Estrutura de Chaves Públicas Brasileira

Manual de Condutas Técnicas 9 - Volume II

Procedimentos de Ensaios para Avaliação de Conformidade aos Requisitos Técnicos de Softwares Provedores de Serviços Criptográficos (CSP) no Âmbito da ICP-Brasil

versão 1.0

São Paulo, 22 de novembro de 2007

Sumário

CONTROLE DE VERSÃO....3
1 INTRODUÇÃO....4
1.1 Organização deste documento....4
2 PARTE 1....6
2.1 Introdução....7
2.2 Requisitos de documentação....7
2.3 Requisitos de segurança....10
2.3.1 Requisitos de segurança baseados no padrão FIPS....10
2.3.1.1 Especificação da biblioteca criptográfica....10
2.3.1.2 Interfaces da biblioteca criptográfica....11
2.3.1.3 Algoritmos criptográficos....12
2.3.1.4 Auto-testes....23
2.3.1.5 Garantia do projeto....28
2.4 Requisitos de interoperabilidade....30
2.4.1 Gerenciamento de chaves criptográficas....30
2.4.2 Exportação e importação....32
2.4.3 Assinatura e certificação digital....34
2.4.4 Requisitos gerais de interoperabilidade....35
2.4.4.1 Requisitos gerais de um CSP....35
2.4.4.2 Requisitos sobre Microsoft CSP (CryptoAPI)....37
2.4.4.3 Requisitos sobre PKCS#11....43
2.4.4.4 Requisitos sobre Java Cryptographic Extension (JCE)....45
2.4.4.5 Requisitos sobre OpenSSL....47
3 REFERÊNCIAS NORMATIVAS....50

Controle de versão

Versão revisada	Data de emissão	Alterações realizadas

1 Introdução

O objetivo deste documento é especificar os procedimentos de homologação que serão aplicados para verificar os requisitos de documentação, segurança e interoperabilidade para CSPs [1] [30][31][32].

Os procedimentos de homologação se referem ao conjunto de métodos que serão usados para avaliar se um CSP está ou não em conformidade com os requisitos técnicos definidos pelo “Manual de Condutas Técnicas 9 - Volume I” [2][3].

Ao final de cada requisito avaliado, é preciso descrever os resultados dos ensaios realizados e emitir um relatório, cuja conclusão deve indicar a aderência ao respectivo requisito.

1.1 Organização deste documento

Cada seção deste documento contém um conjunto de requisitos que representam citações diretas do próprio texto do Manual de Condutas Técnicas 9 - Volume I. Os requisitos estão organizados da seguinte forma:

- REQUISITO <número_do_requisito>.<número_de_seqüência_do_requisito>
  - “número_do_requisito”: corresponde ao número de área definido no Manual de Condutas Técnicas 9 – Volume I;
  - “número_de_seqüência_do_requisito”: corresponde a um identificador seqüencial dos requisitos.

Os procedimentos de homologação visam a orientar sobre como proceder nos ensaios para um CSP. Os procedimentos de ensaio estão classificados e agrupados por níveis de segurança de homologação da seguinte forma:

- NSH 1: Este nível não requer depósito e análise de código-fonte associado ao CSP em homologação;
- NSH 2: Este nível requer depósito e análise apenas do código-fonte de componentes específicos associados ao CSP em homologação. Por exemplo, código-fonte do algoritmo gerador de números aleatórios;

- NSH 3: Este nível requer depósito e análise de código-fonte completo associado ao CSP em homologação. Por exemplo, código-fonte de todo software do CSP.

Os procedimentos de ensaio (EN) que devem ser desempenhados pelo analista LEA estão organizados da seguinte forma:

- EN.<número_do_requisito>.<número_de_seqüência_do_requisito>.<número_de_seqüência_do_ensaio>
  - “número_do_requisito”;
  - “número_de_seqüência_do_requisito”;
  - “número_de_seqüência_do_ensaio”: corresponde a um identificador seqüencial dos procedimentos que devem ser desempenhados.

Os termos usados neste documento estão referenciados no MCT – Glossário Geral [4].

2 PARTE 1

Procedimentos de ensaio a serem observados no processo de homologação de CSP

2.1 Introdução

Esta parte apresenta os procedimentos de ensaio que devem ser verificados no processo de homologação de CSP´s.

Os procedimentos de ensaio descritos nesta parte englobam:

- Requisitos de documentação;
- requisitos de segurança;
- requisitos de interoperabilidade;

2.2 Requisitos de documentação

REQUISITO II.01: A documentação deve estar escrita nos idiomas português do Brasil ou inglês.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.II.01.01: Verificar se a documentação está escrita nos idiomas português do Brasil ou inglês.

REQUISITO II.02: A PI deve fornecer manual de instalação e configuração, especificando os processos de instalação e configuração do CSP. Além disso, o manual de instalação deve especificar os sistemas operacionais suportados pelo CSP.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.II.02.01: Analisar a documentação e verificar se o manual de instalação e configuração especifica corretamente o processo de instalação e configuração do CSP que está sendo homologado.

EN.II.02.02: Analisar a documentação e verificar se o manual de instalação especifica quais procedimentos de inicialização devem ser adotados previamente a sua ativação.

EN.II.02.03: Analisar a documentação e verificar se o manual de instalação especifica quais componentes de software serão necessários para a ativação e configuração do CSP, tais como, JVM e sistema operacional compatíveis.

EN.II.02.04: Analisar a documentação e verificar se o manual de instalação e configuração especifica os procedimentos que devem ser adotados para habilitação do CSP na arquitetura criptográfica.

REQUISITO II.03: A PI deve fornecer o manual do usuário, detalhando as ferramentas e recursos disponíveis aos operadores do CSP.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.II.03.01: Analisar a documentação e verificar se o manual do usuário detalha corretamente as ferramentas e recursos disponíveis do CSP que está sendo homologado.

EN.II.03.02: Analisar a documentação e verificar se o manual do usuário especifica quais interfaces de administração e/ou configuração estão disponíveis, como por exemplo, arquivos-texto de configuração.

EN.II.03.03: Analisar a documentação e verificar se o manual do usuário especifica a versão e configuração do CSP.

REQUISITO II.04: A PI deve fornecer o manual de desenvolvedor detalhando a(s) API(s) para desenvolvimento de aplicações utilizando o CSP [5].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.II.04.01: Analisar a documentação e verificar se o manual do desenvolvedor especifica corretamente as funções da API do CSP que está sendo homologado.

EN.II.04.02: Analisar a documentação e verificar se o manual do desenvolvedor especifica a arquitetura do sistema.

EN.II.04.03: Analisar a documentação e verificar se o manual do desenvolvedor especifica as SPIs implementadas da arquitetura criptográfica.

EN.II.04.04: Analisar a documentação e verificar se o manual do desenvolvedor especifica os tratamentos de erros das chamadas das funções da API.

REQUISITO II.05: A PI deve fornecer um manual de integração do CSP com as APIs de mercado para dispositivos de armazenamento como *smart cards* ou *tokens*.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.II.05.01: Analisar a documentação e verificar se o manual de integração especifica corretamente as integrações das funções da API do CSP que está sendo homologado com as APIs de mercado para dispositivos de armazenamento.

REQUISITO II.06: A PI deve fornecer trechos de código-fonte para utilização do CSP.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.II.06.01: Analisar a documentação e verificar se são fornecidos trechos de código-fonte do CSP que está sendo homologado.

2.3 Requisitos de segurança

2.3.1 Requisitos de segurança baseados no padrão FIPS

2.3.1.1 Especificação da biblioteca criptográfica

REQUISITO III.01: A documentação deve especificar cada subsistema empregado pelo CSP [5].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.III.01.01: Verificar se a documentação descreve cada subsistema empregado pelo CSP.

REQUISITO III.02: Caso o CSP carregue dinamicamente subsistemas na hora de execução do mesmo, deve existir um mecanismo de integridade do CSP, impedindo substituição de subsistemas por sistemas mal intencionados [5].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.III.02.01: Verificar se o CSP carrega dinamicamente subsistemas na hora de execução.

EN.III.02.02: Verificar se o CSP possui mecanismos de integridade, impedindo substituição de subsistemas por sistemas mal intencionados.

REQUISITO III.03: A documentação deve especificar o método para garantia de integridade do CSP [5].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.III.03.01: Verificar se a documentação especifica o método para garantia de integridade do CSP.

2.3.1.2 Interfaces da biblioteca criptográfica

REQUISITO III.04: A documentação técnica do CSP deve especificar claramente as seguintes interfaces [5]:

- Entrada de dados: Parâmetros de entrada para todas as funções que aceitam entrada do invocador da API;
- Saída de dados: Parâmetros de saída de funções que retornam dados como argumentos ou como valor de retorno da função;
- Saída de estado: Informação retornada por meio de exceções (códigos de retorno ou *exit*).

