5. THE PORPOSED SCHEME

5.1 System Initialization Phase

在進行驗證之前，我們假設所有的可信的 6G 運營商（6G Operator）都已初始化並公開了系統參數。以 6G 運營商 A（O_A）為例，O_A 將選擇$s_{O_A} \in \mathbb{Z}_q^*$作為系統私鑰，並且$s_{O_A}$應秘密保存，計算$P_{pub_{O_A}} = s_{O_A}P$作為系統公鑰。使用變色龍雜湊生成一個公開雜湊密鑰 HK_{O_A} 和秘密陷門密鑰 TK_{O_A}。然後，從一個橢圓曲線 $E(\mathbb{F}_p)$ 上的生成元 P 開始選擇一個階數為 q 的循環群 G，其中 p 和 q 是兩個質數。此外，定義安全的雜湊函數，其中H：$\{0,1\}^* \rightarrow \{0,1\}^*$，公共參數$param = \{G, P, q, p, g, P_{pub_{O_A}}, HK_{O_A}, \mathrm{H}\}$ 將 $param$ 寫入區塊鏈，以便不同網路切片請求和授權。

5.2 Registration

在訪問網路切片上的應用服務器之前，UE 需要完成註冊以及 3GPP TS 33.501 中定義的初始認證。用戶在完成 6G 網絡接入後，才可以選擇網絡切片服務。為了確保用戶身份的安全性，我們將使用PID^i來代替真實身份ID_i，並且只有信任的 6G 運營商可以還原用戶的真實身份。這樣的設計確保了在一般情況下用戶的隱私不會被洩露，但在需要追蹤身份時，信任的 6G 運營商可以進行追蹤。當用戶註冊時，我們將採用 FIDO 的驗證方式，要求用戶的設備提供相關的用戶資訊，同時要求設備生成公鑰和私鑰。在生成密鑰對之後，設備將會把公鑰回傳至系統以便儲存。

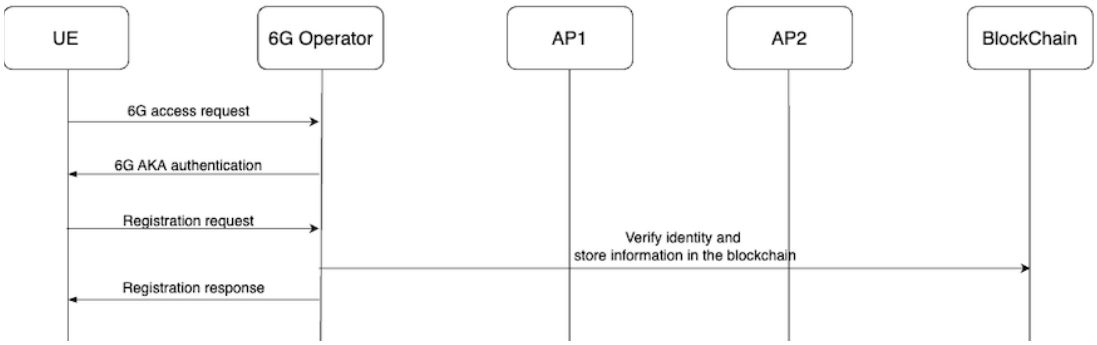

圖 x. UE 註冊流程圖

(1) UE $\rightarrow O_A : \{PID^i, X_i, FIDO_{PK}, coin, T1\}$

用戶選擇一個隨機數 $x_i \in \mathbb{Z}_q^*$ 作為私鑰，x_i 應被秘密保存，再來計算 $X_i = x_i P$ 作為公鑰。設備將透過 FIDO 認證器生成一對公私鑰 $FIDO_{PK}, FIDO_{SK}$。$FIDO_{SK}$ 由設備保管，$FIDO_{PK}$ 將用於註冊。然後選擇一個隨機數 n_u 來計算匿名身份 $PID^i = (PID_1^i, PID_2^i)$，其中 $PID_1^i = n_u P$, $PID_2^i = ID_i \oplus (n_u P_{pub_{O_A}})$，$ID_i$ 為用戶的真實身份。並將身份註冊請求 $\{PID^i, X_i, FIDO_{PK}, coin, T1\}$ 發送給 BC，其中 $coin$ 為 UE 的計費帳戶。

