



www.boku.com

SECURITY IMPLEMENTATION GUIDE

Guidance about Passwords and Security Hashing

Version
2013-08-05

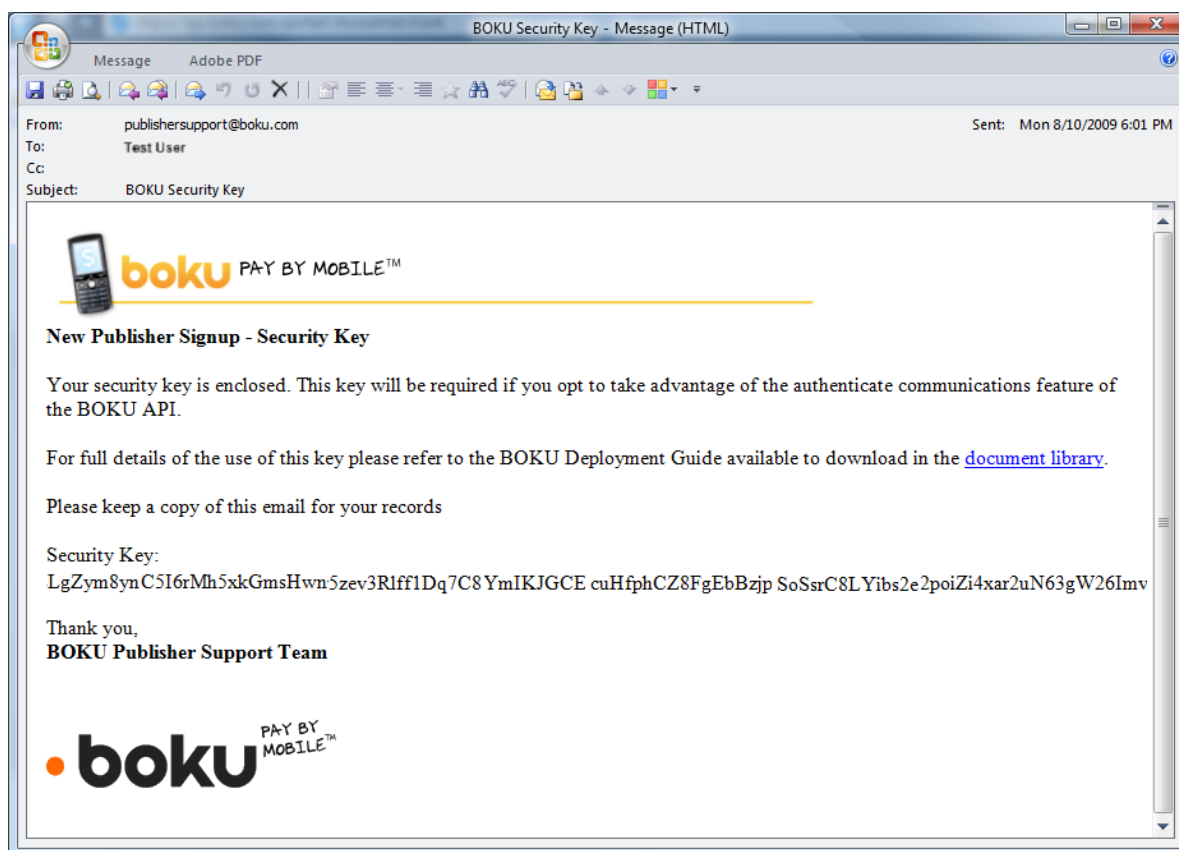
Table of Contents

TABLE OF CONTENTS	2
1 INTRODUCTION	3
2 API PASSWORD VS PORTAL LOGIN PASSWORD.....	4
3 API SIGNATURE AUTHENTICATION	6
3.1 SECURITY RECOMMENDATIONS FOR API CALLS & CALLBACKS	6
3.2 NOTES ABOUT SIGNING API CALLS FOR SECURITY.....	6
3.3 API AUTHENTICATION METHODS	7
4 WORKING WITH MD5 SIGNATURES.....	8
4.1 CREATING SIGNATURES ON FORM-BASED API REQUESTS.....	8
4.2 CREATING SIGNATURES ON XML-BASED POST REQUESTS.....	10
4.3 TO VERIFY A SIGNATURE IN A CALLBACK URL	12
4.4 TO VERIFY A SIGNATURE IN AN XML RESPONSE	14
5 SSL CERTIFICATES ON YOUR CALLBACK SERVERS.....	16
6 FRAUD CONTROLS & SPEND LIMITS.....	17
7 ADDITIONAL SECURITY INFORMATION	18
CHANGE HISTORY	19

1 Introduction

Security is always a concern especially when it comes to payment platforms. As a security measure, the Boku API has a MD5 hash function to sign all communications between Boku and the publisher. MD5 (Message-Digest Algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value.