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.III.04.01: Verificar se a documentação descreve as interfaces presentes no CSP. Para cada interface, é preciso verificar sua classificação quanto a:

- Interface de entrada de dados;
- interface de saída de dados;
- interface de saída de estado.

EN.III.04.02: Analisar o CSP e suas interfaces de entrada de dados, verificando que os seguintes tipos de dados podem ser inseridos e processados:

- Dados em texto claro que deve ser cifrado ou assinado pelo CSP;
- texto cifrado ou assinado que deve ser decifrado ou verificado pelo CSP;
- chaves criptográficas em texto claro ou cifradas e outros dados de gerenciamento de chaves que são inseridos e utilizados pelo CSP, tais como, vetores e dados de iniciação, informação sobre particionamento de chaves, etc;

- dados de autenticação em texto claro ou cifrado que devem ser inseridos no CSP, tais como, senhas, PINs, e/ou informações biométricas;
- informações de estado de fontes externas (por exemplo, dispositivo criptográfico);
- quaisquer outras informações que são inseridas no CSP para processamento ou armazenamento.

EN.III.04.03: Analisar o CSP e suas interfaces de saída de dados, verificando que os seguintes tipos de dados podem ser emitidos:

- Dados em texto claro que foram decifrados pelo CSP;
- texto cifrado que foi criptografado pelo CSP;
- assinaturas digitais que foram geradas pelo CSP;
- chaves criptográficas em texto claro ou cifradas e outros dados de gerenciamento de chaves que foram gerados pelo CSP, tais como, vetores e dados de iniciação, informação sobre particionamento de chaves, etc;
- informações de controle emitidas pelo CSP para entidades externas (por exemplo, dispositivo criptográfico);
- quaisquer outras informações que são emitidas pelo CSP após processamento ou armazenamento, exceto informações de estado.

EN.III.04.04: Analisar o CSP e suas interfaces de saída de estado, verificando que todas informações de estado utilizadas para mostrar o estado do CSP podem ser emitidas, tais como:

- informação retornada por meio de exceções (códigos de retorno ou *exit*).
- quaisquer outras informações de saída de estado.

2.3.1.3 Algoritmos criptográficos

REQUISITO III.05: O CSP deve suportar no mínimo as seguintes funções criptográficas [6]:

- Criptografia de dados:

- DES (*Data Encryption Standard*) nos modos de operação ECB e CBC, apenas para uso legado (conforme padrão NIST FIPS PUB 46-3) [9];
- *Triple*-DES (3DES ou TDES) nos modos de operação ECB e CBC (conforme padrão NIST FIPS PUB 46-3) [9];
- AES (*Advanced Encryption Standard*) com tamanho de chave 128 bits nos modos de operação ECB e CBC (conforme padrão NIST FIPS PUB 197) [17];
- RSA com utilização de chaves de comprimento maior do que 1024 bits, conforme padrões ANSI X9.31 [18] e PKCS#1 v. 1.5 [11].

- Autenticação e assinatura digital de dados:
  - RSA com tamanho mínimo de chaves de 1024 bits, conforme padrões NIST FIPS PUB 186, FIPS PUB 196 e PKCS#1 v. 1.5 [11].
- Resumo criptográfico de dados (*Hash*) [15] :
  - SHA-1 (*Secure Hash Algorithm*), apenas para uso legado, conforme padrão NIST FIPS PUB 180-2;
  - SHA-256 (*Secure Hash Algorithm*) conforme padrão NIST FIPS PUB 180-2.

Procedimentos de ensaio para NSH 1, 2 e 3 [7]:

EN.III.05.01: Verificar se a documentação descreve os sistemas criptográficos suportados pelo CSP.

EN.III.05.02: Executar testes de criptografia de dados verificando o suporte pelo CSP dos algoritmos DES e 3DES. Este ensaio deve estar baseado nos testes de validação publicados pelo NIST, conforme documentos listados na seção de referências normativas [8]. Os documentos de testes de validação estão organizados para realizar testes automáticos em componentes denominados "*Implementation Under Test* (IUT)". Os testes de criptografia de dados consistem em:

- Testes de respostas conhecidas (*Known Answer Tests*), que avaliam os componentes do DES e 3DES por meio dos parâmetros de entrada e de

saída conhecidos. Estes testes também verificam a confiabilidade com relação aos erros de implementação nos componentes dos algoritmos;

- testes de modo de operação (*Modes of Operation Tests*), que avaliam a possibilidade de haver erros feitos deliberadamente ou de forma maliciosa nas implementações dos algoritmos;
- testes de mensagens de múltiplos blocos (*Multi-block message tests*), que avaliam os algoritmos no que se refere às mensagens com múltiplos blocos. Para o modo de operação CBC, verificar a exatidão das operações de cifragem/decifragem destes algoritmos.

EN.III.05.03: Executar testes de criptografia de dados verificando o suporte pelo CSP do algoritmo AES. Este ensaio deve estar baseado nos testes de validação publicados pelo NIST, conforme documentos listados na seção de referências normativas [16]. Os documentos de testes de validação estão organizados para realizar testes automáticos em componentes denominados "*Implementation Under Test* (IUT)". Os testes de criptografia de dados consistem em:

- Testes de respostas conhecidas (*Known Answer Tests*), que avaliam os componentes do AES por meio dos parâmetros de entrada e de saída conhecidos. Estes testes também verificam a confiabilidade com relação aos erros de implementação nos componentes dos algoritmos;
- testes de modo de operação (*Modes of Operation Tests*), que avaliam a possibilidade de haver erros feitos deliberadamente ou de forma maliciosa nas implementações dos algoritmos;
- testes de mensagens de múltiplos blocos (*Multi-block message tests*), que avaliam os algoritmos no que se refere às mensagens com múltiplos blocos.

EN.III.05.04: Executar testes de criptografia de chave pública verificando o suporte pelo CSP do algoritmo RSA. Este ensaio deve estar baseado nos testes de validação publicados pelo NIST, conforme documentos listados na seção de referências normativas [10]. O documento de testes de validação está organizado para realizar testes automáticos em componentes denominados "*Implementation Under Test* (IUT)". Os testes de criptografia de chave pública consistem em:

- Teste de geração de chaves, que avalia a habilidade de uma IUT de gerar os valores corretos dos componentes do algoritmo;
- teste de geração de assinaturas, que avalia a habilidade de uma IUT em gerar a assinatura correta que pode ser validada pela chave pública associada;
- teste de verificação de assinaturas, que avalia a habilidade da IUT em reconhecer assinaturas válidas e inválidas.

EN.III.05.05: Executar testes de resumo criptográfico de dados verificando o suporte pelo CSP dos algoritmos SHA-1 e SHA-256. Este ensaio deve estar baseado nos testes de validação publicados pelo NIST, conforme documentos listados na seção de referências normativas [14]. O documento de testes de validação está organizado para realizar testes automáticos em componentes denominados "*Implementation Under Test* (IUT)". Os testes de resumo criptográfico de dados consistem em:

- Testes de mensagens curtas (*Short Message Test*), que avaliam a exatidão na geração do resumo criptográfico de dados com relação ao tamanho da mensagem de entrada;
- testes de mensagens longas selecionadas (*Selected Long Message Test*), que avaliam a exatidão na geração do resumo criptográfico para mensagens que contêm múltiplos blocos;
- testes de mensagens geradas pseudo aleatoriamente (*Pseudorandomly generated messages test*), que verificam a exatidão dos resumos criptográficos de dados para mensagens geradas pseudo aleatoriamente.

EN.III.05.06: Executar testes de performance para a criptografia de dados verificando os resultados obtidos pelo CSP dos algoritmos do REQUISITO III.05. Os testes de performance para criptografia de dados consistem em obter a velocidade em que os algoritmos são executados e verificar se esses valores são compatíveis para os algoritmos em questão.

No quadro abaixo temos alguns exemplos de valores obtidos a partir de códigos em C++, compilado com Microsoft Visual C++ 2005 SP1 e rodando em um Intel Core 2 1.83 GHz processor com Windows XP SP 2 em modo 32-bit.

