Supplemental Authentication in TLS 1.3

`draft-rosomakho-tls-supplemental-auth-00`

Yaroslav Rosomakho

Tirumaleswar Reddy

Rifaat Shekh-Yusef

Hannes Tschofenig

TLS
IETF126, July 2026, Vienna

Primary use case to address

Multiple identities associated with the TLS Client

- Separate user and device certificate validation is a very common requirement in enterprise VPN space
- User certificate authenticates the user, device certificate authenticates the device
- Device certificate is a common way to prove that the device is company managed
- Enterprise VPN clients implement this in proprietary ways with severe security limitations…

Secondary use case to address

Multiple identities associated with the TLS Server

- SPIFFE/WIMSE workload authentication can use either webPKI issued hostname certificate or a private X509-SVID
- Presenting both certificate chains at the same time would help

- Potentially could be useful to present multiple hostname certificates at the connection time

Additional potential use cases

- PQ/T (alternative solution to dual signatures)
- PQ/PQ

Server-side supplemental authentication example

Client

ClientHello

Server

Server-side supplemental authentication example

Client **Server**

ClientHello
+supplemental_certificate_requests

- **Indicates that client can accept supplemental certificates from the server**
- **May include a list of CR extensions to constrain provided supplemental server certificates**
- **Ignored by servers that do not support supplemental certificates or do not wish to provide them**

Server-side supplemental authentication example

Client

ClientHello
+supplemental_certificate_requests

Server

ServerHello
EncryptedExtensions
Certificate

Server-side supplemental authentication example

Client

ClientHello
+supplemental_certificate_requests

Server

ServerHello
EncryptedExtensions
Certificate
+supplemental_certificate

- **A flag indicating that at least one supplemental certificate will follow Server's Finished**
- **Finished must be followed by the next supplemental Certificate and nothing else**

Server-side supplemental authentication example

Client

ClientHello
+supplemental_certificate_requests

Server

ServerHello
EncryptedExtensions
Certificate
+supplemental_certificate
CertificateVerify
Finished

Server-side supplemental authentication example

Client

ClientHello
+supplemental_certificate_requests

- **Supplemental Certificate message only allowed if supplemental_certificate flag was set on the previous Certificate**
- **Encrypted with *server_application_traffic_secret_0***

Server

ServerHello
EncryptedExtensions
Certificate
+supplemental_certificate
CertificateVerify
Finished
Certificate

Server-side supplemental authentication example

Client

ClientHello
+supplemental_certificate_requests

Server

ServerHello
EncryptedExtensions
Certificate
+supplemental_certificate
CertificateVerify
Finished
Certificate
+supplemental_certificate

- **The flag indicates that one more supplemental Certificate will follow**

Server-side supplemental authentication example

Client

ClientHello
+supplemental_certificate_requests

Server

ServerHello
EncryptedExtensions
Certificate
+supplemental_certificate
CertificateVerify
Finished
Certificate
+supplemental_certificate
CertificateVerify
Finished

- **Transcript of ClientHello..server Finished + previous server supplemental authentication messages**

Server-side supplemental authentication example

Client

ClientHello
+supplemental_certificate_requests

- **Certificate of final supplemental authentication flight does not have *supplemental_certificate* flag set**

Server

ServerHello
EncryptedExtensions
Certificate
+supplemental_certificate
CertificateVerify
Finished
Certificate
+supplemental_certificate
CertificateVerify
Finished
Certificate
CertificateVerify
Finished

Client-side supplemental authentication example

Client

ClientHello

Server

ServerHello
EncryptedExtensions
CertificateRequest
+supplemental_certificate_requests
Certificate
CertificateVerify
Finished

Certificate
+supplemental_certificate
CertificateVerify
Finished
Certificate
CertificateVerify
Finished

- **Client supplemental authentication messages are encrypted with *client_application_traffic_secret_0***
- **Transcript of ClientHello..client Finished + previous client supplemental authentication messages**

Key features of the proposed design

- Does not modify main TLS 1.3 handshake
- Does not modify application protocols (unlike Exported Authenticators)
- All supplemental authentication flights are sent before application data, so application has all the context from the start (unlike TLS 1.3 Post-Handshake Authentication)
- Can apply to both sides (unlike TLS 1.3 Post-Handshake Authentication)
- Applicable to various use cases benefiting from multiple Certificates

Next steps

- Thoughts?
- Suggestions?
- Comments?