

EVALCryptoSPBSecDLL 5.0.9.0

Manual de integração



SOBRE ESTE MANUAL

Publicado por

E-VAL Tecnologia em Informática Ltda.

Rua Paulistânia, nº 381, 1º Andar

Sumarezinho - São Paulo – SP – Brasil

CEP 05440-000

Fone/Fax: +55 (11) 3670-3825

<http://www.evaltec.com.br>

Avisos e marcas registradas

Este software e sua documentação são de propriedade da E-VAL Tecnologia em Informática Ltda. sendo seu uso e distribuição sujeitos a licença de uso.

Outras companhias, produtos ou nomes de serviços também podem ser marcas registradas.

e-val

Sumário

1	INTRODUÇÃO	1
1.1	PROPÓSITO E ESCOPO DO MANUAL	1
1.2	PÚBLICO ALVO	1
1.3	ORGANIZAÇÃO DO DOCUMENTO	1
1.4	CONVENÇÕES	2
1.5	PRODUTOS RELACIONADOS E-VAL	2
1.6	SUPORTE	2
2	PRÉ-REQUISITOS E RECOMENDAÇÕES INICIAIS	3
3	PROCESSO DE INTEGRAÇÃO	4
3.1	ANTES DA INTEGRAÇÃO	4
3.2	DURANTE A INTEGRAÇÃO	4
3.3	APÓS A INTEGRAÇÃO	4
4	ARQUITETURA	5
4.1	PROTOCOLO CRYPTOSPB	5
5	BIBLIOTECA DE COMUNICAÇÃO SPBSEC DLL	7
5.1	FUNCIONALIDADES PRINCIPAIS	7
5.1.1	REDUNDÂNCIA E BALANCEAMENTO DE CARGA	8
6	INTERFACES DE PROGRAMAÇÃO	10
6.1	INITIALIZECONNSRV()	11
6.1.1	SINTAXE VISUAL BASIC	11
6.1.2	SINTAXE NA LINGUAGEM C	11
6.1.3	PARÂMETROS	11
6.1.4	VALORES DE RETORNO	11
6.2	INITIALIZECONN()	12
6.2.1	SINTAXE EM VISUAL BASIC	12
6.2.2	SINTAXE NA LINGUAGEM C	12
6.2.3	VALORES DE RETORNO	12
6.3	ENCRYPTMSG()	13
6.3.1	SINTAXE VISUAL BASIC	13
6.3.2	SINTAXE NA LINGUAGEM C	13

6.3.3	SINTAXE NA LINGUAGEM C - VERSÃO 5.0.2	14
6.3.4	SINTAXE NA LINGUAGEM C - VERSÃO 5.0.7.0	14
6.3.5	PARÂMETROS.....	15
6.3.6	NOVOS PARÂMETROS - VERSÃO 5.0.2 E 5.0.7	17
6.3.7	NOVOS PARÂMETROS - VERSÃO 5.0.8.	17
6.3.8	SINTAXE NA LINGUAGEM C – VERSÃO 5.0.9.0.....	18
6.3.9	NOVOS PARÂMETROS - VERSÃO 5.0.9.	19
6.3.10	VALORES DE RETORNO	19
6.3.11	PARÂMETRO VLOG.....	20
6.4	DECRYPTMSG()	21
6.4.1	SINTAXE VISUAL BASIC.....	21
6.4.2	SINTAXE NA LINGUAGEM C.....	21
6.4.3	SINTAXE NA LINGUAGEM C - VERSÃO 5.0.2	22
6.4.4	SINTAXE NA LINGUAGEM C - VERSÃO 5.0.7.0	22
6.4.5	PARÂMETROS.....	23
6.4.6	NOVOS PARÂMETROS - VERSÃO 5.0.2 E 5.0.6.....	24
6.4.7	SINTAXE NA LINGUAGEM C - VERSÃO 5.0.9.0	24
6.4.8	PARÂMETROS.....	25
6.4.9	PARÂMETRO VLOG	26
6.4.10	VALORES DE RETORNO	26
6.5	TERMINATEALLCONN()	27
6.5.1	SINTAXE VISUAL BASIC.....	27
6.5.2	SINTAXE NA LINGUAGEM C.....	27
6.5.3	VALORES DE RETORNO	27
6.6	TERMINATECONN()	28
6.6.1	SINTAXE VISUAL BASIC.....	28
6.6.2	SINTAXE NA LINGUAGEM C.....	28
6.6.3	PARÂMETROS.....	28
6.6.4	VALORES DE RETORNO	28
6.7	RECONNECT()	29
6.7.1	SINTAXE VISUAL BASIC.....	29
6.7.2	SINTAXE NA LINGUAGEM C.....	29
6.7.3	PARÂMETROS.....	29
6.7.4	VALORES DE RETORNO	29
6.8	STATUS DAS CONEXÕES COM OS SERVIDORES DE SEGURANÇA	30

