

|  |  |
| --- | --- |
| **Number** | **Description** |
| 1 | Mobile application requests public key info from IB Application. Once received it saves the public key info string within the application session.   1. IB Application requests the public key info by calling Salt RemotePIN REST API ***GetPubliccKeyInfo*** 2. Salt RemotePIN service retrieves the public key info, which cached within the service session at the start-up process. 3. Returns the public key info to IB application, which then gets returned to the mobile application. |
| 2 | Mobile application prompts user to select their PIN |
| 3 | Mobile application calls Salt SDK to public key encrypt the PIN. Salt SDK first generates an intermediate PINBlock then utilises the public key to encrypt the PIN. |
| 4 | Mobile application sends the encrypted intermediate PIN block (***eIntPINBlock***) to IB application. |
| 5 | IB application retrieves the corresponding ***PAN***, and other transaction details (***Txn Info***) from Bank backend systems. |
| 6 | IB application calls Salt RemotePIN REST API ***SelectPIN*** along with ***eIntPINBlock***, ***PAN*** and ***Txn-Info*** as parameters. |
| 7 | Salt RemotePIN SelectPIN API utilises nShield Connect HSM to translates the eIntPINBlock.   1. HSM unwrap/decrypt the eIntPINBlock using the private key, uses PAN to generate ANSI PINBlock then wrap/encrypt with PEK (AS2805 session key) to generate ***ePINBlock.*** 2. Encodes AS2805 SelectPIN message using ***ePINBlock*, PAN*,* *Txn*-*Info***, etc. 3. Sends the AS2805 SelectPIN message request to Tandem/Interchange. 4. Tandem process the request and respond to Salt RemotePIN service with pass/fail status. |
| 8 | Salt RemotePIN service respond to IB application with SelectPIN pass/fail status, where IB application then passes the response to Mobile application. |