Algoritmo	MiB/Seg	Ciclos / Byte
HMAC(SHA-1)	152	11.5
CBC-MAC/AES	86	20.2
DMAC/AES	82	21.3
SHA-1	155	11.3
SHA-256	81	21.5
AES/ECB (chave de 128-bit)	99	17.7
DES	34	51.1

Operação	Miliseg/Operação	Megaciclos/Operação
RSA 1024 Cifração	0.07	0.13
RSA 1024 Decifração	1.52	2.78
RSA 2048 Cifração	0.15	0.28
RSA 2048 Decifração	5.95	10.89
RSA 1024 Assinatura	1.42	2.60
RSA 1024 Verificação	0.07	0.13
RSA 2048 Assinatura	5.95	10.89
RSA 2048 Verificação	0.15	0.28

RECOMENDAÇÃO III.01: O CSP também pode suportar a função AES (*Advanced Encryption Standard*) com utilização de chaves de comprimento de 192 e 256 bits, conforme padrão NIST FIPS PUB 197, para cifração e decifração de dados [17].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.REC.III.01.01: Verificar se a documentação descreve os sistemas criptográficos suportados pelo CSP.

EN.REC.III.01.02: Executar testes de cifração e decifração de dados verificando o suporte pelo CSP do algoritmo AES. Este ensaio deve estar baseado nos testes de validação publicados pelo NIST, conforme documentos listados na seção de referências normativas [16]. Os documentos de testes de validação estão organizados para realizar testes automáticos em componentes denominados "*Implementation Under Test* (IUT)". Os testes de cifração e decifração de dados consistem em:

- Testes de respostas conhecidas (*Known Answer Tests*), que avaliam os componentes do AES por meio dos parâmetros de entrada e de saída conhecidos. Estes testes também verificam a confiabilidade com relação aos erros de implementação nos componentes dos algoritmos;
- testes de modo de operação (*Modes of Operation Tests*), que avaliam a possibilidade de haver erros feitos deliberadamente ou de forma maliciosa nas implementações dos algoritmos;
- testes de mensagens de múltiplos blocos (*Multi-block message tests*), que avaliam os algoritmos no que se refere às mensagens com múltiplos blocos.

EN.REC.III.01.03: Executar testes de performance para a criptografia de dados verificando os resultados obtidos pelo CSP dos algoritmos AES (para 192 e 256 bits). Os testes de performance para criptografia de dados consistem em obter a velocidade em que os algoritmos são executados e verificar se esses valores são compatíveis para os algoritmos em questão.
No quadro abaixo temos alguns exemplos de valores obtidos a partir de códigos em C++, compilado com Microsoft Visual C++ 2005 SP1 e rodando em um Intel Core 2 1.83 GHz processor com Windows XP SP 2 em modo 32-bit.

Algoritmo	**MiB/Seg**	**Ciclos / Byte**
AES/ECB (chave de 192-bit)	86	20.2
AES/ECB (chave de 256-bit)	77	22.6

RECOMENDAÇÃO III.02: O CSP também pode suportar a função DSA (*Data Signature Algorithm*) com utilização de chaves de comprimento maior do que 512 bits, conforme padrão NIST FIPS PUB 186, para autenticação e assinatura digital de dados [12].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.REC.III.02.01: Verificar se a documentação descreve os sistemas criptográficos suportados pelo CSP.

EN.REC.III.02.02: Executar testes de autenticação e assinatura digital de dados verificando o suporte pelo CSP do algoritmo DSA. Este ensaio deve estar baseado nos testes de validação publicados pelo NIST, conforme documentos listados na seção de referências normativas [7]. Os documentos de testes de validação estão organizados para realizar testes automáticos em componentes denominados "*Implementation Under Test* (IUT)". Os testes de autenticação e assinatura digital de dados consistem em:

- Testes de respostas conhecidas (*Known Answer Tests*), que avaliam os componentes do DSA por meio dos parâmetros de entrada e de saída conhecidos. Estes testes também verificam a confiabilidade com relação aos erros de implementação nos componentes dos algoritmos;
- testes de modo de operação (*Modes of Operation Tests*), que avaliam a possibilidade de haver erros feitos deliberadamente ou de forma maliciosa nas implementações dos algoritmos;
- testes de mensagens de múltiplos blocos (*Multi-block message tests*), que avaliam os algoritmos no que se refere às mensagens com múltiplos blocos.

EN.REC.III.02.03: Executar testes de performance para a criptografia de dados verificando os resultados obtidos pelo CSP dos algoritmos DSA. Os testes de performance para criptografia de dados consistem em obter a velocidade em que os algoritmos são executados e verificar se esses valores são compatíveis para os algoritmos em questão.

No quadro abaixo temos alguns exemplos de valores obtidos a partir de códigos em C++, compilado com Microsoft Visual C++ 2005 SP1 e rodando em um Intel Core 2 1.83 GHz processor com Windows XP SP 2 em modo 32-bit.

Operação	Miliseg/Operação	Megaciclos/Operação
DSA 1024 Assinatura	0.47	0.85
DSA 1024 Verificação	0.52	0.95

RECOMENDAÇÃO III.03: O CSP também pode suportar as seguintes funções para a geração de resumos criptográficos de dados, conforme padrão NIST FIPS PUB 180-2 [15]:

- SHA-224;
- SHA-256
- SHA-384;
- SHA-512.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.REC.III.03.01: Verificar se a documentação descreve algoritmos opcionais para geração de resumos criptográficos de dados.

EN.REC.III.03.02: Repetir o ensaio **EN.III.05.05** para os algoritmos SHA-224, SHA-256, SHA-384 e SHA-512 [14].

EN.REC.III.03.03: Executar testes de performance para a criptografia de dados verificando os resultados obtidos pelo CSP dos algoritmos SHA-224, SHA-256, SHA-384 e SHA-512. Os testes de performance para criptografia de dados consistem em obter a velocidade em que os algoritmos são executados e verificar se esses valores são compatíveis para os algoritmos em questão.
No quadro abaixo temos alguns exemplos de valores obtidos a partir de códigos em C++, compilado com Microsoft Visual C++ 2005 SP1 e rodando em um Intel Core 2 1.83 GHz processor com Windows XP SP 2 em modo 32-bit.

Algoritmo	MiB/Seg	Ciclos / Byte
SHA-256	81	21.5
SHA-512	99	17.6

RECOMENDAÇÃO III.04: O CSP também pode suportar as seguintes funções para a autenticação e integridade de dados:

- CBC-MAC baseado nos algoritmos 3DES ou AES, conforme padrão NIST PUB 800-38B [20];
- HMAC baseado nos algoritmos de resumos criptográficos implementados, conforme padrão NIST FIPS PUB 198 [19];
- CMAC baseado nos algoritmos 3DES ou AES, conforme padrão NIST PUB 800-38B [20];
- MAC-CCM baseado nos algoritmos 3DES ou AES, conforme padrão NIST PUB 800-38C [21].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.REC.III.04.01: Verificar se a documentação descreve os sistemas criptográficos suportados pelo CSP.

EN.REC.III.04.02: Executar testes de autenticação e integridade de dados verificando o suporte pelo CSP dos algoritmos CBC-MAC, HMAC, CMAC e MAC. Este ensaio deve estar baseado nos testes de validação publicados pelo NIST, conforme documentos listados na seção de referências normativas [7]. Os documentos de testes de validação estão organizados para realizar testes automáticos em componentes denominados "*Implementation Under Test* (IUT)". Os testes de autenticação e integridade de dados consistem em:

- Testes de respostas conhecidas (*Known Answer Tests*), que avaliam os componentes de CBC-MAC, HMAC, CMAC e MAC por meio dos parâmetros de entrada e de saída conhecidos. Estes testes também verificam a confiabilidade com relação aos erros de implementação nos componentes dos algoritmos;
- testes de modo de operação (*Modes of Operation Tests*), que avaliam a possibilidade de haver erros feitos deliberadamente ou de forma maliciosa nas implementações dos algoritmos;

- testes de mensagens de múltiplos blocos (*Multi-block message tests*), que avaliam os algoritmos no que se refere às mensagens com múltiplos blocos.

EN.REC.III.04.03: Executar testes de performance para a criptografia de dados verificando os resultados obtidos pelo CSP dos algoritmos CBC-MAC, HMAC, CMAC e MAC. Os testes de performance para criptografia de dados consistem em obter a velocidade em que os algoritmos são executados e verificar se esses valores são compatíveis para os algoritmos em questão.
No quadro abaixo temos alguns exemplos de valores obtidos a partir de códigos em C++, compilado com Microsoft Visual C++ 2005 SP1 e rodando em um Intel Core 2 1.83 GHz processor com Windows XP SP 2 em modo 32-bit.

Algoritmo	MiB/Seg	Ciclos / Byte
HMAC(SHA-1)	152	11.5
CBC-MAC/AES	86	20.2
CMAC/AES	82	21.3

RECOMENDAÇÃO III.05: Para a biblioteca criptográfica ICP-Brasil que suportar funções para derivação de chaves simétricas baseada em senha, é recomendável a seguinte função de derivação de chaves:

- Função 2 de derivação de chaves baseada em senha, PBKDF2, como especificada em PKCS#5 [22].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.REC.III.05.01: Verificar se a documentação descreve os sistemas criptográficos suportados pelo CSP.

