

YourApp – EveryWhere- AnyTime

Pascal.Urien@telecom-paris.fr

<https://tools.ietf.org/html/draft-urien-tls-se-01>

<https://tools.ietf.org/html/draft-urien-tls-im-03>

<https://github.com/purien/TLS-SE>

The Concept

- YourApp, On-Line
 - Is embedded in a secure element EAL 5/6 (up to 7 levels)
- YourApp server works over a TLS1.3 embedded server
 - TLS-SE: TLS Secure Element
 - <https://tools.ietf.org/html/draft-urien-tls-se-01>
- YourApp client works over a TLS1.3 client
 - Client credentials are (optionally) stored and used in a secure element
 - TLS-IM: TLS Identity Module
 - <https://tools.ietf.org/html/draft-urien-tls-im-03>



Why TLS1.3 ?

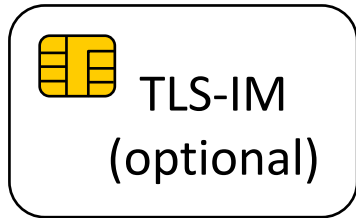
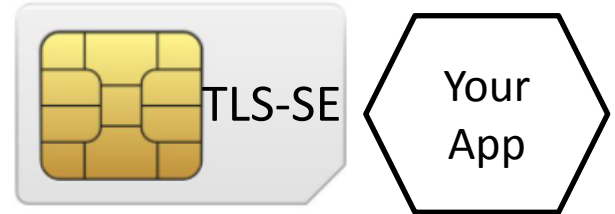
- State of art for communication security
 - Several years of debates between security experts at IETF.
 - Privacy enforcement with Diffie-Hellman Exchange over Elliptic Curve (ECDHE)
 - Authenticated Encryption with Associated Data (AEAD)
 - Server and client authentication based on PKI or pre-shared-key (PSK)
- TLS-SE 1.0 works with AES-128-CCM cipher-suite, ECDHE (over SECP256k1), and 32 bytes PSK.
- Next version will support PKI.

TLS1.3 -RFC 8446- Basic Exchange

Cipher-suite(s)

```

Key  ^ ClientHello
Exch | + key_share*   PublicKey for ECDHE
    | + signature_algorithms*
    | + psk_key_exchange_modes*  ECDHE
    v + pre_shared_key*         ----->
                                     PSK-ID and PSK binding
    
```



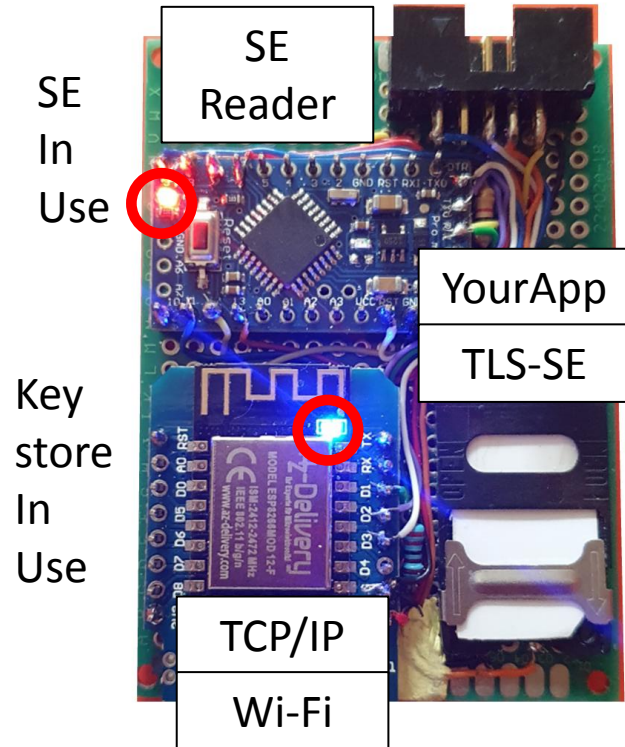
```

ServerHello ^ Key
+ key_share* | Exch
+ pre_shared_key* v
{EncryptedExtensions} ^ Server
{CertificateRequest*} v Params
{Certificate*} ^
{CertificateVerify*} | Auth
{Finished} v
<----- [Application Data*]
    
```

```

^ {Certificate*}
Auth | {CertificateVerify*}
v {Finished} ----->
  [Application Data] <-----> [Application Data]
    
```

