OATH Provisioning Sub-group
Requirement-Provisioning Protocol Matrix
Updated: October 30, 2006

Mandatory Requirements

Requirements	**CT-KIP (all variants)**	**DSKPP**
1. Web services protocol (or XML-based)	*Yes*	*Yes*
2. Supports OATH PSKC payload format	*Partial*[1]	*Yes*
3. Allows for different credential types including vendor-specific credential formats	*Yes*	*Yes*[2]
4. Allows for multiple credential provisioning to the same device (uniquely identifiable)	*Yes*	*Yes*
5. Supports password-based encryption (e.g., soft tokens)	*Yes*	*Yes*
6. Supports PKI-based encryption (e.g., USB tokens)	*Yes*	*Yes*
7. Supports pre-shared key encryption (e.g., smart cards/SIM)	*Yes*	*Yes*
8. Supports server-generated key delivery	*Yes*	*Yes*
9. Supports mutual client-server key generation	*Yes*	*No*
10. Does not rely on transport level encryption (e.g., TLS) for seed protection	*Yes*	*Yes*
11. Supports OTA delivery to mobile devices (for soft token app or SIM)	*Yes*	*Yes*
12. Supports Internet delivery to PC/USB.	*Yes*	*Yes*
13. Supports credential renewal on existing token/device (same or new token ID, new key)	*Yes*	*Yes*[3]
14. Supports credential expiration (allowing for token licensing based on time).	*Yes*	*Yes*[4]
15. Supports credential replacement in case of stolen/lost device	*Yes*	*Yes*
16. Supports user authentication prior to provisioning	*Yes*[5]	*Yes*
17. Supports device authentication (based on device cert)	*Yes*[6]	*Yes*[7]
18. Extensible to support new algorithm specific configuration data (OATH HOTP, OCRA, SecurID and others)	*Yes*	*Yes*[8]

[1] CT-KIP is capable of handling PSKC through extension payload.

[2] Supported via PSKC extensions for vendor-specific algorithms under OTP type.

[3] The current draft allows for credential renewal using either a new token ID or keeping the existing ID (allows for flexibility in implementation).

[4] When PSKC is used for the credential payload

[5] CT-KIP user authentication handled through initial user authentication followed by a trigger message containing a nonce which then is part of the ClientHello [rsa].

[6] 4-pass supports implicit device authentication through the shared key variant (no other device than the one with the key will get access to the credential).Also, Internet-Draft http://www.ietf.org/internet-drafts/draft-doherty-ct-kip-ws-00.txt suggests an alternative mechanism for doing device client authentication.

[7] Supported via device certificate.

[8] Algorithm-specific data can be added to DSKPP (for request) and PSKC (for response/payload).

19. Allows client to specify device capabilities and preferences in requests	*Yes*	*Yes*
20. Allows server to deliver user interface attributes in response (e.g. logo)	*Yes*	*Yes*
21. Negotiation of supported/desired key types	*Yes*	*Yes*
22. Negotiation of MAC algorithms	*Yes*	*No*[9]
23. Negotiation of Encryption algorithms	*Yes*	*Yes*

Desirable Requirements

Requirement	**CT-KIP (all variants)**	**DSKPP**
24. Supports token deletion and notification to server	*No*	*No*
25. Supports credential transfer from one device to another (device upgrade).	*No*	*No*
26. Support device confirmation to server upon credential delivery.	*No*	*No*
27. Key validation option upon credential delivery.	*Yes*[10]	*No*[11]
28. Allow for trigger message to couple previous browsing session to start of protocol	*Yes*[12]	*No*
29. HTTP binding	*Yes*	*Partial*[13]

[9] MAC algorithm negotiation to be supported in next draft.

[10] Yes for 4-pass as server's message confirms it uses the same credential as the client. In two and one-pass CT-KIP, there is key confirmation from the server due to the K_MAC being sent wrapped with K_TOKEN. 4-pass CT-KIP should be changed in a similar manner.

[11] Currently viewed as not in scope of protocol – could be added.

[12] Yes for CT-KIP 4- and 2-pass, N/A for 1-pass.

[13] Supports simple http binding, but without defining a new header type.