



## RSA SecurID Ready Implementation Guide

Last Modified: June 23, 2011

### Partner Information

---

| Product Information |  |
|---------------------|--|
| Partner Name        | Openstream, Inc  |
| Web Site            | <a href="http://www.openstream.com">http://www.openstream.com</a>  |
| Product Name        | Cue-me™  |
| Version & Platform  | 1.2 (iOS 4.2 and Android 2.x)  |
| Product Description | Cue-me™ Multimodal Platform enables Secure Enterprise Applications on all popular mobile devices, tablets, laptops and desktop platforms |
| Product Category    | Wireless Products, Operating Systems   |

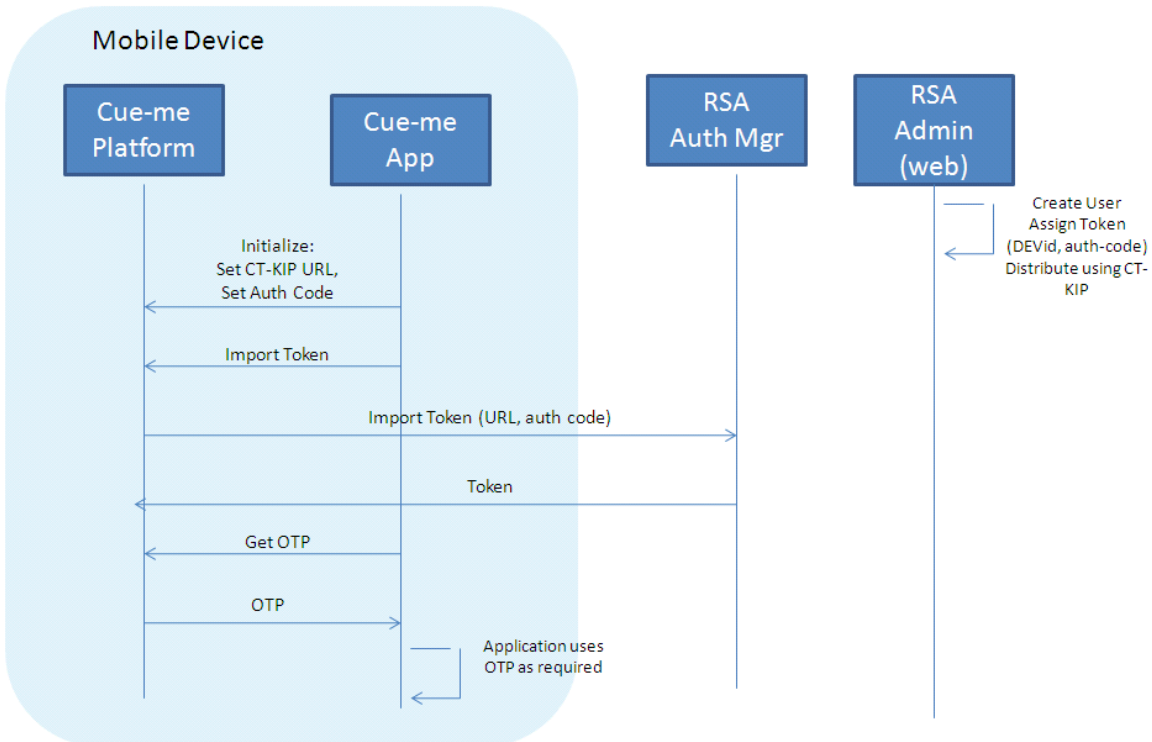




## Solution Summary

| Functional Description   |           |
|--|-----------|
| Authenticator provides its own GUI to present tokencode  | N/A (SDK) |
| Authenticator can securely store token seed record   | N/A (SDK) |
| Authenticator supports copy/paste of tokencode   | N/A (SDK) |
| Authenticator supports multiple seed records   | No        |
| Authenticator supports passphrase protection of application  | N/A (SDK) |
| Authenticator provides RSA Software Token Automation (user enters only PIN to authenticate)        | Yes       |
| Partner product provisions Authenticator (creates account, assigns token, delivers seed to device) | No        |
| Authenticator supports CT-KIP provisioning protocol  | Yes       |