演算法：

(2) $O_A \rightarrow BC$

當 O_A 收到註冊請求後，根據 PID^i 還原用戶的真實身份 ID_i。利用 FIDO 公鑰 $FIDO_{PK}$ 對用戶進行驗證，確定用戶身份。透過智能合約，把 PID^i 做為索引儲存訊息，並將初始查詢時間設為零。用戶計費帳戶也會被儲存，初始值同時會為零。O_A 將記錄用戶的註冊訊息 $\lambda_{ij} = \{PID^i, X_i\}$，並把所有註冊後用戶的訊息 $w_i = \{\lambda_{1j}, \lambda_{2j}, \ldots, \lambda_{ij}\}$，打包成一個消息 m，其中 $m = \{ID_{O_A}, w_i, Sig_{TK_{O_A}}\{\mathrm{H}(w_i)\}$，$ID_{O_A}$ 為當下的 6G 運營商的真實身份。然後，隨

機選擇 $r, s \in \mathbb{Z}_q^*$，計算變色龍雜湊 $h = r - (HK_{O_A}{}^{(\mathrm{H}(m\|r)} \cdot g^s mod\ p)\ mod\ q$，其中 HK_{O_A} 是公開的雜湊密鑰。最後，將 m 和變色龍雜湊 h 打包成交易 $TX_{O_A} = \{m, h\}$，存在區塊鏈上，並發給其他 6G 運營商。

同樣，不同的網路切片上的應用服務器在認證前也需要進行身份註冊，註冊的過程與用戶註冊過程類似，不過網路切片不需要進行 FIDO 註冊，以及由於需要了解當前的應用服務器，因此不需要用匿名的身份。

Algorithm 2 Chameleon Hash

Input: m, HK, TK
Output: CH
1: Select prime p, q where $p = 2q + 1$;
2: Select prime g;
3: Select random values r and s, where $r, s \in Z_q$;
4: Compute chameleon Hash value $h = r - (y^{H(m\|r)} \cdot g^s \mod p) \mod q$, where $H : \{0,1\}^* \to Z_q$;
5: **return** (h, r, s);