As part of the sign-up process, all Boku publishers receive an MD5 (API) security key that can be used to authenticate communications in either direction. Below is an example signup message containing this security key. This document will explain how to properly use the MD5 security key to secure communications between the publisher and the Boku servers.



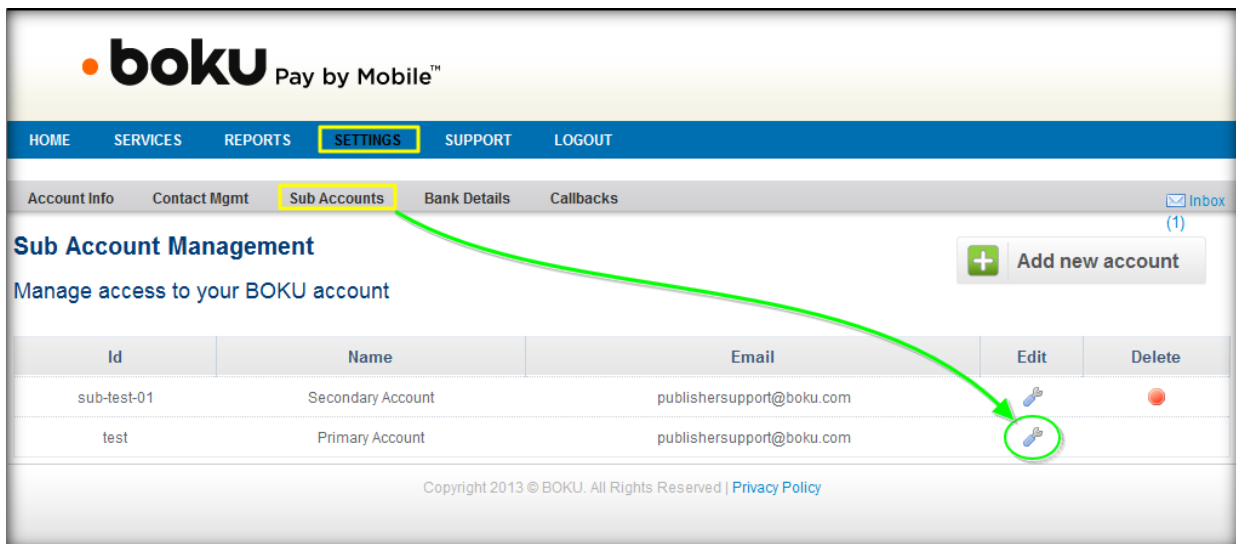
2 API Password vs Portal Login Password

Your primary Boku Publisher Portal account actually has two passwords associated with the same account ID. These passwords are:

- API password for sending unsigned API calls
- Portal password to log into the Boku Publisher Portal

When you first register for your Boku Publisher Portal account, both of these passwords are set concurrently to the value you entered in the registration page. However, you can change these passwords individually and it is recommended that if you do not plan to securely sign your API calls with a MD5 hash, then you should change your API password so that it is different from your portal password. Please be careful when changing the password to ensure that you change the correct password.

