9.1.1. Collocating other functions with the EAP-PPT Server

As discussed in Section 4 of [RFC9576] and in Section 6.5, it is recommended to use a deployment model that guarantees EAP peer-server, Issuer-EAP peer, and Attester-EAP server unlinkability. This is especially pertinent in public use cases. In private use cases a single entity could deploy all functions.

It is recommended to collocate the phase 1 EAP-Server with the EAP-PPT server, as EAP-Server separation can introduce vulnerabilities as described in Section 10.3.

9.1.2. Protecting client identity

Please refer to the Section 6.5 section for deployment considerations that are required to protect the client identity.

9.1.3. Separating Issuance and Verification over time

Section 3.1 of [RFC9576] describes the interaction between Privacy Pass Issuance and Verification protocols. As described, in many cases, when a Client interacts with an Origin, a Client will obtain a token at the time of that interaction. In this case the time between Issuance and Verification is short enough to allow for correlation.

In order to further reduce the probability of collusion between actors participating in Issuance and Verification and achieve Issuer-Client and Origin-Client unlinkability, Issuance and Verification can be separated over time. A client can request Issuance of one or more tokens and cache them in secure storage. This allows separation in time between Issuance and Verification of the token, so time-based correlation is not possible. When leveraging EAP-PPT to access network resources, it is possible that the client does not have a network interface available to perform Issuance over, so also for this reason caching tokens is preferred.

9.2. Recommendations for usage in public use cases

In public use cases, a network service provider may be working with one or more identity providers that are authenticating end-user devices using privacy pass tokens. As described in Section 9.1.1 it is recommended for the EAP-PPT server to be implemented by an entity other than the Attester or Issuer, to avoid the perception of collusion. In a public deployment scenario, the EAP-PPT server is likely to be collocated with the network service provider, or could be a service that the network service provider consumes from a 3rd party service provider, other than the Attester or Issuer.

In order to verify a token, EAP-PPT Server requires key material for the issuers specified in the TokenChallenge. In a public use case, this information will have to be shared between the issuer and EAP-PPT Server. The mechanism in which the Issuer shares this information with the EAP-PPT server is out of scope of this document.

9.3. Recommendations for usage in private use cases

It is recommended that the guidelines stipulated in Section 9.2 are also followed for private deployments, however in use cases where the network service provider is also the Attester, collocation of entities may be unavoidable. When collocating entities, separating Issuance and Verification over time as described in Section 9.1.3 provides additional privacy protection, as it becomes harder for entities to collude.

9.4. Recommendations for usage in federated use cases (OpenRoaming)

OpenRoaming, as described in [I-D.draft-tomas-openroaming], is an open federation of entities of different types, mainly targeted at providing public Wi-Fi access. OpenRoaming defines distinct roles in its federation architecture: Network Access Providers provide access to network resources, and Identity Providers authenticate users for those network access providers. Members of the federation are identified by private PKI, managed by the Wireless Broadband Alliance (WBA). The members use these certificates to mutually authenticate each-others and secure RADIUS over TLS (RadSec) messages used to transport EAP conversations between Network Access Providers and Identity Providers. A Network Access Provider discovers the authoritative Identity Provider for a client by resolving the realm portion of the outer identity provided by the client as described in [RFC7585].

OpenRoaming comprises of a privacy policy, and aims to protect en-user privacy, however as it uses RADIUS attributes and EAP, inherently, information about end-users could be shared between Identity Provider and Network Access Provider. Examples of RADIUS attributes that could expose user privacyt are Calling-Station-Id (mac address of hte device), Chargeable-User-ID, NAS-ID (location). {Section 8 of ?I-D.draft-tomas-openroaming}} describes the RADIUS attributes OpenRoaming supports. EAP-PPT can add additional privacy protection to a federated use case such as OpenRoaming by separating the Issuance from Verification, so the entity performing the Authentication is not able to willingly or unwillingly share private information.