6.8.1	SINTAXE VISUAL BASIC.....	30
6.8.2	SINTAXE NA LINGUAGEM C.....	30
6.8.3	PARÂMETROS.....	31
6.8.4	VALORES DE RETORNO	31
6.9	SERVIDORES DISPONÍVEIS	32
6.9.1	SINTAXE NA LINGUAGEM C.....	32
6.9.2	VALORES DE RETORNO	32
6.10	CONEXÕES ATIVAS COM OS SERVIDORES DE SEGURANÇA.....	33
6.10.1	SINTAXE NA LINGUAGEM C.....	33
6.10.2	VALORES DE RETORNO	33
6.11	LIBERA A MEMORIA.....	34
6.11.1	SINTAXE NA LINGUAGEM C.....	34
6.11.2	VALORES DE RETORNO	34
6.12	CONVERSÃO DE VARIANT PARA CHAR *	35
6.12.1	SINTAXE NA LINGUAGEM VISUAL BASIC	35
6.12.2	SINTAXE NA LINGUAGEM C.....	35
6.12.3	VALORES DE RETORNO	35
6.13	PARÂMETROS DE CONFIGURAÇÃO DA DLL SPB_SECDLL	36
	DETECÇÃO DE ERROS NA MENSAGEM.....	38
	ERROS ESPECÍFICOS DO EVALCRYPTOSFNMSG	39

LISTA DE TABELAS

Tabela 1. Convenções tipográficas - ícones.	2
Tabela 2. Convenções tipográficas - estilos.	2
Tabela 3. Lista de funções disponíveis.	10

e-val

LISTA DE FIGURAS

Figura 1. Protocolo CryptoSPB.	5
Figura 2. Formato de um comando de extração entre Mensageria e EVALCryptoSFNMsg.	6
Figura 3. Comando de envelopamento entre Mensageria e EVALCryptoSFNMsg.	6
Figura 4. Biblioteca SPBSecDLL.....	7
Figura 5. API da biblioteca SPBSecDLL.	7
Figura 6. Mensageria conectada a diversos EVALCryptoSFNMsg.	8
Figura 7. Diversas mensagerias conectadas a diversos EVALCryptoSFNMsg.....	9

1 Introdução

1.1 Propósito e escopo do manual

O objetivo deste manual é descrever todos os aspectos relevantes do processo de instalação do EVALCryptoSPBSecDLL.

1.2 Público alvo

Este manual destina-se aos responsáveis por instalar o EVALCryptoSPBSecDLL.

1.3 Organização do documento

Este manual está organizado nas seguintes seções:

- **Seção 1 – Introdução.** Essa seção provê informações sobre o propósito e o escopo do manual, como ele está estruturado e as referências externas que devem ser consideradas;
- **Seção 2 – Pré-requisitos e recomendações iniciais.** Essa seção provê informações sobre os pré-requisitos de *hardware* e *software* que devem ser considerados para uma utilização bem sucedida do EVALCryptoSPBSecDLL;
- **Seção 3 – Processo de integração.** Essa seção descreve os métodos e propriedades disponíveis para integração do EVALCryptoSPBSecDLL com sua aplicação.
- **Seção 4 – Arquitetura.** Essa seção descreve a arquitetura do EVALCryptoSFNMsg e contextualiza a utilização do EVALCryptoSPBSecDLL.
- **Seção 5 – A biblioteca de comunicação EVALCryptoSPBSecDLL.** Essa seção indica o papel do EVALCryptoSPBSecDLL na arquitetura descrita na seção anterior.
- **Seção 6 – Interfaces de programação.** Essa seção descreve a interface de programação da EVALCryptoSPBSecDLL, incluindo todos os métodos e parâmetros de entrada e saída.