The configuration screens for changing the API and portal login passwords are found in the 'Settings > User Accounts' menu after logging into your Boku Publisher Portal account. The primary account will not provide the option to "delete" that account. If you create secondary portal login accounts, you will see the "delete" option will be available for those secondary accounts.



boku Pay by Mobile™

HOME SERVICES REPORTS **SETTINGS** SUPPORT LOGOUT

Account Info Contact Mgmt **Sub Accounts** Bank Details Callbacks Inbox (1)

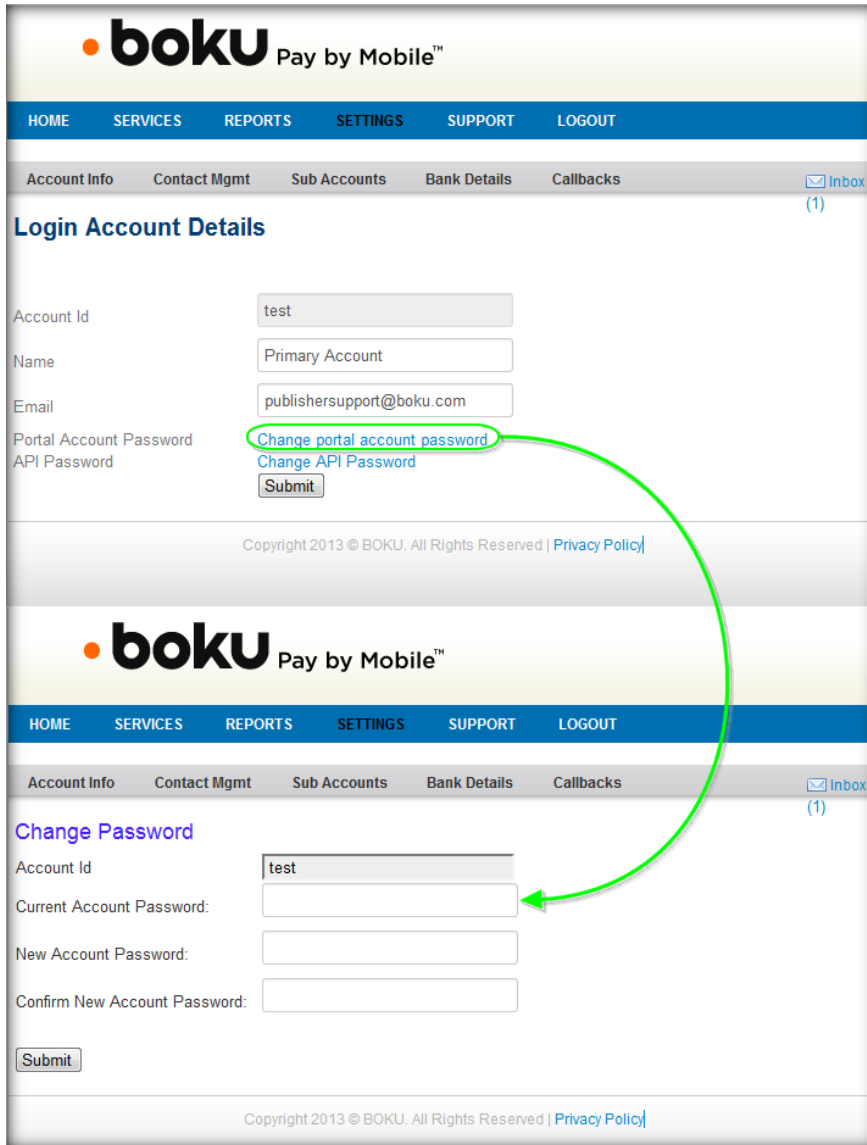
Sub Account Management
Manage access to your BOKU account

[+ Add new account](#)

Id	Name	Email	Edit	Delete
sub-test-01	Secondary Account	publishersupport@boku.com		
test	Primary Account	publishersupport@boku.com		

Copyright 2013 © BOKU. All Rights Reserved | [Privacy Policy](#)

When you edit the primary account details by clicking on the corresponding spanner/wrench icon, you will see two separate links for changing the password. Click the proper link to proceed with your password change action. For the primary account, you must enter the correct current password in order to successfully change to a new portal or API password.