### Interaction Flow

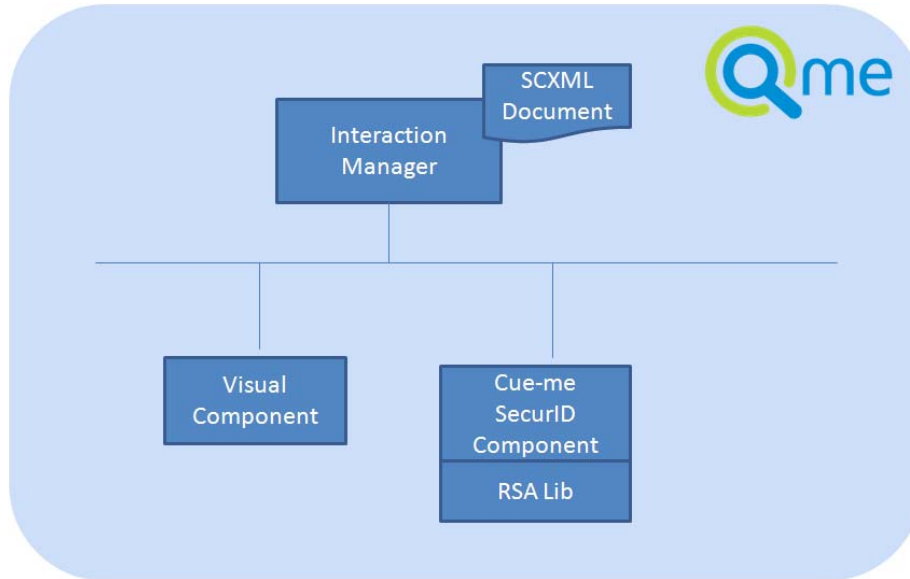


Please refer the “SCXML Code Snippets” section below for details of how token is provisioned and used on the device.



## Product Configuration for Interoperability

---



Cue-me™ Applications interact with RSA SecurID by raising SCXML events to the Cue-me SecurID Component (x-securid). This component is the integration point that allows an application to generate a SecurID Passcode in order to perform two-factor authentication. The events supported are:

### Events received and handled by the component

- setCtKipUrl – set the address of the CT-KIP service
- setCtKipAuthCode – set software token's activation code
- importCtKipToken – import software token using CT-KIP
- getCurrentOtp – request to get the current OTP
- getNextOtp – request to get the next OTP
- deleteToken – delete the provisioned software token
- getIsTokenProvisioned – get whether or not a software token is provisioned

### Events generated by the component


- currentOtp – contains the current OTP
- nextOtp – contains the next OTP
- importCtKipTokenSuccessful



- importCtKipTokenFailed – contains failure description
- deleteTokenSuccessful
- deleteTokenFailed – contains failure description
- isTokenProvisioned – contains whether or not a software token is currently provisioned
- error – contains error details

Mobile devices are provisioned in the field using CT-KIP. The Cue-me™ Client (with SecurID component) software needs to be downloaded on the device. Credential provisioning is performed via the CT-KIP dynamic seeding protocol.

---

 **Note: Openstream Cue-me is a software control framework for applications. All interaction is via SCXML events as described above, and there is no specific “application” screen shot. The screens provided below are therefore examples provided for informational purposes only.**

