Camellia Cipher Suites for TLS

<draft-kato-tls-rfc4132bis-02.txt>

IETF 72nd July 2008
TLS Working Group

Satoru Kanno
Back Ground

• RFC4132 Published July 2005
• Adapted several OSS
  – OpenSSL 0.9.8c or later
  – GnuTLS
  – Firefox 3.0
  – Linux
  – FreeBSD
  – Bouncy Castle
Connect to SSL Server by

Reference: Pay Pal(https://www.paypal-deutschland.de/haendler/was_ist_paypal.html)
Goal

• Our policy is to follow up AES's Cipher Suites.

The Camellia cipher

This cipher is recommended by the European Union NESSIE project, the Japanese CRYPTREC project, and was added to the SSL/TLS cipher list by RFC 4132. The Camellia algorithm will be in FireFox 3. It is not enabled by default in OpenSSL.

The Camellia home site mentions that there are export (from Japan) restrictions which may make Japanese OpenSSL distributors cautious, but these are general restrictions on all strong (64+ bit) cryptography. There is nothing camellia-specific about these Japanese export restrictions, so adding Camellia does not change the Japanese export situation.

[If you build OpenSSL for distribution to Japan or Europe, adding camellia is recommended]

Code:

```perl
PERL Configure VC-WIN32 enable-camellia
...
```


Bundled OpenSSL with enable-camellia

openSUSE 10.3, Fedora core 9 …

• As an action of this policy, we propose these Camellia Cipher Suites.
Proposal New Cipher Suites

<table>
<thead>
<tr>
<th>Proposal Camellia Cipher Suites</th>
<th>RFC4346-bis (AES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256</td>
<td>TLS_RSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256</td>
<td>TLS_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA256</td>
<td>TLS_DH_DSS_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA256</td>
<td>TLS_DH_RSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA256</td>
<td>TLS_DHE_DSS_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256</td>
<td>TLS_DHE_RSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA256</td>
<td>TLS_DH_anon_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA256</td>
<td>TLS_DH_DSS_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA256</td>
<td>TLS_DH_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA256</td>
<td>TLS_DHE_DSS_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256</td>
<td>TLS_DHE_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA256</td>
<td>TLS_DH_anon_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA256</td>
<td>TLS_DH_anon_WITH_AES_256_CBC_SHA256</td>
</tr>
</tbody>
</table>

The varieties are same!
What’s Next?

• We would like to be adopted this draft as a *WG item*.

Questions and comments?