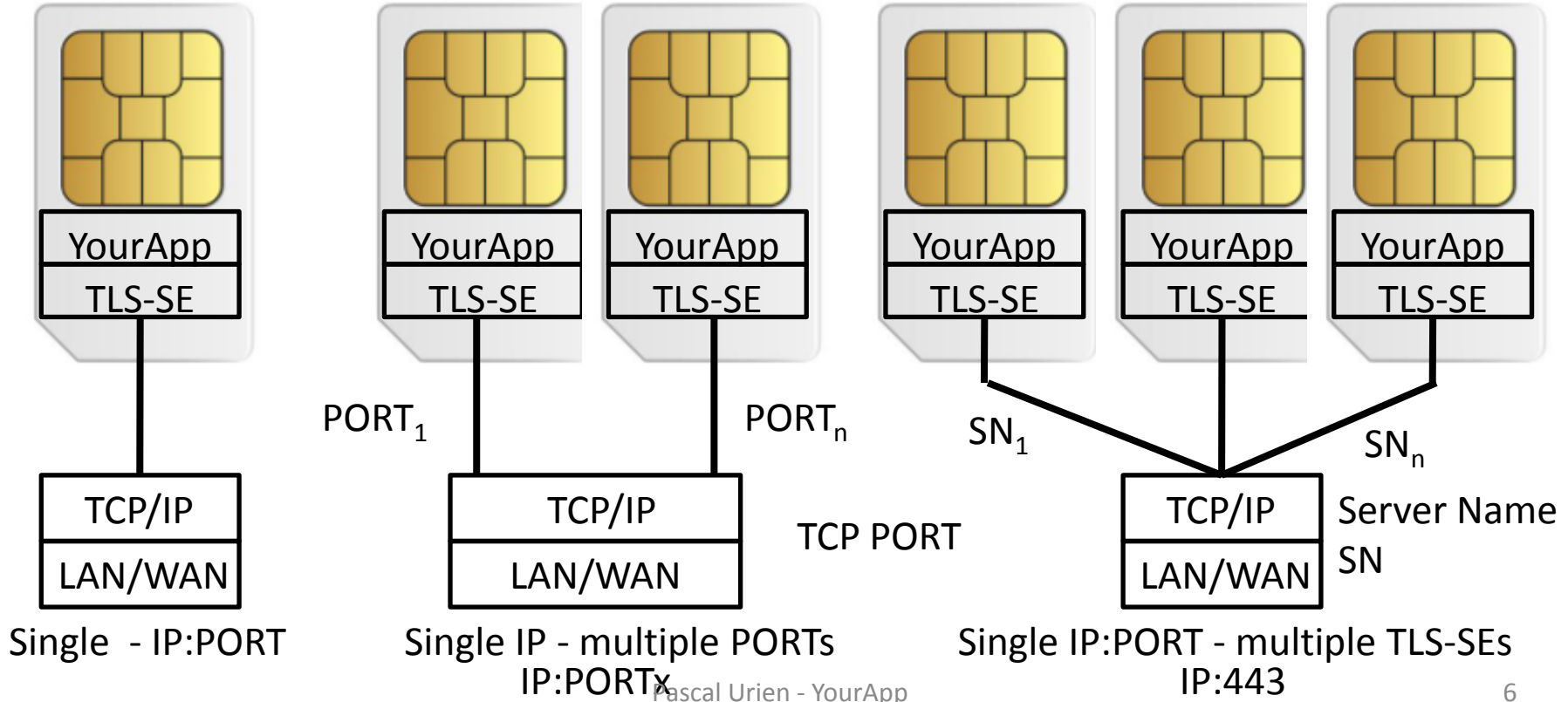
YourApp example: Blockchain Keystore



xy= key index (two hexadecimal digits)

?00CrLf	Get Version
?01[data]CrLf	Echo
cxyCrLf	Clear Key
gxyCrLf	Generate Key Pair
Xxy[PrivKey]CrLf	Set Private & Public key
txy[Seed]CrLf	Set BIP32 Seed
vxyCrLf	Get BIP32 Seed
bxy[n ₁ ...n _p]CrLf	Set a BIP32 (p x 32bits) path
pxyCrLf	Get Public Key
rxycrLf	Get Private Key
sxy[data]CrLf	Sign

Scalability Issue



Questions ?

<https://github.com/purien/TLS-SE>