EN.REC.III.05.02: Executar testes de derivação de chaves simétricas baseada em senha verificando o suporte pelo CSP do algoritmo PBKDF2. Este ensaio deve estar baseado nos testes de validação publicados pelo NIST, conforme documentos listados na seção de referências normativas [7]. Os documentos de testes de validação estão organizados para realizar testes automáticos em componentes

denominados "*Implementation Under Test* (IUT)". Os testes de derivação de chaves simétricas baseada em senha consistem em:

- Testes de respostas conhecidas (*Known Answer Tests*), que avaliam os componentes do PBKDF2 por meio dos parâmetros de entrada e de saída conhecidos. Estes testes também verificam a confiabilidade com relação aos erros de implementação nos componentes dos algoritmos;
- testes de modo de operação (*Modes of Operation Tests*), que avaliam a possibilidade de haver erros feitos deliberadamente ou de forma maliciosa nas implementações dos algoritmos;
- testes de mensagens de múltiplos blocos (*Multi-block message tests*), que avaliam os algoritmos no que se refere às mensagens com múltiplos blocos.

REQUISITO III.06: O CSP deve suportar o formato PKCS#1 para armazenamento das chaves assimétricas [11] [24].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.III.06.01: Verificar se a documentação descreve funções que demonstrem o suporte ao formato PKCS#1.

EN.III.06.02: Executar testes de chamada dessas funções e verificar se foram geradas informações no formato PKCS#1, conforme definição no requisito III.1.6 do Documento Manual de Condutas Técnicas 9 – Volume I.

RECOMENDAÇÃO III.06: O CSP pode suportar o formato PKCS#12 [23] para troca de identificações pessoais. O formato de troca de informações pessoais (PFX), permite a transferência de certificados e das chaves particulares correspondentes entre computadores ou de um computador para mídia removível. Isso pode ser feito entre produtos do mesmo fornecedor ou de diferentes fornecedores.
Para tal, o CSP precisa importar como arquivo para dentro do módulo, decifrar a chave privada e depois criar objetos de PKCS#11 como chave pública e chave

privada baseado nisso. O certificado também precisa ser importado e ligado ao par de chaves.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.REC.III.06.01: Verificar se a documentação descreve funções que demonstrem o suporte ao formato PKCS#12.

EN.REC.III.06.02: Executar testes de chamada dessas funções e verificar se foram geradas informações no formato PKCS#12, conforme definição na recomendação III.1.6 do Documento Manual de Condutas Técnicas 9 – Volume I.

RECOMENDAÇÃO III.07: O CSP pode suportar o formato PKCS#8 [25] para armazenamento das chaves assimétricas.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.REC.III.07.01: Verificar se a documentação descreve funções que demonstrem o suporte ao formato PKCS#8.

EN.REC.III.07.02: Executar testes de chamada destas funções e verificar se foram geradas informações no formato PKCS#8, conforme definição na recomendação III.1.7 do Documento Manual de Condutas Técnicas 9 – Volume I.

2.3.1.4 Auto-testes

REQUISITO III.07: O CSP deve executar um número de auto-testes para garantir a operação correta do mesmo [5].
Podemos citar as seguintes classes de auto-testes para um CSP:

- Testes de algoritmos criptográficos;
- testes da integridade de software;
- testes de carregamento de software;

- testes de funções críticas;
- testes de respostas conhecidas.

Procedimentos de ensaio para NSH 1:

EN.III.07.01: Executar os auto-testes criptográficos de algoritmos criptográficos, de integridade de software, de carregamento de software, de funções críticas e de respostas conhecidas.

EN.III.07.02: Provocar erro nas funcionalidades do CSP através dos auto-testes e verificar se o CSP trata o erro devidamente.

EN.III.07.03: Para cada estado de erro dos auto-testes que não pôde ser alcançado, avaliar o código-fonte do CSP e a documentação do projeto para determinar se há algum tipo de controle que impeça qualquer operação criptográfica de ser realizada enquanto o estado de erro persistir.

EN.III.07.04: Verificar se a documentação do CSP especifica a utilização de ofuscação de código binário.

EN.III.07.05: Verificar, por meio de auto-testes, se o código binário do CSP está utilizando ofuscação e estimar a qualidade disso.

EN.III.07.06: Verificar, por meio de auto-testes, a possibilidade de achar chaves em memória durante operações criptográficas.

REQUISITO III.08: Os testes de integridade devem utilizar um tipo de algoritmo criptográfico como HMAC-SHA-1 calculado em cima do código compilado de cada componente do CSP.

Procedimentos de ensaio para NSH 2:

EN.III.08.01: Executar auto-testes criptográficos utilizando um tipo de algoritmo criptográfico como HMAC-SHA-1 calculado em cima do código compilado de cada componente do CSP.

EN.III.08.02: Executar os auto-testes e verificar se a integridade de cada componente do CSP é mantida.

REQUISITO III.09: Os auto-testes devem ser chamados na instanciação do CSP. Adicionalmente, deve ser possível chamar por meio de função API como por exemplo *Executar_auto_testes( );*

Procedimentos de ensaio para NSH 3:

EN.III.09.01: Realizar auto-testes chamando a instanciação do CSP.

EN.III.09.02: Verificar se os auto-testes são chamados através de uma função da API do CSP.

REQUISITO III.10: Se o CSP apresentar falhas durante um auto-teste, o mesmo deve ser conduzido a um estado de erro e emitir um indicador de erro via “Interface de Saída de Estado”.

Procedimentos de ensaio para NSH 1:

EN.III.10.01: Verificar a documentação que descreve os estados de erro dos auto-testes suportados pelo CSP, bem como o indicador de erro associado em cada estado de erro.

EN.III.10.02: Provocar os estados de erro dos auto-testes por meio de software específico e analisar os indicadores de erro emitidos via “Interface de Saída de Estado”. Após a obtenção dos indicadores de erro dos auto-testes, verificar se os estados e indicadores de erro estão consistentes com a documentação.

Procedimentos de ensaio para NSH 2 e 3:

EN.III.10.03: Para cada estado de erro dos auto-testes que não puder ser alcançado, avaliar o código-fonte do CSP e a documentação do projeto para determinar o respectivo indicador de erro que seria emitido via “Interface de Saída de Estado”.

REQUISITO III.11: Nenhuma funcionalidade criptográfica deve estar disponível até a execução com sucesso dos auto-testes.

Procedimentos de ensaio para NSH 1:

EN.III.11.01: Analisar a documentação, sendo que as seguintes funções criptográficas devem estar incluídas na lista de funções inibidas quando o CSP estiver num estado de erro:

- Cifragem;
- decifração;
- geração segura de resumos criptográficos (*secure message hashing*);
- verificação e criação de assinaturas digitais;
- outras operações que necessitam do uso de criptografia.

EN.III.11.02: Provocar os estados de erro dos auto-testes suportados pelo CSP e para cada estado de erro alcançado em um auto-teste, efetuar tentativas de realização de operações criptográficas específicas. Para cada tentativa realizada, verificar se as operações criptográficas não devem ser concluídas de forma bem sucedida.

EN.III.11.03: Verificar se quando o CSP é conduzido a um estado de erro, não há qualquer saída de dados pela “Interface de Saída de Dados”.

Procedimentos de ensaio para NSH 2 e 3:

EN.III.11.04: Para cada estado de erro dos auto-testes que não puder ser alcançado, avaliar o código-fonte do CSP e a documentação do projeto para determinar se há algum tipo de controle que impeça qualquer operação criptográfica de ser realizada enquanto o estado de erro persistir.

REQUISITO III.12: Quando um estado de erro ocorrer devido a falhas em um auto-teste, toda saída ou envio de dados via "Interface de Saída de Dados" deve ser impedido.

Procedimentos de ensaio para NSH 1:

EN.III.12.01: Analisar a documentação e verificar se o CSP impede a saída ou envio de dados via "Interface de Saída de Dados" enquanto houver um estado de erro devido a falhas num auto-teste.

EN.III.12.02: Provocar uma falha em cada auto-teste suportado pelo CSP e para cada estado de erro alcançado em um auto-teste, efetuar testes em um sistema específico na "Interface de Saída de Dados". Durante a observação dos testes verificar se há qualquer evidência de tráfego pela "Interface de Saída de Dados".