Where an OpenRoaming IDP both issues and verifies a credential, with EAP-PPT these roles are separated. In order to implement EAP-PPT in OpenRoaming, the Attester/Issuer would have to have an agreement with the EAP-PPT server verifying or redeeming the token. Together they are the OpenRoaming IDP. Alternatively, new roles could be defined in the OpenRoaming federation to allow Attesters/Issuers to interoperate with EAP-PPT servers within the OpenRoaming federation.

The EAP-PPT Server could be implemented by the Network Access Provider directly, or by an entity in the federation.

10. Security Considerations

10.1. PrivateToken authentication Scheme

Security considerations applicable discussed in Section 5 of [RFC9577] are applicable to EAP-PPT.

10.2. Integrity Protection

Since EAP-PPT method is used for anonymous authentication of EAP peer, it is REQUIRED to execute it within a server authenticated TLS tunnel, provided by a tunnnel-based EAP method. When EAP-PPT is used to authenticate IKEv2 initiator to the responder, it is REQUIRED to use it in conjunction with a public-key-signature- based authentication of the responder to the initiator, before initiating the EAP-PPT authentication.

10.3. EAP Server implementation

Allowing the EAP Phase 1 conversation to be terminated at a different server than the EAP-Phase 2 conversation can introduce vulnerabilities if there is not a proper trust relationship and protection for the protocol between the two servers.

As EAP-PPT is an identity-free credential, it mitigates loss of identity protection scenarios better than EAP-methods carrying identity. Identity protection is ensured, even if the credential is exposed to an attacker. Offline dictionary attacks are also mitigated with EAP-PPT as the credential is a single-use cryptograhically signed token.

Separation of Phase 1 and Phase 2 EAP server with EAP-PPT as the inner EAP method can still introduce vulnerabilities to on-path active attacks between these EAP Servers if there is not a proper trust relationship between the servers, or if the protocol between the servers is not properly secured. An attacker could intercept a token in the PPT-Challenge response, or alter an EAP-Success or EAP-Failure message. It is important to note however that due to the single-use identity-free nature of the credential, the longevity of the attack is limited.

Therefore, separation of the EAP-Server (Phase 1) from the EAP-PPT Server i (Phase 2) conversation is NOT RECOMMENDED.

10.4. Channel Binding

[RFC6677] defines channel bindings for EAP which solve the "lying NAS" and the "lying provider" problems, using a process in which the EAP peer gives information about the characteristics of the service provided by the authenticator to the Authentication, Authorization, and Accounting (AAA) server protected within the EAP authentication method. This allows the server to verify the authenticator is providing information to the peer that is consistent with the information received from this authenticator as well as the information stored about this authenticator.

When collocating the EAP and EAP-PPT servers, as recommended in Section 10.3, channel binding can be implemented by leveraging a Phase 1 EAP method that supports Channel binding as defined in [RFC6677]. It is therefore RECOMMENDED to leverage a Phase 1 EAP method that supports Channel binding with EAP-PPT, for example TEAP [RFC7170], as described in Section 3.11.4 of [RFC7170].

10.5. Token Redemption Server implementation

EAP-PPT server MAY be implemented to perform token Redemption flow with an external redemption service, configured with required keys for redemption. In such scenario, a malicious EAP peers may generate a lot of protocol requests to mount a denial-of-service attack on the service. The EAP-PPT server implementation SHOULD take this into account and SHOULD take steps to limit the requests it generates towards the redemption service.

10.6. Abuse

EAP-PPT provides anonymous network access to peers possessing valid Privacy Pass tokens. This anonymous access can potentially be abused. This is not a problem that is unique to EAP-PPT. Other EAP tunneled EAP methods or other methods that provide anonymous access, such as EAP-PSK [RFC4764], EAP-TTLS and EAP-TLS with anonymous certificates, also have similar abuse potential.