The image displays two screenshots of the Boku Pay by Mobile web interface, illustrating the process of changing a password for a primary account.

Top Screenshot: Login Account Details

- Header:** boku Pay by Mobile™
- Navigation Bar:** HOME, SERVICES, REPORTS, SETTINGS, SUPPORT, LOGOUT
- Sub Navigation Bar:** Account Info, Contact Mgmt, Sub Accounts, Bank Details, Callbacks, [Inbox \(1\)](#)
- Section:** Login Account Details
- Form Fields:**
 - Account Id: test
 - Name: Primary Account
 - Email: publishersupport@boku.com
 - Portal Account Password: [Change portal account password](#)
 - API Password: [Change API Password](#)
 - Submit button
- Footer:** Copyright 2013 © BOKU. All Rights Reserved | [Privacy Policy](#)

Bottom Screenshot: Change Password

- Header:** boku Pay by Mobile™
- Navigation Bar:** HOME, SERVICES, REPORTS, SETTINGS, SUPPORT, LOGOUT
- Sub Navigation Bar:** Account Info, Contact Mgmt, Sub Accounts, Bank Details, Callbacks, [Inbox \(1\)](#)
- Section:** Change Password
- Form Fields:**
 - Account Id: test
 - Current Account Password: [Input Field]
 - New Account Password: [Input Field]
 - Confirm New Account Password: [Input Field]
 - Submit button
- Footer:** Copyright 2013 © BOKU. All Rights Reserved | [Privacy Policy](#)

A green arrow points from the [Change portal account password](#) link in the top screenshot to the **Current Account Password** input field in the bottom screenshot, indicating the next step in the process.

For secondary accounts, you only get the option to click a button which will send an email containing a reset portal password link. Be careful because clicking that button will automatically trigger the email message without any additional warning message.

3 API Signature Authentication

3.1 Security recommendations for API calls & callbacks

The security recommendations for Boku API calls and callback notifications are listed below:

API Call or Callback Notification	MD5 Signature
lookup API call	Recommended
price API call	Mandatory
prepare API call	Recommended
verify-trx-id API call	Recommended
report API call	Recommended
service-prices API call	Mandatory
sub-cycle API call	Recommended
sub-cancel API call	Recommended
Callback Notifications	Mandatory

3.2 Notes about signing API calls for security

Keep in mind the following points about using the API authentication function.

- Currently, publishers are only required to sign 'price' API call requests but it is recommended that publishers take advantage of this feature for all communications with Boku servers. Publishers should sign requests to the 'prepare' API as well as the 'price' and 'verify-trx-id' API calls if used.
 - If an API call is signed, Boku will authenticate the signature against the pre-agreed API security key (which was sent upon account approval).
 - If the request is not signed correctly or the timestamp differs from NTP by a delta greater than **300 seconds** (5 minutes) the request will be rejected with one of the following error codes/messages:
 - Error code '28'- 'Invalid Signature'
 - Error code '32'- 'Bad Bind Credentials'
 - Boku will continue to respond to unsigned requests as normal.
- XML responses to the 'prepare' and 'verify-trx-id' API requests will be signed only if the publisher has signed the initial request. Alternatively, the publisher can authenticate using the 'password' parameter if desired.

3. All callback notifications (e.g. event and final billing status) will be signed.
 - It is recommended that publishers authenticate the callback.
 - If the callback is not signed correctly or the time stamp differs from NTP by a delta greater than **300 seconds**, the callback should not be trusted.
4. The two parameters in the following table are used specifically for API authentication and both parameters must be reported in lieu of the "password" parameter when signing an API request.

Field	Description	Online Reference Tool
timestamp	UNIX timestamp (UTC).	http://www.epochconverter.com/
sig	MD5 hash computation.	http://www.md5hashgenerator.com/

3.3 API authentication methods

Only the primary account can be used to perform API calls. There are two ways to make an API call request to Boku servers which are indicated below.