Procedimentos de ensaio para NSH 2 e 3:

EN.III.12.03: Para cada estado de erro dos auto-testes que não puder ser alcançado, avaliar o código-fonte do CSP e a documentação do projeto, com o intuito de determinar se há algum tipo de controle que impeça que qualquer sinal trafegue pela "Interface de Saída de Dados".

REQUISITO III.13: A documentação do CSP deve especificar os seguintes itens:

- Os auto-testes realizados pelo CSP;
- os estados de erro que o CSP pode entrar quando um auto-teste falha;
- as condições e ações necessárias para sair dos estados de erro e reiniciar a operação normal do CSP.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.III.13.01: Verificar se a documentação atende ao **REQUISITO III.13**.

2.3.1.5 Garantia do projeto

REQUISITO III.14: A parte interessada deve fornecer documentação de utilização de ferramenta de controle de versão do código-fonte do CSP [5].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.III.14.01: Verificar se a documentação inclui ferramenta de controle de versão do código-fonte do CSP.

REQUISITO III.15: A documentação do CSP deve incluir diagramas de engenharia de software que representem a arquitetura do elemento de software [5].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.III.15.01: Verificar se a documentação inclui diagramas de engenharia de software que representem a arquitetura do elemento de software.

REQUISITO III.16: A documentação do CSP deve incluir diagramas que ilustrem sua relação de uso por outros elementos de software ou hardware [5].

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.III.16.01: Verificar se a documentação inclui diagramas que ilustrem sua relação de uso por outros elementos de software ou hardware.

REQUISITO III.17: Em caso do CSP ser *multi-threaded*, as funções que envolvem operações criptográficas devem ser *thread-safe*, ou seja, que não coloquem em risco nenhum tipo de informação protegida compartilhada contra divulgação, modificação e substituição não autorizada [5].

Procedimentos de ensaio para NSH 1:

EN.III.17.01: Verificar se a documentação do CSP descreve como sendo *multi-threaded.*

Procedimentos de ensaio para NSH 2 e 3:

EN.III.17.02: Verificar se o CSP apresenta operações criptográficas que sejam *thread-safe.*

EN.III.17.03: Analisar as operações criptográficas que sejam *thread-safe* do ensaio EN.III.17.02 por meio de aplicação específica e verificar se tais resultados das análises estão de acordo com os padrões especificados no **REQUISITO III.17.**

RECOMENDAÇÃO III.08: Os parâmetros de saída das funções das APIs do CSP podem apresentar as seguintes características [5]:

- Nenhum destes parâmetros deve ser utilizado como variável temporária durante a sua execução, isto é, não devemos atribuir os valores dos parâmetros de uma função a uma variável temporária durante a execução desta função;
- todos os parâmetros de saída devem somente retornar os tipos que foram pré-determinados na API, isto é, não devemos modificar os parâmetros de saída de uma função na execução da mesma.

Procedimentos de ensaio para NSH 3:

EN.REC.III.08.01: Verificar nos parâmetros das funções do CSP se os mesmos estão sendo utilizados como variável temporária.

EN.REC.III.08.02: Verificar nos parâmetros das funções do CSP se os parâmetros de saída estão retornando os tipos que foram pré-determinados na API.

EN.REC.III.08.03: Analisar nos parâmetros das funções do CSP dos ensaios EN.REC.III.08.01 e EN.REC.III.08.02 por meio de aplicação específica e verificar se tais resultados das análises estão de acordo com os padrões especificados na RECOMENDAÇÃO III.08**.**

2.4 Requisitos de interoperabilidade

REQUISITO IV.01: [item 2.1. do DOC-ICP-10.03] O CSP deve atender aos requisitos funcionais estabelecidos, conforme descrito nos itens a seguir. No escopo deste documento, pelo menos uma das seguintes APIs serão consideradas para análise dos requisitos funcionais:

- Microsoft *CryptoAPI [26]*;
- PKCS#11 V.2.11 [27];
- JCE [28];
- OpenSSL Engine [29].

Nota: Este requisito não é testado separadamente e faz parte da **seção 2.4.4**.

2.4.1 Gerenciamento de chaves criptográficas

REQUISITO IV.02: [referente ao item 2.1. do DOC-ICP-10.03] Os seguintes requisitos funcionais de gerenciamento de chaves criptográficas do CSP devem estar disponíveis por invocação via API:

- Gerar chave criptográfica assimétrica de forma randômica;

- destruir chave criptográfica assimétrica com sobrescrita de valores;
- recuperar parâmetros sobre uma determinada chave criptográfica assimétrica, tais como:
  - Algoritmo;
  - expoente público (RSA);
  - módulo (RSA);
  - tamanho da chave;
  - permissões.

Procedimentos de ensaio para NSH 1:

EN.IV.02.01: Analisar se a documentação descreve os requisitos funcionais de gerenciamento de chaves criptográficas.

Nota: Os ensaios referentes a esta seção serão realizados por meio de uma ferramenta de software ou utilitário desenvolvido pelo LEA, com base nas APIs e nas plataformas de sistemas operacionais.

EN.IV.02.02: Gerar chaves criptográficas assimétricas de forma aleatória no CSP. Após a geração, verificar se a chave gerada está presente e executar operações criptográficas que validem as chaves assimétricas.

EN.IV.02.03: Escolher uma determinada chave criptográfica assimétrica e depois recuperar seus parâmetros associados. Após a recuperação, verificar se os parâmetros obtidos correspondem à chave selecionada.

Procedimentos de ensaio para NSH 2 e 3:

EN.IV.02.04: Verificar, por meio de inspeção direta do código-fonte do CSP, se a destruição de chaves criptográficas assimétricas é realizada por meio da técnica de sobrescrita de valores.

2.4.2 Exportação e importação

REQUISITO IV.03: [referente ao item 2.1. do DOC-ICP-10.03] Os seguintes requisitos funcionais de exportação e importação devem estar disponíveis no CSP por invocação via API:

- Exportar chave criptográfica simétrica;
- importar chave criptográfica simétrica;
- exportar chave criptográfica assimétrica pública. A exportação de chave criptográfica assimétrica privada só deve ser possível para certificados dos tipos A1, A2, S1 e S2;
- importar chave criptográfica assimétrica pública ou privada;
- exportar certificado segundo padrão X.509 versão 3;
- importar certificado segundo padrão X.509 versão 3.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.03.01: Analisar se a documentação descreve os requisitos de exportação e importação.

Nota: Os ensaios referentes a esta seção serão realizados por meio de uma ferramenta de software ou utilitário desenvolvido pelo LEA, com base nas APIs e nas plataformas de sistemas operacionais.

EN.IV.03.02: Exportar a chave criptográfica simétrica. Após a exportação, verificar se a chave foi exportada e executar operações criptográficas que validem a chave.

EN.IV.03.03: Importar a chave criptográfica simétrica. Após a importação, verificar se a chave foi importada e executar operações criptográficas que validem a chave.

EN.IV.03.04: Exportar a chave criptográfica assimétrica pública. Após a exportação, verificar se a chave foi exportada e executar operações criptográficas que validem a chave.

EN.IV.03.05: Exportar a chave criptográfica assimétrica privada (para certificados dos tipos A1, A2, S1 e S2). Após a exportação, verificar se a chave foi exportada e executar operações criptográficas que validem a chave.

EN.IV.03.06: Verificar e garantir que não sejam exportadas chaves criptográficas assimétricas privadas para certificados dos tipos A3, A4, S3 e S4.

EN.IV.03.07: Importar a chave criptográfica assimétrica pública. Após a importação, verificar se a chave foi importada e executar operações criptográficas que validem a chave.

EN.IV.03.08: Importar a chave criptográfica assimétrica privada. Após a importação, verificar se a chave foi importada e executar operações criptográficas que validem a chave.

EN.IV.03.09: Exportar certificado digital segundo formato CMS. Após a exportação, verificar se o certificado foi exportado no formato CMS e se tal certificado corresponde àquele selecionado durante a operação de exportação.

EN.IV.03.10: Exportar cadeia de certificação. Após a exportação, verificar se a cadeia de certificação exportada é válida e corresponde àquela selecionada durante a operação de exportação.

EN.IV.03.11: Importar certificado digital segundo padrões X.509 versão 3. Após a importação, verificar se o certificado foi importado nos padrões requisitados e se tal certificado corresponde àquele selecionado durante a operação de importação.

EN.IV.03.12: Importar cadeia de certificação. Após a importação, verificar se a cadeia de certificação é válida e corresponde àquela selecionada durante a operação de importação.

RECOMENDAÇÃO IV.01: A exportação de chaves criptográficas assimétricas públicas pode ser feita automaticamente por software, como uma facilidade oferecida pelo fabricante ao usuário.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.REC.IV.01.01: Analisar se a documentação descreve a exportação de chaves criptográficas assimétricas públicas feita automaticamente por software.