5.3 Slice Access Authentication

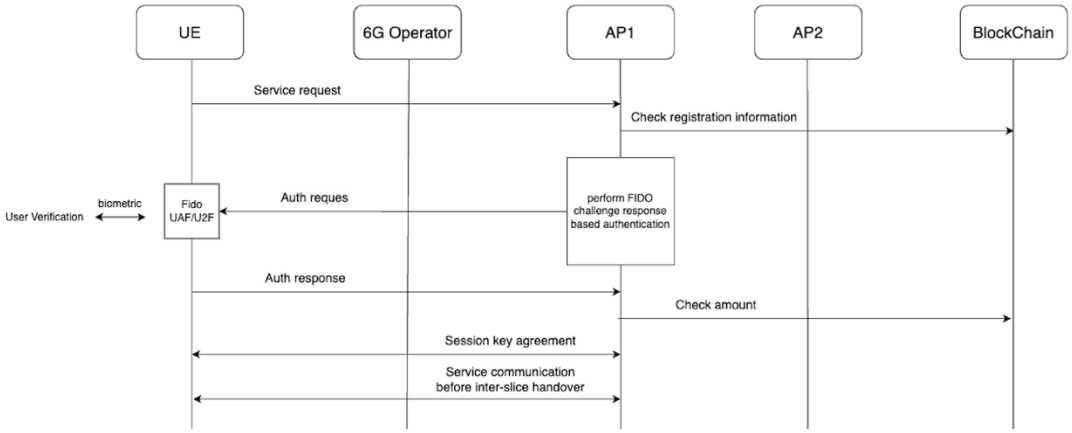

圖 x Slice Access Authentication

為了要訪問網路切片上的應用服務器 AP_1，UE 也需要進行對應的驗證。

(1) $\mathrm{UE} \to \mathrm{AP}_1 : \{X_i, PID_i, T2\}$

用戶 UE 將會根據需求選擇對應的應用服務器發送連線驗證請求，其中 X_i 為 UE 的公鑰，PID_i 為 UE 的匿名身份，$T2$ 為請求的時間戳。

(2) $\mathrm{AP}_1 \to BC : \{PID_i\}$

應用服務器（AP_1）收到用戶的請求後，首先先檢查 T2 是否小於 Δt，驗證 T2 的

有效性。然後透過發送 $\{PID_i\}$ 給 BC，並檢查該用戶是否註冊。

(3) $\text{AP}_1 \rightarrow UE : \{C_{FIDO}, r_{ap_1}, T3\}$

應用服務器（AP_1）將發送一個 FIDO 的 challenge-response 給 UE，進行 FIDO 的身份認證，其中r_{ap_1}為一個隨機數。

(4) $UE \rightarrow \text{AP}_1 : \{sig(C_{FIDO}),\ r_u\ , T4\}$

用戶收到 AP_1 發送來的 FIDO 挑戰後，首先先驗證 T3 是否小於Δt，然後將 C_{FIDO} 使用其 FIDO 私鑰 $FIDO_{SK}$ 進行簽章，並回傳$\{sig(C_{FIDO}),\ r_u\ , T4\}$給 AP_1，其 r_u 為一個隨機數。

(5) AP_1

AP_1收到$\{sig(C_{FIDO}),\ r_u\ , T4\}$ 後，首先先驗證 T4 是否小於Δt，並透過 UE 的 FIDO 公鑰 $FIDO_{PK}$來進行驗證簽章，如果驗證成功，則表示身份認證通過，進行密鑰協商。

(6) $\text{AP}_1 \rightarrow UE : \{Y_{ap_1}\}$

在認證完成後，AP_1 將使用 Diffie-Hellman 密鑰交換協議來生成會話密鑰。AP_1 將生成一個隨機數 α 並計算$DH_{ap_1} = g^a \ mod\ p$，並發送給 UE。

(7) $UE \rightarrow \text{AP}_1 : \{Y_u\}$

同時，UE 將生成一個隨機數 b 並計算$DH_u = g^b \ mod\ p$，並發送給AP_1。

(8) UE, AP_1 收到傳送來的 Diffie-Hellman 公鑰後，雙方使用對方的公鑰和自己的私鑰計算共享密鑰K。$K_{ap_1} = (DH_u)^a \ mod\ p$，$K_u = (DH_{ap_1})^b \ mod\ p$，由於 Diffie-Hellman 協議的特性，$K_{ap_1} = K_u$，即共享密鑰K相同。

(9) 最終，UE 和 AP_1 計算會話密鑰 $SK_{u,ap_1} = \text{H}\big(K \parallel r_u \parallel r_{ap_1} \parallel T3 \parallel T4\big)$。

(10) 當身份認證和會話密鑰交換完成，我們將使用智能合約自動開始計費系統。系統將記錄用戶的開始使用時間。智能合約將檢查用戶的帳戶餘額是否足夠支付一天的服務，如果不足，通知用戶並拒絕服務。我們使用智能合約在每天固定時間檢查所有用戶，並自動扣除對應的日費用。如果在任何時間用戶的帳戶不足下一天的費用，系統將自動登出用戶，並通知用戶。而