1.4 Convenções

Algumas frases desse manual merecem destaque, seja pelo fato de indicarem instruções importantes, seja pelo fato de indicar informações complementares. Para isso, em caráter puramente visual, utilizamos as convenções listadas nas Tabela 1.





CONVENÇÕES TIPOGRÁFICAS – ÍCONES	
ÍCONE	UTILIZAÇÃO
	Ícone utilizado para registrar dicas.
	Ícone utilizado para registrar notas.
	Ícone utilizado para indicar avisos importantes,
	Ícone utilizado para indicar lembretes.

Tabela 1. Convenções tipográficas - ícones.

Adicionalmente, em caráter textual, utilizamos as convenções listadas na Tabela 2.

CONVENÇÕES TIPOGRÁFICAS – ESTILOS	
ESTILO	UTILIZAÇÃO
<i>Itálico</i>	Estilo que indica um termo técnico e/ou termo em língua estrangeira.
Negrito	Estilo que indica um conceito importante. Quando usado, o termo e seu significado podem ser encontrados no Glossário.
<code>Comando</code>	Estilo que representa instruções, comandos de execução de programa e código-fonte.

Tabela 2. Convenções tipográficas - estilos.

1.5 Produtos relacionados E-VAL

Consulte <http://www.evaltec.com.br> para conhecer produtos relacionados ou outras linhas de soluções que a E-VAL Tecnologia em Informática oferece ao mercado.

Para dúvidas comerciais entre em contato com comercial@evaltec.com.br.

1.6 Suporte

Para suporte ao produto entre em contato com a E-VAL:

suporte@evaltec.com.br

Suporte E-VAL Tecnologia em Informática Ltda.

Telefone/Fax: +55 (11) 3670-3825.

2 Pré-requisitos e recomendações iniciais

A EVALCryptoSPBSecDLL utiliza alguns registros do Windows para armazenar as configurações. Esses registros são criados executando o “SPB_SecDll.reg” ou “SPB_SecDll_win64.reg” disponibilizado juntamente com a DLL.

A **SPB_SecDll.reg** deve ser utilizada nas versões x86 do Windows e a **SPB_SecDll_win64.reg** para versões x64 do Windows.

Os registros abaixo serão criados pra WIN32:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Robo]
[HKEY_LOCAL_MACHINE\SOFTWARE\Robo\SPB]
[HKEY_LOCAL_MACHINE\SOFTWARE\Robo\SPB\Log]
[HKEY_LOCAL_MACHINE\SOFTWARE\Robo\SPB\SECURITY SERVICE]
```

ou para x64:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Robo]
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Robo\SPB]
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Robo\SPB\Log]
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Robo\SPB\SECURITY SERVICE]
```

Esses registros devem ser criados antes do início da integração com a dll.

3 Processo de integração

O processo de integração com o EVALCryptoSPBSecDLL está organizado nas seguintes seções:

1. **Antes da integração...** – seção que lista todas as recomendações que DEVEM ser seguidas antes de que a integração com o produto seja de fato realizada;
2. **Durante a integração...** – seção que lista todas as recomendações que DEVEM ser seguidas ao longo da integração com o produto; e
3. **Após a integração...** – seção que lista todas as recomendações que DEVEM ser seguidas após a integração bem sucedida com o produto.

3.1 Antes da integração...

Antes da integração assegure-se de que os processos de instalação descritos no(s) Manual(is) de Instalação e os processos de configuração descritos no(s) Manual(is) de Configuração que acompanham o pacote de instalação desse produto foram concluídos de maneira bem sucedida.

3.2 Durante a integração...

Durante a integração utilize as informações contidas nas seções 4, 5 e 6. Esta última contém a referência completa sobre a interface de programação da EVALCryptoSPBSecDLL.

3.3 Após a integração...

Atualmente, após uma integração bem sucedida, não há recomendações a serem seguidas.