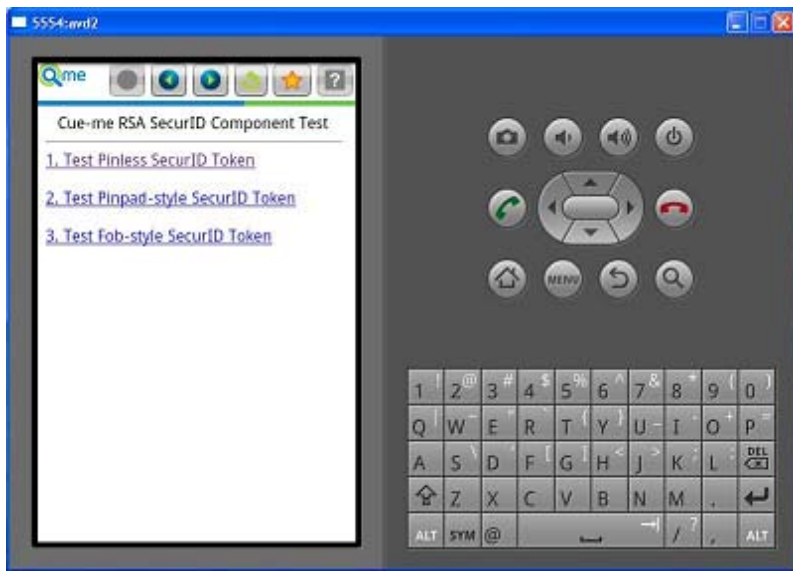
---

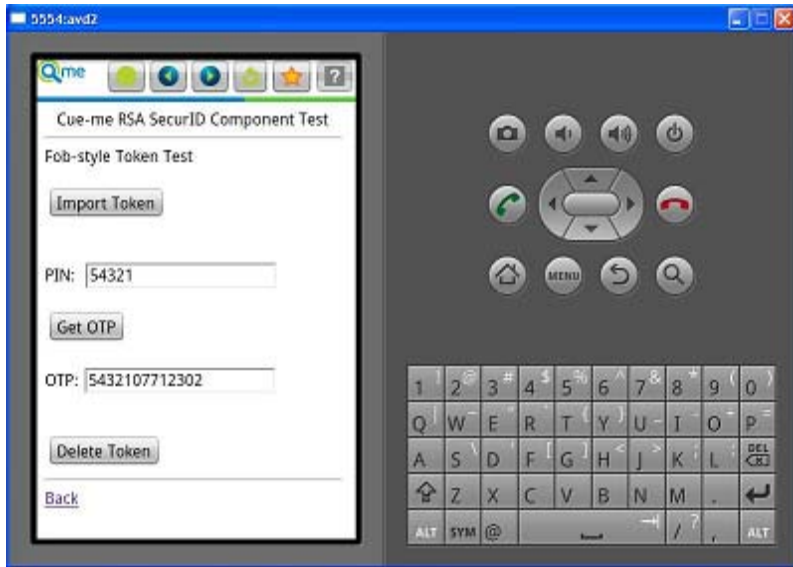
## iOS Screenshots





## Android Screenshots







## **SCXML Code Snippets**

### **1. Import Token via CT-KIP**

```
<send event="setCtKipUrl" to="x-securid" data="https://<ctkip service url>" />
```

```
<send event="setCtKipAuthCode" to="x-securid" data="<authcode>" />
```

```
<send event="importCtKipToken" to="x-securid" />
```

### **2. Get OTP**

```
<go on="click" from="x-html" node="getOtp_id">
```

```
    <send event="getField" to="x-html" target="securIdPin_id" />
```

```
</go>
```

```
<go on="getFieldResponse" from="x-html" if="event.name=='securIdPin_id'">
```

```
    <send event="getCurrentOtp" data="event.value" to="x-securid" />
```

```
</go>
```

```
<go on="currentOtp" from="x-securid">
```

```
    <send event="execute" to="x-html" target="setOtp" data="event.value" />
```

```
</go>
```