當用戶完成服務或服務被終止時，將進行結算。

5.4 Inter-slice Handover Authentication

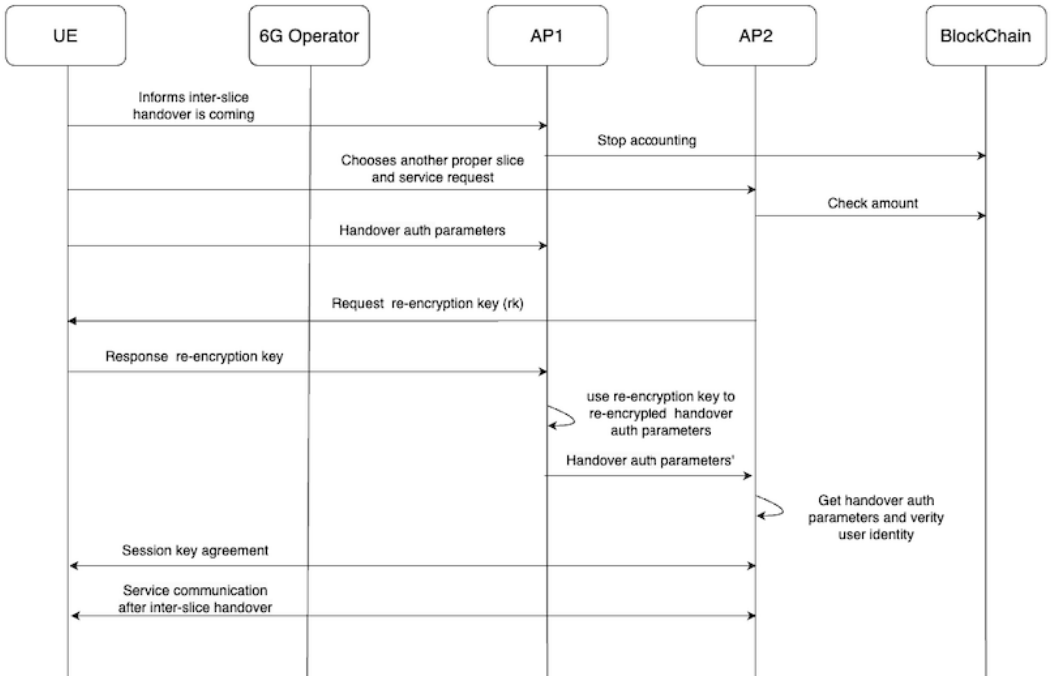

圖 x Inter-slice Handover Authentication

當用戶需要交換網路切片時，首先會先通知目前的應用服務器AP_1，並選擇不同的網路切片上的應用服務器AP_2，來進行跨網路切片的認證。

(1) $\mathrm{UE} \rightarrow \mathrm{AP}_1 : \{PID_i\}$

首先用戶會先發送要交換網路切片的需求給目前的AP_1。

(2) $\mathrm{AP}_1 \rightarrow \mathrm{BC} : \{PID_i\}$

AP_1透過智能合約停止用戶在目前的應用服務器上的時間計算。

(3) $\mathrm{UE} \rightarrow \mathrm{AP}_2 : \{X_i, PID_i, T5\}$

當用戶要選擇不同的網路切片時，用戶將向新的應用服務商發送認證請求。其中X_i為 UE 的公鑰，PID_i 為 UE 的匿名身份，$T5$為請求的時間戳。

(4) $\mathrm{AP}_2 \rightarrow BC : \{PID_i\}$

應用服務器（AP_2）收到用戶的請求後，首先先驗證 T5 是否小於Δt。然後透過發送$\{PID_i\}$給 BC，當 BC 收到後，查詢 UE 在區塊鏈帳本中的註冊交易，並檢

查該用戶的計費帳戶餘額。

(5) UE $\rightarrow$ AP_1 : $\left\{Enc_u\left(Sig_{x_i}\left(SK_{u,ap_1}\right)\right)\right\}$

用戶生成共享密鑰的簽章$Sig_{x_i}\left(SK_{u,ap_1}\right)$，再使用用戶的公鑰對簽章進行加密生成加密後的簽章$Enc_u\left(Sig_{x_i}\left(SK_{u,ap_1}\right)\right)$發送給 AP_1。

(6) AP_2 $\rightarrow$ UE

AP_2為了要驗證$Enc_u\left(Sig_{x_i}\left(SK_{u,ap_1}\right)\right)$的資料，先向 UE 要求一把代理重新加密的 Re-encrypted key (rk)，使他可以使用自己的私鑰解出內容。

(7) UE $\rightarrow$ AP_1

用戶使用自己的私鑰和AP_2的公鑰生成的一個代理重加密密鑰 $r_{u\rightarrow ap_2}$。

(8) AP_1 $\rightarrow$ AP_2 : $\left\{Enc_{AP_2}\left(Sig_{x_i}\left(SK_{u,ap_1}\right)\right), T6\right\}$

當 AP_1 收到 $rk_{u\rightarrow ap}$ 後，把加密簽章 $Enc_u\left(Sig_{x_i}\left(SK_{u,ap_1}\right)\right)$ 重加密為 $Enc_{AP_2}\left(Sig_{x_i}\left(SK_{u,ap_1}\right)\right)$，並發送給 AP_2。

(9) AP_2 收到資料後，首先先驗證 T6 是否小於Δt。再來AP_2使用重加密簽章 $Enc_{AP_2}\left(Sig_{x_i}\left(SK_{u,ap_1}\right)\right)$後，使用自己的私鑰解密重加密的簽章數據，得到用戶原始簽章$Sig_{x_i}\left(SK_{u,ap_1}\right)$，並透過用戶的公鑰驗證該簽章的真實性和完整性。