4 Arquitetura

O EVALCryptoSFNMsg recebe requisições de clientes utilizando um protocolo de comunicação chamado “Protocolo CryptoSPB”.

4.1 Protocolo CryptoSPB

O protocolo CryptoSPB, ilustrado na Figura 1, define uma forma padronizada de interação entre a Mensageria e o EVALCryptoSFNMsg.

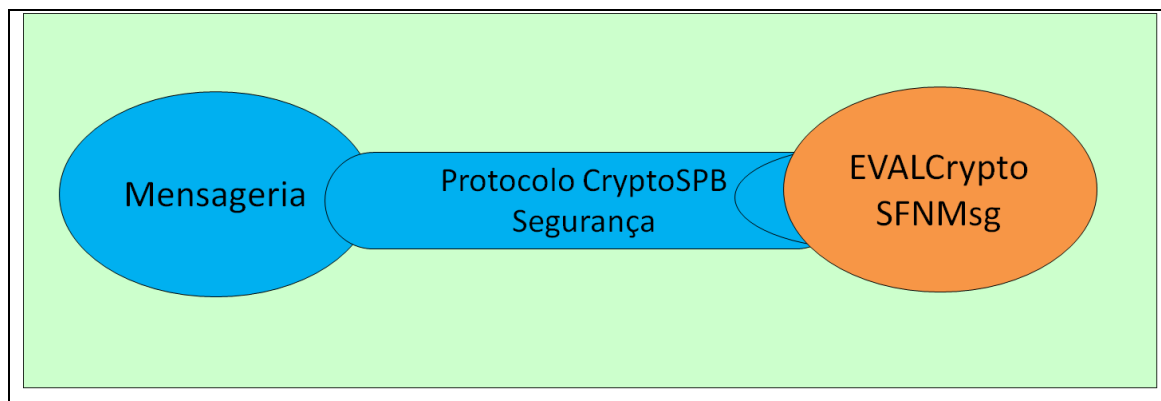


Figura 1. Protocolo CryptoSPB.

Quando a Mensageria recebe uma mensagem que esteja criptografada e assinada proveniente de uma entidade participante do SPB, ela repassa esta mensagem para o EVALCryptoSFNMsg acrescentando o “Cabeçalho CryptoSPB” que contém basicamente informações de controle entre Mensageria e EVALCryptoSFNMsg. O EVALCryptoSFNMsg decifra e verifica a assinatura da mensagem, restaurando o formato original. Em seguida responde à Mensageria acrescentando opcionalmente o *log* da operação. Este processamento está descrito na Figura 2.