### **3. Delete Token**

```
<send event="deleteToken" to="x-securid" />
```

# Certification Checklist for 3<sup>rd</sup> Party Applications

Date Tested: June 23<sup>rd</sup>, 2011

| Product                    | Operating System        | Tested Version |
|----------------------------|-------------------------|----------------|
| RSA Authentication Manager | Windows Server 2003     | 7.1            |
| Cue-me™ Platform           | IOS 4.2 and Android 2.x | 1.2            |

| RSA SecurID Ready Authenticator Criteria**   |                                     |                                     |                                     |
|--|-------------------------------------|-------------------------------------|-------------------------------------|
| <b>RSA Software Token Import</b>   |                                     |                                     |                                     |
| v3.0 (AES) software token seed   | <input checked="" type="checkbox"/> | v3.0 copy & password-protected seed | <input type="checkbox"/>            |
| v3.0 (AES) password-protected seed   | <input type="checkbox"/>            | v3.0 (AES) multi-token seed file    | <input type="checkbox"/>            |
| v3.0 (AES) copy-protected seed   | <input type="checkbox"/>            | v3.0 (AES) pinless token            | <input checked="" type="checkbox"/> |
| <b>RSA Software Token SDK or Embedded RSA OTP Algorithm</b>                            |                                     |                                     |                                     |
| Strong encryption of seed database   |                                     |                                     | <input type="checkbox"/>            |
| Copy protection of seed database   |                                     |                                     | <input type="checkbox"/>            |
| Proper display of current tokencode  |                                     |                                     | <input type="checkbox"/>            |
| Interface to enter PIN   |                                     |                                     | <input type="checkbox"/>            |
| Proper display of current PASSCODE   |                                     |                                     | <input type="checkbox"/>            |
| Proper display of lifetime of current code   |                                     |                                     | <input type="checkbox"/>            |
| Successful removal of installed token(s)   |                                     |                                     | <input checked="" type="checkbox"/> |
| Successful re-provisioning of installed token(s)                                       |                                     |                                     | <input checked="" type="checkbox"/> |
| Proper display of token serial number  |                                     |                                     | <input type="checkbox"/>            |
| Successful addition of token alias/nickname  |                                     |                                     | <input type="checkbox"/>            |
| Successful rename/removal of token alias/nickname                                      |                                     |                                     | <input type="checkbox"/>            |
| Passphrase protection of application or token  |                                     |                                     | <input type="checkbox"/>            |
| Proper setting of default token  |                                     |                                     | <input type="checkbox"/>            |
| Ability to copy/paste PASSCODE   |                                     |                                     | <input type="checkbox"/>            |
| Successful authentication using partner device   |                                     |                                     | <input checked="" type="checkbox"/> |
| Partner product displays RSA SecurID Ready logo  |                                     |                                     | <input type="checkbox"/>            |
| <b>RSA Software Token Automation (SoftID API)</b>                                      |                                     |                                     |                                     |
| SoftID API-enabled application can automatically extract PASSCODE from Partner product |                                     |                                     | <input type="checkbox"/>            |
| Successful authentication using partner device and SoftID API-enabled application      |                                     |                                     | <input type="checkbox"/>            |
| <b>RSA Software Token Provisioning (CT-KIP)</b>  |                                     |                                     |                                     |
| Partner product can be successfully seeded via CT-KIP protocol                         |                                     |                                     | <input checked="" type="checkbox"/> |
| <b>RSA Software Token Provisioning (RSA Authentication Manager Administrative API)</b> |                                     |                                     |                                     |
| Partner product provisions Authentication Manager username                             |                                     |                                     | <input type="checkbox"/>            |
| Partner product provisions RSA Software Token assignment                               |                                     |                                     | <input type="checkbox"/>            |
| Partner product provides delivery mechanism for Software Token (.SDTID)                |                                     |                                     | <input type="checkbox"/>            |

JEC/PAR

✓ = Pass ✗ = Fail N/A = Non-Available Function

\*\*Openstream Cue-me is an application framework, therefore it is the responsibility of the third-party app to implement additional functionality such as displaying the passcodes and/or managing tokens if needed.





## Appendix

---

SCXML : State Chart XML (<http://www.w3.org/TR/scxml/>)

Multimodal Interaction – Based on W3C MMI standards (<http://www.w3.org/TR/mmi-arch/>)