(10) AP_2 $\rightarrow$ UE : $\{DH_{ap_2}\}$

在認證完成後，AP_2 將使用 Diffie-Hellman 密鑰交換協議來生成會話密鑰。AP_2 將生成一個隨機數 α 並計$DH_{ap_2} = g^a \ mod \ p$，並發送給 UE。

(11) UE $\rightarrow$ AP_2 : $\{DH_u\}$

UE 將生成一個隨機數 b 並計算$DH_u = g^b \ mod \ p$，並發送給AP_2。

(12) UE, AP_2 收到傳送來的 Diffie-Hellman 公鑰後，雙方使用對方的公鑰和自己的私鑰計算共享密鑰K。$K_{ap_2} = (DH_u)^a \ mod \ p$，$K_u = (DH_{ap_2})^b \ mod \ p$，由

於 Diffie-Hellman 協議的特性，$K_{ap_2} = K_u$，即共享密鑰K相同。

(13) 最終，UE 和 AP_2 計算會話密鑰 $SK_{u,ap_2} = \mathrm{H}(K \parallel r_u \parallel r_{ap_2} \parallel T5 \parallel T6)$。

(14) 當跨網路切片的身份認證和會話密鑰交換完成，我們將使用智能合約自動開始計費系統。系統將記錄用戶的開始使用時間。智能合約將檢查用戶的帳戶餘額是否足夠支付一天的服務，如果不足，通知用戶並拒絕服務。我們使用智能合約在每天固定時間檢查所有用戶，並自動扣除對應的日費用。如果在任何時間用戶的帳戶不足下一天的費用，系統將自動登出用戶，並通知用戶。而當用戶完成服務或服務被終止時，將進行結算。

5.5 Identity Revocation

當發現有惡意用戶出現時，6G 運營商通常需要定期維護和更新撤銷列表，這可能會帶來大量存取和通訊開銷。然而，在我們的方法中，我們利用變色龍雜湊來刪除區塊鏈上與惡意用戶相關的註冊信息，而不是定期維護撤銷列表。如果區塊鏈上沒有某用戶的公鑰，則該用戶可以被視為惡意用戶。為了刪除區塊鏈上的註冊訊息，6G 運營商會生成一個不包含惡意用戶註冊訊息的新交易，替換掉包含惡意用戶註冊訊息的原始交易。新交易將經由其他 6G 運營商驗證後，替換掉原始交易。假設 O_A 發現一個惡意用戶 U^*，具體身份撤銷流程如下：

(1) 如果O_A需要撤銷用戶U^*，並其註冊訊息包含在TX_A中，O_A生成一個新的訊息 m'，而 $m' = \{ID_{O_A}, \lambda_{1j}, \lambda_{2j}, \dots, \lambda_{(i-1)j}, Sig_{TK_{O_A}}\{\mathrm{H}(w'_i)\}$，其中 $\lambda_{kj} = \{ID_{U^*}, X_{U^*}\}$，$w'_i = \{ID_{O_A}, \lambda_{1j}, \lambda_{2j}, \dots, \lambda_{(i-1)j}\}$，且 m' 和 m 相同，只是不包含 U^* 的註冊訊息。

(2) O_A 使用其私有陷門密鑰s_{O_A}生成新消息m'的變色龍雜湊碰撞。選擇一個隨機值 $k \in [1, q-1]$，計算更新後的 $r' = h + (g^k mod\ p) mod\ q$，$s' = k - \mathrm{H}(m' \parallel r') \cdot TK_{O_A}\ mod\ q$

(3) 執行完變色龍雜湊碰撞後，O_A可以獲得r'和s'，然後O_A生成一個新的交易

$TX'_A = \{m', h\}$，並將他廣播給其他 6G 運營商。

(4) 當其他 6G 運營商收到新的交易時，首先驗證h是否等於$r' - (HK_{O_A}{}^{(\mathrm{H}(m'\|r')} \cdot g^{s'} mod\ p)\ mod\ q$。如果相等，其他 6G 運營商將儲存新交易$TX'_A$，並刪除原始交易$TX_A$。