- A) Non-secure, password authentication is performed by reporting the following parameters:
 - **merchant-id** (primary account)
 - **password** (for API function)
- B) Secure authentication is performed using a cryptographic hash function* to sign all communications between Boku and the publisher. In this case, the following parameters are used:
 - **merchant-id** (primary account)
 - **timestamp** (needs to be within 300 seconds of when API call request is sent)
 - **sig** (MD5 hash value of which generation steps are indicated in the following chapter)

* Following account registration approvals, all BOKU publishers receive a security key that allows the generation of the "sig" cryptographic hash value which is used to authenticate communications in either direction

4 Working with MD5 Signatures

4.1 Creating signatures on form-based API requests

1. Take all the parameter/value pairs from the URL-decoded query string. Be sure to **exclude** the following items:
 - the base URL
 - the "password" parameter
 - any parameters with empty values (**NOTE:** parameters reporting a value of zero (0) should not be removed).

```
action=verify-trx-id&trx-id=ace98a6f2043cac883558d79&merchant-id=testpublisher
```

2. Add a new parameter/value pair called timestamp with a value in a Unix epoch timestamp notation (be sure that the timestamp is 10-digits long).

```
action=verify-trx-id&trx-id=ace98a6f2043cac883558d79&merchant-id=testpublisher&timestamp=1225911804
```

3. Sort the parameter/value pairs alphabetically based on the parameter field name and maintain the parameter/value association.

```
_action=verify-trx-id&_merchant-id=testpublisher&_timestamp=1225911804&_trx-id=ace98a6f2043cac883558d79
```

4. Concatenate the parameter/value pairs into one string with no delimiters and append the API security key at the end. (eg. < parameter ><value>< parameter ><value><api security key>).

```
actionverify-trx-idmerchant-idtestpublishertimestamp1225911804
trx-idace98a6f2043cac883558d79gvXHls51BaDhqUpfCsUgjsUYn8Xpp5Yh
AF4tTCHePyw4jM0PmukUp3GcQnlansRVXtpeCyITHAsChEWcNmJbGN3mliOM6
1048vU
```


5. Encode the string from the prior step into UTF-8 and hash it using the MD5 algorithm.

NOTE: Your calculated MD5 hash algorithm should be in lowercase. Some server platforms (i.e. Railo) generate the hash value in uppercase.

```
b57eda6c3fba5cfe98baaca66d306254
```

6. Generate the full API call URL with the parameters in any order and add the "sig" parameters using the value created in the previous step.

NOTE: When submitting API calls to Boku HTTP(S) interfaces you are required to URL-encode all parameter values

```
https://api2.boku.com/billing/request?action=verify-trx-id&tr  
x-id=ace98a6f2043cac883558d79&merchant-id=testpublisher&times  
tamp=1225911804&sig=b57eda6c3fba5cfe98baaca66d306254
```

4.2 Creating signatures on XML-based POST requests

1. Take the full XML from the body of the POST request.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dummy-request>
  <Xparam>valueX</Xparam>
  <Yparam>valueY</Yparam>
  <Zparam>valueZ</Zparam>
  <Aparam>valueA</Aparam>
  <Bparam>valueB</Bparam>
  <Cparam>valueC</Cparam>
  <timestamp>1371600000</timestamp>
</dummy-request>
```

2. Keep only the parameter name/value pairs and remove all the XML markup, XML parent node tags, closing tags, and extra whitespaces.

```
XparamvalueX
YparamvalueY
ZparamvalueZ
AparamvalueA
BparamvalueB
CparamvalueC
timestamp1371600000
```

3. Sort the parameter/value pairs alphabetically based on the parameter field name and maintain the parameter/value association.

```
AparamvalueA
BparamvalueB
CparamvalueC
timestamp1371600000
XparamvalueX
YparamvalueY
ZparamvalueZ
```

4. Concatenate the parameter/value pairs into one string and append the API security keys at the end. (eg. < parameter ><value>< parameter ><value><api security key>).

