Clien	t (a)	erver (b)
$r_{\sf a} \stackrel{{\sf random}}{\longleftarrow} \{0,1\}^{256}$ Ephermal key $:Q_{\sf a} \leftarrow d_{\sf a}G$	ClientHello : $r_{\rm a}$ KeyShare : $Q_{\rm a}$ ServerHello : $r_{\rm b}$ KeyShare : $Q_{\rm b}$	
		$\begin{split} S_{handshake} &\leftarrow Extract(Derive(S_{early}, \ 'derived', \emptyset), x) \\ S_{master} &\leftarrow Extract(Derive(S_{handshake}, 'derived', \emptyset), 0) \end{split}$
$Handshake \ traffic \ key : K_{handshake_a} \leftarrow \ Derive(S_{handshake}, \ 'c \ hs \ traffic', \ transcript)$	$\{ \text{Certificate} : \text{ Public key with CA signature} \}_{K_{\text{handshake}_b}}$	— Handshake traffic key : $K_{handshake_b} \leftarrow Derive(S_{handshake}, 's hs traffic', transcript)$
	$\{ CertificateVerify : Transcript \ with \ ECDSA \ signature \}_{K_{handshake_b}} \\ \leqslant \\ \{ Finished : \ HMAC(K_{finished}, \ transcript) \}_{K_{handshake_b}} \\ \leqslant \\ $	— Finished key : $K_{\sf finished} \leftarrow {\sf Derive}(K_{\sf handshake_b}, \ ' \sf finished', \ \sf transcript)$
$Finished \ key : K_{finished} \ \leftarrow \ Derive(K_{handshake_a}, \ 'finished', \ transcript)$	$\{Finished: \ HMAC(K_{finished}, \ transcript)\}_{K_{handshake_{a}}}$	$ ightarrow S_{ ext{resumption}} \leftarrow ext{Derive}(S_{ ext{master}}, ext{ 'res master', transcript})$
$Application\ traffic\ key: K_{a_0} \leftarrow \ Derive(S_{master},\ 'c\ ap\ traffic',\ transcript)$	$\{ \text{Application Data} \}_{K_{b_0}}$ $= \{ \text{Application Data} \}_{K_{a_0}}$	— Application traffic key : $K_{b_0} \leftarrow Derive(S_{master}, 's\ ap\ traffic', \ transcript)$ \rightarrow
	NewSessionTicket : {session key ID, IV, encrypted state, $HMAC()$ } K_{b_0} (Connections terminated. That triggers session resumption with 0-RTT)	 Creates a pre-shared key (PSK) binding to enable session resumption
$S_{early} \leftarrow Extract(0, S_{resumption})$ $Binder \ key : K_{binder} \leftarrow Derive(S_{early}, \ 'res \ binder', \emptyset)$ $Early \ Traffic \ Key : K_{early} \leftarrow Derive(S_{early}, \ 'c \ e \ traffic', \ transcript)$ $Finished \ key : K_{finished} \leftarrow Derive(K_{binder}, \ 'finished', \ transcript)$	ClientHello: KeyShare: PskKeyExchangeModes: 'psk_dhe_ke' EarlyDataIndication PreSharedKey: {session key ID, HMAC($K_{ m finished}$, transcript)} {Application Data} $_{K_{ m early}}$	\Rightarrow
	$ServerHello: \dots \\ KeyShare: \dots \\ PreSharedKey: \{session \ key \ ID\} \\ EncryptedExtensions: \{EarlyDataIndication\}_{K_{handshake_b}} \\ \leftarrow$	$S_{\text{handshake}} \leftarrow \text{Extract}(\text{Derive}(S_{\text{early}}, \text{ 'derived'}, \emptyset), x)$ $\text{Handshake traffic key}: K_{\text{handshake}_b} \leftarrow \text{Derive}(S_{\text{handshake}}, \text{ 's hs traffic'}, \text{ transcript})$
	$\{ Finished: \ HMAC(K_{finished}, \ transcript) \}_{K_{handshake_{b}}} \\ <$	$S_{master} \leftarrow Extract(Derive(S_{handshake}, 'derived', \emptyset), 0)$ $$
	$\{ {\sf Application \ Data} \}_{K_{b_0}}$ $\{ EndOfEarlyData \}_{K_{early}}$	— Application traffic key : $K_{ extsf{b}_0} \leftarrow \ extsf{Derive}(S_{ extsf{master}}, \ ' extsf{s ap traffic}', \ extsf{transcript})$
$Finished \ key : K_{finished} \ \leftarrow \ Derive(K_{handshake_a}, \ 'finished', \ transcript)$	$\{ Finished : HMAC(K_{finished}, transcript) \}_{K_{handshake_{a}}}$	→
	$\{$ Application Data $\}_{K_{\mathbf{b}_0}}$	_
Application traffic key : $K_{a_0} \leftarrow Derive(S_{master}, 'c aptraffic', transcript)$	$\{ {\sf Application \ Data} \}_{K_{a_0}}$ protocol. It serves as a quickstarter to understand the protocol flows. It may contain inaccurate of	> oversimplified representations
1) TLS Settings	protocor. It serves as a quickstarter to understand the protocol flows. It may contain inaccurate o	n oversimplined representations.

1) TLS Settings Cipher Suite: TLS_AES_128_GCM_SHA256 Digital Signature: ecdsa_secp256r1_sha256

Key Exchange: secp256r1 (NIST P-256) with (G,n) as part of domain parameters, with public and private key in the form of (Q,d) Pre-Shared Key Cipher: TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA384

2) Protocol Notations

Key Extraction Function: Extract(salt, keying material)

Key Derive Function: Derive(secret, label, transcript), where transcript is the concatenation of each included handshake message.

Encryption: {plaintext}_{key}, which denotes an AEAD-Encrypt operation with write key and IV generated from key.