To counter such abuse, network operators may implement various abuse mitigation techniques, such as:

* Leverage the attestation: EAP-PPT relies on a token that is proof of an attestation. The attestation required for network access will rely on the policy of the network provider. As such the attestation policy can be designed to ensure that both the device and the user meet that policy before being issued a token. This can help mitigate abuse by ensuring that only authorized users and devices are able to obtain tokens. Further more, in case where abuse is detected, future attestation can be denied.

* Leverage a Layer 2 identifier: In case of a public network, it may not be possible to update future attestation based on abuse detection. In this case a session can be blocked based on the Layer 2 identifier of the device (for example the MAC address). As described in [RFC9797], there are various levels of trust a device may have in a network. Based on the trust level, the device may present a Layer 2 identifier that is stable over time, or a randomized one. In case of EAP Authentication, devices present a stable Layer 2 identifier that is stable across sessions within a certain timeframe. This layer 2 identifer is used for

association, so the advantage of leveraging it for abuse mitigation is that access can be denied at association time, before EAP authentication is performed.

* Leverage network-level abuse mitigation techniques: Network operators may have various network-level abuse mitigation techniques in place, such as rate-limiting, traffic filtering, and monitoring of network traffic to detect and mitigate abusive behavior. These techniques can be applied to EAP-PPT authenticated sessions. With these techniques, mitigation does not happen by excluding the user or device, but happens by mitigating the abuse itself.

10.7. Security Claims

This section provides the security claims required by [RFC3748].

Auth. mechanism: Privacy Pass token

Ciphersuite negotiation: No

Mutual authentication: No

Integrity protection: NO. However, EAP-PPT method executed within a tunnel-based EAP method established TLS tunnel is integrity protected. The cleartext EAP-PPT messages outside the tunnel are not integrity protected.

Replay protection: NO. However, EAP-PPT method executed within a tunnel-based EAP method established TLS tunnel is replay protected. The cleartext EAP-PPT messages outside the tunnel are not replay protected.

Confidentiality: No. However, EAP-PPT method executed within a tunnel-based EAP method established TLS tunnel is encrypted.

Key derivation: Yes

Key strength: See Section 5.1 of [RFC5216]

Dictionary attack prot.: N/A

Fast reconnect: No

Cryptographic binding: N/A

Session independence: N/A

Fragmentation: No

Key Hierarchy: No

Channel binding: Yes

11. IANA Considerations

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the EAP-PPT protocol, in accordance with BCP 26 [RFC8126].

An EAP Method Type number will be requested for EAP-PPT.

This document also calls for a registry of EAP-PPT error codes described in Section 7.3.3.1.

12. References

12.1. Normative References

[I-D.draft-ietf-privacypass-auth-scheme-extensions] Hendrickson, S. and C. A. Wood, "The PrivateToken HTTP Authentication Scheme Extensions Parameter", Work in Progress, Internet-Draft, draft-ietf-privacypass-auth-scheme-extensions-02, 27 May 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-auth-scheme-extensions-02>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/rfc/rfc2865>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <https://www.rfc-editor.org/rfc/rfc3748>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/rfc/rfc4648>.

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <https://www.rfc-editor.org/rfc/rfc5216>.

[RFC5705] Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)", RFC 5705, DOI 10.17487/RFC5705, March 2010, <https://www.rfc-editor.org/rfc/rfc5705>.

[RFC6677] Hartman, S., Ed., Clancy, T., and K. Hoeper, "Channel-Binding Support for Extensible Authentication Protocol (EAP) Methods", RFC 6677, DOI 10.17487/RFC6677, July 2012, <https://www.rfc-editor.org/rfc/rfc6677>.

[RFC7170] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, DOI 10.17487/RFC7170, May 2014, <https://www.rfc-editor.org/rfc/rfc7170>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/rfc/rfc7296>.

[RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <https://www.rfc-editor.org/rfc/rfc7542>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/rfc/rfc8446>.

[RFC9577] Pauly, T., Valdez, S., and C. A. Wood, "The Privacy Pass HTTP Authentication Scheme", RFC 9577, DOI 10.17487/RFC9577, June 2024, <https://www.rfc-editor.org/rfc/rfc9577>.

