

Use Identity as Raw Public Key in EAP-TLS

draft-chen-emu-eap-tls-ibs-00

IETF108-2020-EMU

Meiling Chen /China Mobile

Li Su /China Mobile

Back Ground

- X509 Certificate management costs;
- Certificate can be relatively large;
- Certificate chains long, too many intermediate certificates;
- Certificate-based authentication is not suitable for restricted environment, such as IoT devices;
- RFC 7250 specified using Raw Public key in TLS and DTLS with two extensions(client_certificate_type, server_certificate_type);
- RFC 6507 specified an IBS algorithm with Elliptic curve cryptography called ECCSI;

Objective & Contents

- **Objective**

- specifies the use of identity as a raw public key in EAP-TLS with TLS1.2 and TLS1.3.

- **Contents**

- Structure of the Raw Public Key Extension
- EAP-TLS1.2 extends raw public key in authentication procedure
- EAP-TLS1.3 authentication procedure with raw public keys

-EAP-TLS1.2 extends raw public key in authentication procedure

the types of server certificates the client is able to process

```
(TLS client_hello  
+signature_algorithm  
server_certificate_type,  
client_certificate_type)->
```

Indicates the selected type of client certificate

the types of certificate the client can provide to the server

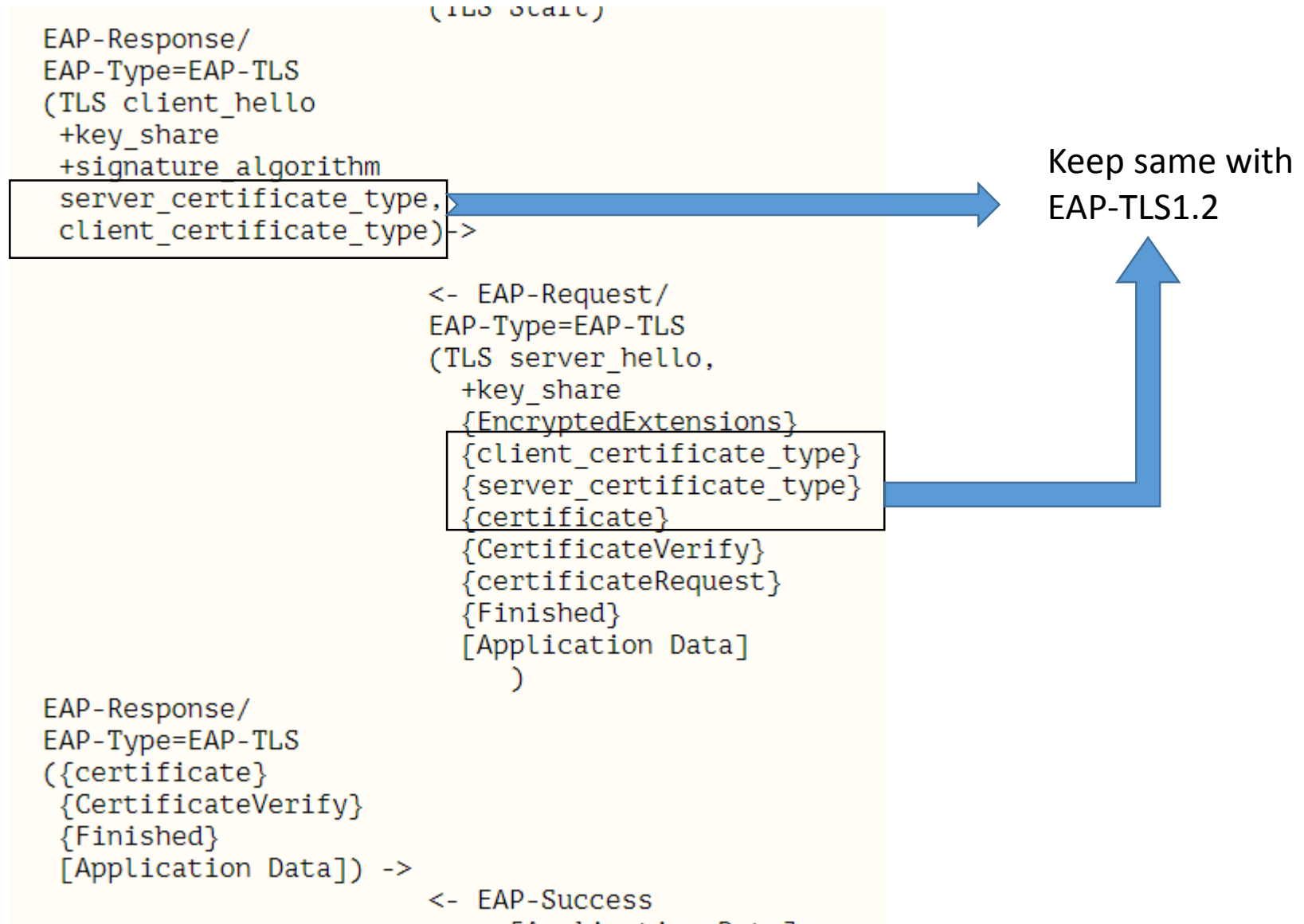
```
<- EAP-Request/  
EAP-Type=EAP-TLS  
(TLS server_hello,  
{client_certificate_type}  
{server_certificate_type}  
{TLS certificate}  
{TLS server_key_exchange}  
{TLS certificate_request}  
{TLS server_hello_done}  
)
```

Indicates the type of Certificate in the Certificate load

```
EAP-Response/  
EAP-Type=EAP-TLS  
(TLS certificate,  
TLS client_key_exchange,  
TLS certificate_verify,  
TLS change_cipher_spec,  
TLS finished) ->
```

Contains:
1、 raw public key ,
2、 the selected signature
3、 algorithm the hash of the algorithm's public parameters

-EAP-TLS1.3 authentication procedure with raw public keys



Example for EAP-TLS1.3-IBS

```
EAP-Response/  
EAP-Type=EAP-TLS  
(TLS client_hello  
signature_algorithm = (eccsi_sha256)  
server_certificate_type = (RawPublicKey,...)  
client_certificate_type = (RawPublicKey,...))->
```

```
<- EAP-Request/  
EAP-Type=EAP-TLS  
(TLS Start)
```

```
<- EAP-Request/  
EAP-Type=EAP-TLS  
(TLS server_hello,  
+key_share  
{client_certificate_type = RawPublicKey}  
{server_certificate_type = RawPublicKey}  
{certificate = (1.3.6.1.5.5.7.6.29, hash  
value of ECCSIPublicParameters),  
serverID})  
{certificate_request = (eccsi_sha256)}  
{certificate_verify = {ECCSI-Sig-Value}}  
{Finished}  
[Application Data]  
)
```

```
EAP-Response/  
EAP-Type=EAP-TLS  
({certificate = ((1.3.6.1.5.5.7.6.29,  
hash value of ECCSIPublicParameters),  
ClientID}),  
{certificate_verify = (ECCSI-Sig-Value)},  
{Finished})  
[Application Data] ->
```

```
<- EAP-Success
```



OID for ECCSI

ToDo

- More discussion
- Comments and co-authors are welcome!