EN.REC.IV.01.02: Verificar se a exportação de chaves criptográficas assimétricas públicas é feita automaticamente por software.

2.4.3 Assinatura e certificação digital

REQUISITO IV.04: [referente ao item 2.1. do DOC-ICP-10.03] Os seguintes requisitos funcionais de assinatura e certificação digital do CSP devem estar disponíveis por invocação via API:

- Realizar a assinatura digital de uma mensagem conforme o padrão PKCS #1 V.1.5;
- realizar a verificação de uma assinatura digital de uma mensagem conforme o padrão PKCS #1 V.1.5.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.04.01: Analisar se a documentação descreve os requisitos de assinatura e certificação digital.

Nota: Os ensaios referentes a esta seção serão realizados por meio de uma ferramenta de software ou utilitário desenvolvido pelo LEA, com base nas APIs e nas plataformas de sistemas operacionais.

EN.IV.04.02: Realizar assinatura digital de uma mensagem conforme o padrão PKCS #1 V.1.5.

EN.IV.04.03: Realizar verificação de assinatura digital de uma mensagem conforme o padrão PKCS #1 V.1.5.

2.4.4 Requisitos gerais de interoperabilidade

REQUISITO IV.05: No mínimo uma das seguintes APIs serão consideradas para análise dos requisitos de interoperabilidade:

- Microsoft CSP (*CryptoAPI)*;
- PKCS#11 v. 2.11;
- JCE/JCA;
- OpenSSL Engine.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.05.01: Analisar a documentação e verificar se no mínimo uma das APIs estão implementadas no CSP em homologação.

2.4.4.1 Requisitos gerais de um CSP

REQUISITO IV.06: Um CSP deve ser capaz de fazer, no mínimo, as seguintes operações:

- Gerar par de chaves especificando os componentes de chaves assimétricas em texto claro;
- cifrar e decifrar chaves especificando os componentes de chaves assimétricas em texto claro;
- importar e exportar chaves (PKCS#12) especificando os componentes de chaves assimétricas privadas criptografados. A exportação de chaves

assimétricas privadas é permitida apenas no caso de certificados do tipo A1, S1, A2 e S2;

- assinar conteúdo especificando os componentes de chaves assimétricas públicas em texto claro;
- verificar assinatura especificando os componentes de chaves assimétricas públicas em texto claro.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.06.01: Analisar se a documentação corresponde aos requisitos gerais do CSP.

Nota: Os ensaios referentes a esta seção serão realizados por meio de uma ferramenta de software ou utilitário desenvolvido pelo LEA, com base nas APIs e nas plataformas de sistemas operacionais.

EN.IV.06.02: Gerar par de chaves assimétricas especificando os componentes das chaves em texto claro. Após a geração, verificar se as chaves foram geradas, se os componentes das chaves estão em texto claro e executar operações criptográficas que validem as chaves.

EN.IV.06.03: Cifrar e decifrar chaves assimétricas especificando os componentes das chaves em texto claro. Após cifrar/decifrar as chaves, verificar se as operações foram geradas, se os componentes das chaves estão em texto claro e executar operações criptográficas que validem as chaves.

EN.IV.06.04: Importar e exportar chaves assimétricas especificando os componentes de chaves assimétricas privadas criptografados. Após importar/exportar as chaves, verificar se as operações foram realizadas, se os componentes da chave pública estão em texto claro, se os da chave privada estão criptografados e executar operações criptográficas que validem as chaves.

EN.IV.06.05: Assinar conteúdo especificando os componentes das chaves assimétricas públicas em texto claro. Após a assinatura, verificar se a mesma foi gerada corretamente, se os componentes das chaves assimétricas públicas estão em texto claro e executar operações criptográficas que validem a operação.

EN.IV.06.06: Verificar uma assinatura especificando os componentes de chaves assimétricas públicas em texto claro. Após a verificação, verificar se os componentes das chaves assimétricas públicas estão em texto claro e executar operações criptográficas que validem a operação.

REQUISITO IV.07: A implementação da interface nativa deve suportar os algoritmos criptográficos descritos na seção “Algoritmos criptográficos obrigatórios”.

EN.IV.07.01: Verificar se a documentação que acompanha o CSP especifica o suporte aos algoritmos criptográficos descritos na seção “Algoritmos criptográficos obrigatórios”.

2.4.4.2 Requisitos sobre Microsoft CSP (CryptoAPI)

REQUISITO IV.08: O CSP deve suportar, no mínimo, uma implementação do MS CSP (*CryptoAPI)*, versão 1.0.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.08.01: Verificar na documentação se o CSP suporta uma implementação do MS CSP (*CryptoAPI)*.

EN.IV.08.02: Verificar se a documentação que acompanha o CSP especifica a versão do MS CSP (*CryptoAPI)* suportado.

REQUISITO IV.09: O CSP deve ser implementado na forma de DLL.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.09.01: Verificar na documentação se o CSP foi implementado na forma de DLL.

RECOMENDAÇÃO IV.02: É recomendável que o CSP seja implementado em apenas um módulo DLL, para reduzir problemas relativos ao contexto de execução.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.REC.IV.02.01: Verificar na documentação se o CSP foi implementado em apenas um módulo de DLL.

REQUISITO IV.10: O CSP deve ser assinado pela Microsoft. Caso seja composto de mais de uma DLL, todas devem ser assinadas.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.10.01: Verificar se o CSP foi assinado pela Microsoft.

REQUISITO IV.11: O CSP deve possuir um software instalador, que copie os arquivos necessários a um diretório apontado no PATH e que crie as entradas no registro adequadas. Este software deve ser capaz de desinstalar os componentes do CSP, apagando as informações não compartilhadas por outros sistemas instalados.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.11.01: Verificar na documentação se o CSP possui um software instalador.

EN.IV.11.02: Executar o software instalador e verificar se os arquivos necessários para o funcionamento do CSP são copiados a um diretório apontado no PATH e se as entradas no registro são criadas adequadamente.

EN.IV.11.03: Executar o software instalador para desinstalar o CSP e verificar se os arquivos necessários para o funcionamento do CSP são apagados.

REQUISITO IV.12: [referente ao artigo do MSDN: "*The Smart Card CSP Cookbook*"] O CSP deve exportar, isto é, expor sua interface, das seguintes chamadas:

- *CPAcquireContext*
- *CPCreateHash*
- *CPDecrypt*
- *CPDeriveKey*
- *CPDestroyHash*
- *CPDestroyKey*
- *CPEncrypt*
- *CPExportKey*
- *CPGenKey*
- *CPGenRandom*
- *CPGetHashParam*
- *CPGetKeyParam*
- *CPGetProvParam*
- *CPGetUserKey*
- *CPHashData*
- *CPHashSessionKey*
- *CPImportKey*
- *CPReleaseContext*
- *CPSetHashParam*
- *CPSetKeyParam*
- *CPSetProvParam*
- *CPSignHash*
- *CPVerifySignature*

Sendo obrigatória a implementação das seguintes funções:

- *CPAcquireContext* para criação e remoção de *key containers* existentes*;*
- *CPGenKey* tanto para chaves simétricas quanto para assimétricas;
- *CPImportKey* especificando tanto as chaves simétricas quanto as assimétricas;
- *CPGetKeyParam* para recuperação de parâmetros de permissões de acesso às chaves criadas/existentes em um *key container*;
- *CPHashData e CPSignHash* para geração de assinatura utilizando chave assimétrica;
- *CPVerifySignature* para verificação da assinatura após a importação da chave pública via CP*ImportKey;*
- *CPDecrypt e CPEncrypt* para decifrar e cifrar chaves simétricas e assimétricas.

As funções não implementadas devem retornar o código de erro E_NOTIMPL.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.12.01: Verificar na documentação se o CSP exporta as chamadas citadas no requisito.

EN.IV.12.02: Verificar na documentação se o CSP implementa as chamadas tidas como obrigatórias.

Procedimentos de ensaio para NSH 2 e 3:

EN.IV.12.03: Verificar na implementação das funções do CSP, se as funções do MS *CryptoAPI* não implementadas retornam o código de erro E_NOTIMPL.

- **Gerenciamento do cache de PIN:**

REQUISITO IV.13: Caso o CSP utilize cartão inteligente, deve ser configurável o armazenamento do PIN em cache após ter sua validade verificada pelo cartão. Mecanismo de relacionamento entre o usuário corrente e o PIN deve ser implementado.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.13.01: Verificar na documentação se o CSP utiliza cartão inteligente. Se usar, verificar na documentação se o armazenamento do PIN em cache é configurável.

EN.IV.13.02: Verificar na documentação se um mecanismo de relacionamento entre o usuário corrente e o PIN foi implementado.