```
AparamvalueABparamvalueBCparamvalueCtimestamp1371600000Xp
aramvalueXYparamvalueYZparamvalueZgvXHls51BaDhqUpfCsUgjsU
Yn8Xpp5YhAF4tTCHePyw4jM0PmukUp3GcQnlansRVXtpeCyITHAsChEW
CnMJbGN3mliOM6lO48vU
```

5. Encode the string from the prior step into UTF-8 and hash it using the MD5 algorithm.

NOTE: Your calculated MD5 hash algorithm should be in lowercase. Some server platforms (i.e. Railo) generate the hash value in uppercase.

```
71da906c24a7511e3c5ce66b9ef980d7
```

6. Regenerate the full XML body of the POST request with the parameters in any order and add the "sig" tag to the XML body using the value created in the previous step.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dummy-request>
  <Xparam>valueX</Xparam>
  <Yparam>valueY</Yparam>
  <Zparam>valueZ</Zparam>
  <Aparam>valueA</Aparam>
  <Bparam>valueB</Bparam>
  <Cparam>valueC</Cparam>
  <timestamp>1371600000</timestamp>
  <sig>71da906c24a7511e3c5ce66b9ef980d7</sig>
</dummy-request>
```

4.3 To verify a signature in a Callback URL

1. Start with the full URL-decoded callback notification URL.

```
https://your-domain.com/callback.php?action=billingresult&trx-id=b8b2db3f0117e53b6bdef56e&test=1&result-code=0&result-msg=Ok - Transaction successful&merchant-ref=test ref 12345&content-id=test id&mobilenumber=98765432100&paid=300&amount=300&currency=GBP&locale=en_GB&receivable-gross=184&receivable-net=147&reference-currency=USD&reference-amount=535&reference-paid=535&reference-receivable-gross=328&reference-receivable-net=262&timestamp=1225911804&sig=c8cac6b131f22ef50876a9eb64f2a1e6
```

2. Take all the parameter/value pairs from the query string. Make sure to exclude the following items:
 - the base URL
 - the signature parameter
 - any parameters with empty values (**NOTE:** parameters reporting a value of zero (0) should not be removed).

```
action=billingresult&trx-id=b8b2db3f0117e53b6bdef56e&test=1&result-code=0&result-msg=Ok - Transaction successful&merchant-ref=test ref 12345&content-id=test id&mobilenumber=98765432100&paid=300&amount=300&currency=GBP&locale=en_GB&receivable-gross=184&receivable-net=147&reference-currency=USD&reference-amount=535&reference-paid=535&reference-receivable-gross=328&reference-receivable-net=262&timestamp=1225911804
```

3. Sort the parameter/value pairs alphabetically based on the parameter field name and maintain the parameter/value association.

```
action=billingresult&amount=300&content-id=test id&currency=GBP&locale=en_GB&merchant-ref=test ref 12345&mobilenumber=98765432100&paid=300&receivable-gross=184&receivable-net=147&reference-amount=535&reference-currency=USD&reference-paid=535&reference-receivable-gross=328&reference-receivable-net=262&result-code=0&result-msg=Ok - Transaction successful&test=1&timestamp=1225911804&trx-id=b8b2db3f0117e53b6bdef56e
```

4. Remove all the delimiters between the parameters as well as the field names and parameter values. (e.g. <parameter><value><parameter><value>).

```
actionbillingresultamount300content-idtest_idcurrencyGBPlocale
en_GBmerchant-reftest_ref12345mobilenumber98765432100paid300r
eceivable-gross184receivable-net147reference-amount535referenc
e-currencyUSDreference-paid535reference-receivable-gross328ref
erence-receivable-net262result-code0result-msgOk - Transaction
successfultesttimestamp1225911804trx-idb8b2db3f0117e53b6bdef
56e
```