[RFC9578] Celi, S., Davidson, A., Valdez, S., and C. A. Wood, "Privacy Pass Issuance Protocols", RFC 9578, DOI 10.17487/RFC9578, June 2024, <https://www.rfc-editor.org/rfc/rfc9578>.

12.2. Informative References

[I-D.draft-hendrickson-privacypass-expiration-extension] Hendrickson, S. and C. A. Wood, "Privacy Pass Token Expiration Extension", Work in Progress, Internet-Draft, draft-hendrickson-privacypass-expiration-extension-03, 24 January 2025, <https://datatracker.ietf.org/doc/html/draft-hendrickson-privacypass-expiration-extension-03>.

[I-D.draft-ietf-privacypass-batched-tokens] Robert, R., Wood, C. A., and T. Meunier, "Batched Token Issuance Protocol", Work in Progress, Internet-Draft, draft-ietf-privacypass-batched-tokens-06, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-privacypass-batched-tokens-06>.

[I-D.draft-tomas-openroaming] Tomas, B., Grayson, M., Canpolat, N., Cockrell, B., and S. Gundavelli, "WBA OpenRoaming Wireless Federation", Work in Progress, Internet-Draft, draft-tomas-openroaming-07, 23 January 2026, <https://datatracker.ietf.org/doc/html/draft-tomas-openroaming-07>.

[IEEE-802.11] IEEE, "IEEE Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", February 2021.

[IEEE-802.1X] IEEE, "IEEE Standard for Local and Metropolitan Area Networks--Port-Based Network Access Control", February 2020.

[PEAP] Microsoft Corporation, "Protected Extensible Authentication Protocol (PEAP)", June 2021.

[RFC4764] Bersani, F. and H. Tschofenig, "The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method", RFC 4764, DOI 10.17487/RFC4764, January 2007, <https://www.rfc-editor.org/rfc/rfc4764>.

[RFC4851] Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, "The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST)", RFC 4851, DOI 10.17487/RFC4851, May 2007, <https://www.rfc-editor.org/rfc/rfc4851>.

[RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, DOI 10.17487/RFC5281, August 2008, <https://www.rfc-editor.org/rfc/rfc5281>.

[RFC6678] Hoeper, K., Hanna, S., Zhou, H., and J. Salowey, Ed., "Requirements for a Tunnel-Based Extensible Authentication Protocol (EAP) Method", RFC 6678, DOI 10.17487/RFC6678, July 2012, <https://www.rfc-editor.org/rfc/rfc6678>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/rfc/rfc6973>.

[RFC7585] Winter, S. and M. McCauley, "Dynamic Peer Discovery for RADIUS/TLS and RADIUS/DTLS Based on the Network Access Identifier (NAI)", RFC 7585, DOI 10.17487/RFC7585, October 2015, <https://www.rfc-editor.org/rfc/rfc7585>.

[RFC7593] Wierenga, K., Winter, S., and T. Wolniewicz, "The eduroam Architecture for Network Roaming", RFC 7593, DOI 10.17487/RFC7593, September 2015, <https://www.rfc-editor.org/rfc/rfc7593>.

[RFC9576] Davidson, A., Iyengar, J., and C. A. Wood, "The Privacy Pass Architecture", RFC 9576, DOI 10.17487/RFC9576, June 2024, <https://www.rfc-editor.org/rfc/rfc9576>.

[RFC9797] Henry, J. and Y. Lee, "Randomized and Changing Media Access Control (MAC) Addresses: Context, Network Impacts, and Use Cases", RFC 9797, DOI 10.17487/RFC9797, June 2025, <https://www.rfc-editor.org/rfc/rfc9797>.

Authors' Addresses

Paresh Sawant
Apple Inc.
Email: paresh_sawant@apple.com

Bart Brinckman
Cisco Systems
Email: bbrinckm@cisco.com