REQUISITO IV.14: Caso o CSP utilize cartão inteligente, o cache do PIN deve ser apagado quando o mesmo for removido da leitora ou quando encerrar a sessão do Windows.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.14.01: Verificar na documentação se o CSP utiliza cartão inteligente. Se usar, verificar na documentação se é informado que o cache do PIN é apagado quando o mesmo for removido da leitora ou quando encerrar a sessão do Windows.

EN.IV.14.02: Verificar se a documentação descreve as situações nas quais o código PIN:

- É mantido em cache;
- deve ser eliminado do cache.

EN.II.14.03: Avaliando o cache do código PIN, é preciso realizar por meio de uma aplicação específica, uma seqüência de operações criptográficas que necessite do uso do PIN e verificar se, após a primeira operação criptográfica, o PIN não é mais solicitado.

EN.II.14.04: Por meio de uma aplicação específica, executar a técnica de *dump* de memória no equipamento de ensaio e depois verificar nos dados de memória coletados se há indícios do código PIN em cache.

EN.II.14.05: Retirar o cartão inteligente da leitora, colocá-lo novamente e verificar se há a solicitação do PIN antes da realização da primeira operação criptográfica.

EN.II.14.06: Encerrar a sessão do Windows, reiniciá-lo e verificar se há a solicitação do PIN antes da realização da primeira operação criptográfica.

REQUISITO IV.15: Se o serviço gerenciador de cartões inteligentes do Windows (*Smart Card Service*) for desativado, todos os contextos e *handles* em uso pelo CSP devem ser invalidados.

Procedimentos de ensaio para NSH 1,2 e 3:

EN.IV.15.01: Desativar o *Smart Card Service* e verificar se todos os contextos e *handles* em uso pelo CSP foram invalidados.

REQUISITO IV.16: O CSP deve suportar hibernação. Todos os *handles*, contextos e sessões devem estar válidos após o processo de hibernação.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.16.01: Verificar na documentação se o CSP suporta hibernação.

EN.IV.16.02: Se o CSP suportar hibernação, verificar se todos os *handles*, contextos e sessões estão válidos após o processo de hibernação.

REQUISITO IV.17: A implementação de MS *CryptoAPI* deve suportar os algoritmos criptográficos descritos na seção “Algoritmos criptográficos obrigatórios”.

EN.IV.17.01: Verificar se a documentação que acompanha o CSP especifica o suporte aos algoritmos criptográficos descritos na seção “Algoritmos Criptográficos Obrigatórios” por meio de interface MS *CryptoAPI*.

2.4.4.3 Requisitos sobre PKCS#11

REQUISITO IV.18: O CSP deve suportar uma implementação PKCS#11 na versão no mínimo 2.11.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.18.01: Verificar na documentação se o CSP suporta uma implementação PKCS#11.

EN.IV.18.02: Verificar se a documentação que acompanha o CSP especifica a versão do PKCS#11 suportado.

REQUISITO IV.19: O CSP deve suportar as seguintes chamadas de PKCS#11 (*Cryptoki*):

- *C_Initalize*
- *C_Finalize*
- *C_OpenSession*
- *C_CloseSession*
- *C_Init_Token*
- *C_Init_PIN*
- *C_Login*
- *C_Logout*
- *C_CreateObject*
- *C_DestroyObject*
- *C_GetAttributeValue*
- *C_SetAttributeValue*

- *C_EncryptInit*
- *C_Encrypt*
- *C_DecryptInit*
- *C_Decrypt*
- *C_DigestInit*
- *C_Digest*
- *C_DigestKey*
- *C_SignInit*
- *C_Sign*
- *C_VerifyInit*
- *C_Verify*
- *C_GenerateKey*
- *C_GenerateKeyPair*
- *C_DeriveKey*
- *C_GenerateRandom*

Sendo obrigatória a implementação das seguintes funções:

- *C_GenerateKey* especificando templates de chaves simétricas;
- *C_GenerateKeyPair* especificando templates de chaves assimétricas;
- *C_Sign* para realizar assinatura de um conteúdo;
- *C_Verify* para verificar a assinatura de um conteúdo;
- *C_Encrypt* para cifrar um dado com uma chave já construída;
- *C_Decrypt* para decifrar um dado com uma chave já construída;
- *C_CreateObject* especificando templates de chaves assimétricas (no mínimo chave pública);
- *C_DestroyObject* especificando o *handle* do objeto.

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.19.01: Verificar na documentação se o CSP exporta as chamadas citadas no requisito.

EN.IV.12.02: Verificar na documentação se o CSP implementa as funções tidas como obrigatórias.

REQUISITO IV.20: A implementação PKCS#11 deve suportar os algoritmos criptográficos descritos na seção "Algoritmos criptográficos obrigatórios".

EN.IV.20.01: Verificar se a documentação que acompanha o CSP especifica o suporte aos algoritmos criptográficos descritos na seção "Algoritmos criptográficos obrigatórios" por meio de interface PKCS#11.

2.4.4.4 Requisitos sobre Java Cryptographic Extension (JCE)

REQUISITO IV.21: O pacote de classes JCE deve ser suportado pela versão da máquina virtual Java (no mínimo V.1.4.2).

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.21.01: Verificar na documentação se o CSP suporta uma implementação JCE.

EN.IV.21.02: Verificar se a documentação que acompanha o CSP especifica a versão da máquina virtual Java a ser suportada pelo pacote de classes JCE.

REQUISITO IV.22: O CSP deve suportar, no mínimo, as seguintes classes de JCE [Java 2 SDK]:

- *MessageDigest*
- *Signature*
- *KeyPairGenerator*
- *KeyFactory*
- *CertificateFactory*
- *KeyStore*

- *AlgorithmParameters*
- *AlgorithmParameterGenerator*
- *SecureRandom*
- *CertStore*

Procedimentos de ensaio para NSH 1, 2 e 3:

EN.IV.22.01: Verificar na documentação se o CSP suporta as classes JCE tidas como obrigatórias.

REQUISITO IV.23: A documentação deve especificar os componentes de software implementados do provedor de serviço criptográfico.

EN.IV.23.01: Verificar se a documentação que acompanha o CSP especifica os componentes de software implementados do provedor de serviço criptográfico, de acordo com o documento de especificação da API *Java Cryptography Architecture API Specification & Reference*.

REQUISITO IV.24: A documentação deve especificar o processo de configuração e instalação do provedor de serviço criptográfico.

EN.IV.24.01: Verificar se a documentação que acompanha o CSP especifica o processo de configuração e instalação do CSP na máquina virtual Java.

REQUISITO IV.25: A documentação deve especificar serviços criptográficos implementados no provedor de serviço criptográfico que não estejam na especificação JCE versão 1.4 ou superior.

EN.IV.25.01: Verificar se a documentação que acompanha o CSP especifica os serviços criptográficos implementados no CSP que não estejam na especificação JCE versão 1.4 ou superior.

REQUISITO IV.26: A documentação deve informar detalhes sobre o uso do provedor de serviço criptográfico como API no formato Javadoc com trechos de código-fonte.

EN.IV.26.01: Verificar se a documentação que acompanha o CSP informa detalhes sobre o uso do CSP como API no formato Javadoc com trechos de código-fonte

REQUISITO IV.27: A implementação JCE deve suportar os algoritmos criptográficos descritos na seção "Algoritmos Criptográficos Obrigatórios".

EN.IV.27.01: Verificar se a documentação que acompanha o CSP especifica o suporte aos algoritmos criptográficos descritos na seção "Algoritmos criptográficos obrigatórios" por meio de interface JCE.

RECOMENDAÇÃO IV.03: O provedor de serviço criptográfico pode ser assinado por uma chave privada ligado a um certificado digital reconhecido no âmbito ICP-Brasil.

EN.REC.IV.03.01: Verificar na documentação que acompanha o CSP se o CSP é assinado por uma chave privada ligada a um certificado digital reconhecido no âmbito ICP-Brasil.

2.4.4.5 Requisitos sobre OpenSSL

Nota: Os ensaios referentes a esta seção serão realizados por meio de uma ferramenta de software ou utilitário desenvolvido pelo LEA, com base nas APIs e nas plataformas de sistemas operacionais.