5. Concatenate the parameter/value pairs into one string and append the API security key to the end. (e.g. <parameter><value><parameter><value><API security key>).

```
actionbillingresultamount300content-idtest_idcurrencyGBPlocale
en_GBmerchant-reftest_ref12345mobilenumber98765432100paid300r
eceivable-gross184receivable-net147reference-amount535referenc
e-currencyUSDreference-paid535reference-receivable-gross328ref
erence-receivable-net262result-code0result-msgOk - Transaction
successfultesttimestamp1225911804trx-idb8b2db3f0117e53b6bdef
56e_gvXHls51BaDhqUpfCsUgjsUYn8Xpp5YhAF4tTCHePyw4jM0PmukUp3GcQn1
nansRVXtpeCyITHAsChEWCnMJbGN3m1iOM61O48vU
```

6. Encode the string from the prior step into UTF-8 and hash it using the MD5 algorithm.

NOTE: Your calculated MD5 hash algorithm should be in lowercase. Some server platforms (i.e. Railo) generate the hash value in uppercase.

```
c8cac6b131f22ef50876a9eb64f2a1e6
```

7. Compare the outcome from the previous step with the signature reported from the original query string. If they differ, the callback should not be trusted.
8. Compare the timestamp parameter from the query string with the current time (in Unix Epoch time). If they differ by more than 300 seconds, the callback should not be trusted.

4.4 To verify a signature in an XML response

1. Extract the signature from the HTTP header X-PAYMO-RESPONSE-SIGNATURE and save it for the purpose of comparison in the last step.

```
HTTP/1.0 200 OK
Content-Type: text/xml; charset=utf-8
Cache-Control: max-age=1
X-PAYMO-RESPONSE-SIGNATURE: 0f545f81ba96e38342367add6f492e1c
Content-Length: 613
Server: Jetty(6.1.18)
X-Cache: MISS from http02.local-paymo.net
X-Cache-Lookup: MISS from http02.local-paymo.net:80
Via: 1.0 http02.local-paymo.net:80 (squid/2.6.STABLE21)
Connection: keep-alive
```

2. Append the API security key to the end of the entire XML document as one string with no delimiters. Be sure that you are capturing the actual XML response body.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<prepare-request>
  <action>prepare</action>
  <trx-id>8c3f15cd6e0ace69a231a628</trx-id>
  <result-code>0</result-code>
  <result-msg>Operation Successful</result-msg>
  <button-markup>&lt;!-- Begin BOKU Buy-Button Code --&gt;
&lt;div&gt;
&lt;script language="JavaScript"
src="https://buy.boku.com/checkoutbutton/8c3f15cd6e0ace69a23
1a628/buy.js"&gt;&lt;/script&gt;
&lt;/div&gt;
&lt;!-- End BOKU Buy-Button Code --&gt;
</button-markup>
  <buy-
url>https://buy.boku.com/checkoutidentify/8c3f15cd6e0ace69a2
31a628/buy.js</buy-url>
</prepare-request>
gvXHls51BaDhqUpfCsUgjsUYn8Xpp5YhAF4tTCHePyw4jM0PmukUp3GcQn1n
ansRVXtpeCyITHAsChEWcNjMbGN3mliOM61O48vU
```

3. Take the string from the prior step and hash it using the MD5 algorithm.

NOTE: *Your calculated MD5 hash algorithm should be in lowercase. Some server platforms (i.e. Railo) generate the hash value in uppercase.*

