

# SGC Connection Failures: Step-up and Export Ciphers Use Different Digests

Document ID: 45321

## Contents

### Introduction

#### Prerequisites

- Requirements
- Components Used
- Conventions

#### Problem

#### Solution(s)

- Solution 1
- Solution 2

#### Related Information

## Introduction

This document addresses a problem that occurs in the security provider Schannel.dll file, which is used in Microsoft Internet Information Server (IIS) and Microsoft Internet Explorer. This problem presents when you connect to a site that uses Server Gated Cryptography (SGC) to do high encryption, and the export cipher suite uses one hash algorithm while the domestic cipher suite uses another. In this situation, the Schannel.dll file occasionally selects the wrong algorithm, which results in a failed connection. As a result, Web clients may fail to connect to Web sites that use SGC for strong encryption when a secure connection is required. If either the Internet server or Web client is running Microsoft products, then the connection may fail.

Microsoft acknowledges that when a step-up cipher uses a different digest than the export cipher, the connection may fail. For more information on this problem, refer to [SGC Connections May Fail from Domestic Clients](#).

## Prerequisites

### Requirements

There are no specific requirements for this document.

### Components Used

The information in this document is based on these software and hardware versions:

- Cisco Content Services (CSS) with Secure Socket Layer (SSL) module

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

## Problem

With an SGC step-up cert on the CSS SSL module, when the client connects to a site through the SSL module with a 56-bit browser, the browser establishes an SSL connection at 56 rather than stepping up the connection to 128.

For example, imagine the first client hello negotiates a cipher of `rsa-export1024-with-rc4-56-sha`. The module matches based on the order in the configuration (unless the ciphers are weighted) so when the step-up occurs, the module probably tries to use a cipher of `rsa-with-3des-edc-cbc-sha`. The digests of these two ciphers do not match, and the failure occurs. Not only must the digests match, BUT the encryption types must match as well.

## Solution(s)

Based on the example customer proxy list, the solution(s) to this problem are explained in this section.

Currently, the customer has these export ciphers:

- `ssl-server 4`
- `ssl-server 4 vip address 198.22.10.10`
- `ssl-server 4 rsakey CSSRsaKey4`
- `ssl-server 4 rsacert RsaCert4`
- `ssl-server 4 cipher rsa-with-rc4-128-md5 198.22.10.10 20094`
- `ssl-server 4 cipher rsa-with-rc4-128-sha 198.22.10.10 20094`
- `ssl-server 4 cipher rsa-with-des-cbc-sha 198.22.10.10 20094`
- `ssl-server 4 cipher rsa-with-3des-edc-cbc-sha 198.22.10.10 20094`
- `ssl-server 4 cipher rsa-export1024-with-des-cbc-sha 198.22.10.10 20094`
- `ssl-server 4 cipher rsa-export1024-with-rc4-56-sha 198.22.10.10 20094`

To solve the problem discussed in this document, you must pick one export cipher to support (for example, `rsa-export1024-with-rc4-56-sha`). This is usually not a problem because if a 56-bit browser sends one of these ciphers, both are sent. You can now configure the rest of your strong ciphers, but you must weight them such that the cipher (`rsa-with-rc4-128-sha`) has the highest weight. The other strong ciphers must be assigned the next strongest weights, and the export cipher the lowest weight. Here is a sample of what this configuration looks like (note that the export cipher has no weight as the default is 1):

**Note:** In this example, you have two options regarding which export cipher suite to use. Cisco cannot recommend which one to use. You must make a decision based on your business security requirements.

### Solution 1

If you decide to use the export cipher (`rsa-export1024-with-rc4-56-sha`), the proxy list looks like this:

- `ssl-server 5 cipher rsa-with-rc4-128-sha 198.22.124.134 20094 weight 10`
- `ssl-server 5 cipher rsa-with-rc4-128-md5 198.22.124.134 20094 weight 8`
- `ssl-server 5 cipher rsa-with-des-cbc-sha 198.22.124.134 20094 weight 8`
- `ssl-server 5 cipher rsa-with-3des-edc-cbc-sha 198.22.124.134 20094 weight 8`
- `ssl-server 5 cipher rsa-export1024-with-rc4-56-sha 198.22.124.134 20094 weight 1`

## Solution 2

If you decide to support the other export cipher (rsa-export1024-with-des-cbc-sha), your weights look like this:

- ssl-server 5 cipher rsa-with-des-cbc-sha 198.22.124.134 20094 weight 10
- ssl-server 5 cipher rsa-with-rc4-128-sha 198.22.124.134 20094 weight 8
- ssl-server 5 cipher rsa-with-rc4-128-md5 198.22.124.134 20094 weight 8
- ssl-server 5 cipher rsa-with-3des-ede-cbc-sha 198.22.124.134 20094 weight 8
- ssl-server 5 cipher rsa-export1024-with-des-cbc-sha 198.22.124.134 20094 weight 1

## Related Information

- [Configuring SSL Traffic through the CSS](#)
  - [Technical Support – Cisco Systems](#)
- 

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2013 – 2014 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Jan 31, 2006

Document ID: 45321

---