REQUISITO IV.28: O CSP deve ser capaz de implementar as seguintes rotinas do OpenSSL Engine:

- *ENGINE_init;*
- *ENGINE_finish;*

- *bind_fn;*
- *Engine_load;*
- *ENGINE_load_private_key;*
- *ENGINE_load_public_key;*
- *bind_helper;*
- *ENGINE_destroy;*
- Dentre as funções requeridas para operações RSA estão (RSA_METHOD):
    - *RSA_init;*
    - *RSA_finish;*
    - *RSA_pub_dec ou RSA_verify (1);*
    - *RSA_priv_enc ou RSA_sign (1);*
    - *RSA_pub_enc;*
    - *RSA_priv_dec;*

      OBS: (1) Por questão de compatibilidade o OpenSSL ainda mantém as duas funções, tendo um campo para setar um flag (RSA_FLAG_SIGN_VER) de que versão é suportada.
- Funções requeridas para geração de números aleatórios (RAND_METHOD):
    - *RAND_bytes;*
    - *RAND_pseudo_bytes;*
    - *RAND_status.*

Procedimentos de ensaio para NSH 1:

EN.IV.28.01: Analisar se a documentação e verificar se foram implementadas as rotinas do OpenSSL Engine tidos como obrigatórios.

Procedimentos de ensaio para NSH 2 e 3:

EN.IV.28.02: Verificar se as funções e definições do OpenSSL Engine tidas como obrigatórias para um OpenSSL Engine estão operacionais.

REQUISITO IV.29: A implementação do OpenSSL Engine deve suportar os algoritmos criptográficos descritos na seção “Algoritmos criptográficos obrigatórios”.

EN.IV.29.01: Verificar se a documentação que acompanha o CSP especifica o suporte aos algoritmos criptográficos descritos na seção “Algoritmos criptográficos obrigatórios” por meio de interface OpenSSl Engine.

3 Referências normativas

[1] **Cryptographic Service Providers – MSDN (Microsoft Developer Network).** Disponível em: <http://msdn2.microsoft.com/en-us/library/aa380245.aspx>. Acesso em: 20.jul.2007.

[2] COMITÊ GESTOR DA ICP-BRASIL. **DOC ICP-01.01: Padrões e Algoritmos Criptográficos da Infra-Estrutura de Chaves Públicas Brasileira (ICP-BRASIL).** Versão 1.0. Brasília. ICP-BRASIL: 2006.

[3] [ITI] INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **Manual de Condutas Técnicas – Volume 16: Requisitos Técnicos para CSP.** Versão 1.0.

[4] [ITI] GLOSSÁRIO ICP-BR – INFRA-ESTRUTURA DE CHAVES PÚBLICAS BRASILEIRAS. **Glossário ICP-Brasil.** Versão 1.2. Brasília. ICP – BR: 2007.

[5] MESSIER, Matt e VIEGA, John. **Secure Programming Cookbook for C and C++**. O'Reilly Publisher: July, 2003. ISBN 0-596-00394-3.

[6] [FIPS / NIST] DEPARTMENT OF COMMERCE, NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, [ITL] INFORMATION TECHNOLOGY LABORATORY. **Federal Information Processing Standards Publication.** Washington. US Government Printing Office: 2001. Disponível em: <http://www.itl.nist.gov/fipspubs/>. Acesso em: 20.jul.2007.

[7] [NIST SP 800-17] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **Modes of Operation Validation System (MOVS): Requirements and Procedures.** 1998. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-17/800-17.pdf> Acesso em: 25 jul. 2005.

[8] [NIST SP 800-20] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **Modes of Operation Validation System for the Triple Data Encryption Algorithm (TMOVS): Requirements and Procedures.** 2000. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-20/800-20.pdf> Acesso em: 25 jul. 2005.

[9] [NIST. FIPS 46-3]. **Data Encryption Standard (DES).** 1999. Disponível em: <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>. Acesso em: 20.jul. 2007.

[10] [NIST] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **The RSA Validation System (RSAVS).** 2004. Disponível em: <http://csrc.nist.gov/cryptval/dss/RSAVS.pdf>. Acesso em: 25 jul. 2005.

[11] [RSA LABORATORIES] **PKCS#1: RSA Cryptography Standard.** Version 2.1. 2002. Disponível em: <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf>. Acesso em: 30.nov.2006.

[12] [NIST FIPS 186-2] **Digital Signatura Standard (DSS)**. 2001. Disponível em: <http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf>. Acesso em: 20.jul.2007.

[13] [NIST FIPS 196] **Entity Authentication Using Public Key Criptography**. 1997. Disponível em: <http://csrc.nist.gov/publications/fips/fips196/fips196.pdf>. Acesso em: 20.jul.2007.

[14] [NIST] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **The Secure Hash Algorithm Validation System (SHAVS).** 2004. Disponível em: <http://csrc.nist.gov/cryptval/shs/SHAVS.pdf>. Acesso em: 25 jul. 2005.

[15] [NIST FIPS 180-2] **Secure Hash Standard (SHA).** 2001. Disponível em: <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>. Acesso em: 20.jul.2007.

[16] [NIST] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **The Advanced Encryption Standard Algorithm Validation Suite (AESAVS).** 2002. Disponível em: <http://csrc.nist.gov/cryptval/aes/AESAVS.pdf>. Acesso em: 25 jul. 2005.

[17] [NIST FIPS 197] **Advanced Encryption Standard (AES)**. 2001. Disponível em: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>. Acesso em: 20.jul.2007.

[18] [ANSI. X9.31] AMERICAN NATIONAL STANDARDS INSTITUTE. **Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA).** 1998.

[19] [NIST FIPS 198] **The Keyed-Hash Message Authentication Code (HMAC)**. 2002. Disponível em: <http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf>. Acesso em: 20.jul.2007.

[20] [NIST Special Publication 800-38B] **Recommendation for Block Cipher Modes of Operation - The CMAC Mode for Authentication.** 2005. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf>. Acesso em: 20.jul.2007.

[21] [NIST / FIPS Special Publication 800-38C] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. **Counter with Cipher Block Chaining-Message Authentication Code (CCM)**. 2004. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C.pdf>. Acesso em: 23.jul.2007.

[22] [RSA LABORATORIES] **PKCS#5: Password-Based Cryptography Standard.** Version 2.0. 1999. Disponível em: <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-5v2/pkcs5v2-0.pdf>. Acesso em: 30.nov.2006.

[23] [RSA LABORATORIES] **PKCS#12: Personal Information Exchange Syntax Standard**. Version 1.0. 1999. Disponível em: <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf>. Acesso em: 04.jul.2007.

[24] [RSA LABORATORIES] **CMS: Cryptographic Message Syntax Standard.** Version 1.5. 1993. Disponível em: <ftp://ftp.rsasecurity.com/pub/pkcs/ps/pkcs-7.ps>. Acesso em: 27.abril.2007.

[25] [RSA LABORATORIES] **PKCS#8: Private-Key Information Syntax Standard.** Version 1.2. 1993. Disponível em: <ftp://ftp.rsasecurity.com/pub/pkcs/ps/pkcs-8.ps>. Acesso em: 27.abril.2007.

[26] **Cryptography (Windows) – MSDN (Microsoft Developer Network).** Disponível em: <http://msdn2.microsoft.com/en-us/library/aa380255.aspx>. Acesso em: 20.jul.2007.

[27] [RSA LABORATORIES] **PKCS#11: Cryptographic Token Interface Standard**. Version 2.0. 1997. Disponível em: <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/pkcs11v2.pdf> Acesso em: 04.jul.2007.

[28] **Java Cryptography Extension (JCE) for the Java 2 SDK,** versão 1.4. Disponível em: <http://java.sun.com/products/jce/index-14.html>. Acesso em: 20.jul.2007.

[29] **[OpenSSL FIPS 1402] Security Policy Object Module By the Open Source Software Institute.** Version 1.0a, March 24, 2006. Disponível em <http://csrc.nist.gov/cryptval/140-1/140sp/140sp642.pdf>. Acesso em: 20.jul.2007.

[30] [IN 01/2007 – ITI] INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **Instrução normativa 01/2007: Procedimentos administrativos a serem observados nos processos de homologação de sistemas e**

equipamentos de certificação digital no âmbito da ICP-Brasil. DOC-ICP-10.01 versão 2.1. Brasília. ICP-Brasil: 2007.

[31] [IN 02/2007 – ITI] INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **Instrução normativa 02/2007: Estrutura normativa técnica e níveis de segurança de homologação a serem utilizados nos processos de homologação de sistemas e equipamentos de certificação digital no âmbito ICP-Brasil.** DOC-ICP-10.02 versão 2.0. Brasília. ICP-Brasil: 2007.

[32] [IN 06/2007 – ITI] INSTITUTO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO. **Instrução normativa 06/2007: Padrões e procedimentos técnicos a serem observados nos processos de homologação de bibliotecas criptográficas e softwares provedores de serviços criptográficos no âmbito da ICP-Brasil.** DOC-ICP-10.06 versão 1.0. Brasília. ICP-Brasil: 2007.