```
0f545f81ba96e38342367add6f492e1c
```

4. Compare the outcome string from the prior string with the signature from the HTTP header (see the first step).

5 SSL Certificates on your Callback Servers

Boku-to-Publisher communications can be conducted using HTTP or HTTPS at the publisher's discretion. However, HTTPS calls to callback URLs which do not have a valid signed certificate will fail.

If you notice a problem where you are receiving callback notifications multiple times after switching from an HTTP callback URL to an HTTPS callback URL, please note that the Boku callback source servers cannot send callback notifications to a site with an untrusted and/or self-signed security certificate. Our servers will encounter a website security problem and the automated mechanism will not be able to manually override the untrusted certificate.

If however, you know that your HTTPS callback URL has a trusted SSL certificate installed, you may want to verify whether the SSL certificate was installed correctly. You can check this online at the Network4All site:

<http://www.networking4all.com/en/support/tools/site+check/>

6 Fraud Controls & Spend Limits

We do not recommend that publishers add separate velocity rules because this adds little value in terms of reducing fraud and only serves to prevent bona-fide consumers from spending at the level they want.

- Transaction amounts are much smaller than typical credit card transactions and most carriers apply their own velocity / fraud management rules (such as per-transaction, per-day and per-month spending limits for each consumer, which Boku manages on the publisher's behalf).
- In addition, all Boku transactions use two-factor authentication (which is better than a credit card transaction) and largely eliminates chargeback risk. If a handset is stolen, or if a consumer does attempt fraudulent chargebacks, the carrier typically blacklists that number and absorbs any loss as part of their 40-50% fee.
- Due to the low price points in many markets, mobile consumers are used to having to complete multiple transactions to gain enough credits to make purchases.

A further risk from publishers trying to apply credit card-style fraud detection policies to mobile micropayments is that publishers may inadvertently fail to fulfill a valid transaction, resulting in a manual refund charge.

The premise is that if Boku returns a callback notification to the publisher confirming a transaction as successful, then that consumer has been charged. If a publisher's internal fraud prevention rules subsequently reject the Boku transaction as fraudulent, the consumer will have successfully paid, but will not receive their points/coins. This typically causes a consumer complaint and the result is a high cost manual refund, the charges for which ultimately come back to the publisher for failing to properly fulfill a valid transaction.

In addition to enforced regulatory spend limits it is possible for publishers to set their own specific spend limits on a daily, weekly or monthly basis in any market. To take advantage of this feature, please contact your Boku account manager.

7 Additional Security Information

Security:

All communications are conducted using secure interfaces.

Publisher IDs and Passwords:

All publisher-to-Boku communications require the submission of either one of the following sets of information:

- username and password
- **--- OR ---**
- username, timestamp, and sig

The username, password, and API security key were issued during the publisher account setup process.

HTTPS:

All Publisher-to-Boku communications are encrypted using an SSL certificate signed by a valid certificate authority and installed on the Boku servers.

We run all of our server to server API connections over a standard wildcard SSL certificate with 256-bit encryption supplied by Network Solutions
(<https://www.networksolutions.com/SSL-certificates/index.jsp>)

On customer facing sites, such as the payment panel and the publisher services website, where personal details are captured, we use a 256-bit encryption EV SSL certificate supplied by Network Solutions.
(<https://www.networksolutions.com/SSL-certificates/ev-certificate.jsp>)

Change History

DATE	DESCRIPTION	RESPONSIBLE
2013/08-05	Updated '2: API Password vs Portal Login Password' chapter.	ACK
2013/07-26	Updated Online MD5 Hash URL	ACK
2013/06-18	Added new sub-chapter (4.2) on calculating the 'sig' value for XML-based POST requests.	ACK
2013/03-11	Updated any references from "BOKU" to "Boku" wherever possible.	ACK
2012/11-01	Updated response header and matching response body for section 4.3.	ACK
2012/10-29	Corrected errors in sections 4.1 (inconsistent 'trx-id') and 4.2 (incorrect reporting of 'reference-currency' parameter).	ACK
2010/11-09	Updated portal and API password change process.	ACK
2010/09-01	Updated Publisher Portal menu names.	ACK
2010/01-19	Added 'service-prices' and 'lookup' API calls to security recommendations list	ACK
2010/01-07	Updated copyright year.	ACK
2010/06-14	Updated 'service-prices' API call to require an MD5 signature.	ACK
2010/05-28	Added " SSL Certificates on your Callback Servers" chapter and some minor cleanup.	ACK
2010/04-20	Added note to use URL-decoded URLs.	ACK
2010/03-23	Added example for verifying signatures in an XML response.	ACK
2009/11-13	Document reformatted to new template.	ACK
2009/08-10	Document creation	ACK