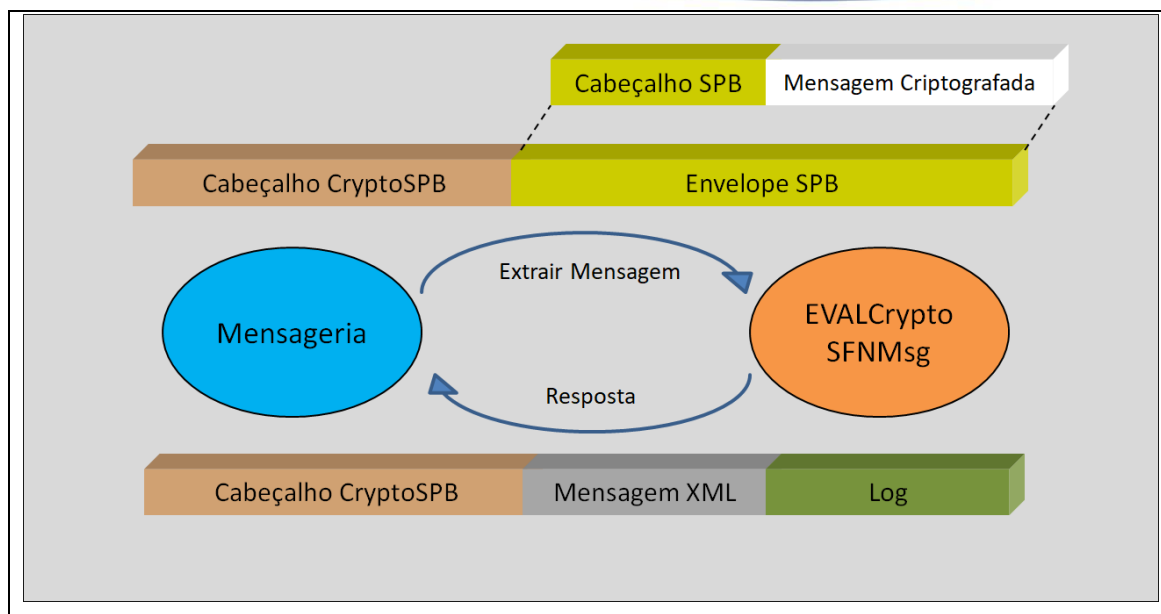


Figura 2. Formato de um comando de extração entre Mensageria e EVALCryptoSFNMsg.

De modo análogo, quando a Mensageria recebe do ambiente interno uma mensagem para ser enviada a uma entidade participante do SPB também irá contatar o EVALCryptoSFNMsg para assinar e criptografar a mensagem. A Mensageria envia um comando ao EVALCryptoSFNMsg requisitando o envelopamento (criptografia) da mensagem SPB. O EVALCryptoSFNMsg assina e cifra a mensagem, gerando o envelope digital, e imediatamente respondendo à Mensageria com a mensagem SPB envelopada juntamente com um conjunto de informações a respeito do *log*. Este processamento está ilustrado na Figura 3.

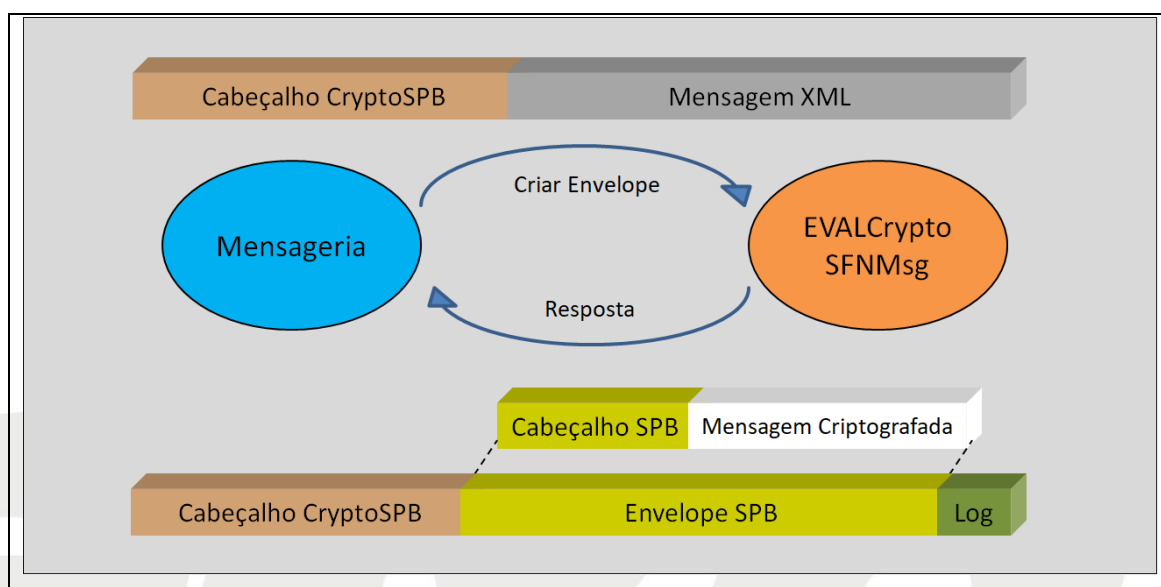


Figura 3. Comando de envelopamento entre Mensageria e EVALCryptoSFNMsg.

5 Biblioteca de comunicação SPBSecDLL

Para facilitar a utilização do EVALCryptoSFNMsg pela mensageria foi desenvolvida a **biblioteca SPBSecDLL**, cujas funções podem ser acionadas para realizar a interação com o EVALCryptoSFNMsg, como ilustrado na Figura 4.

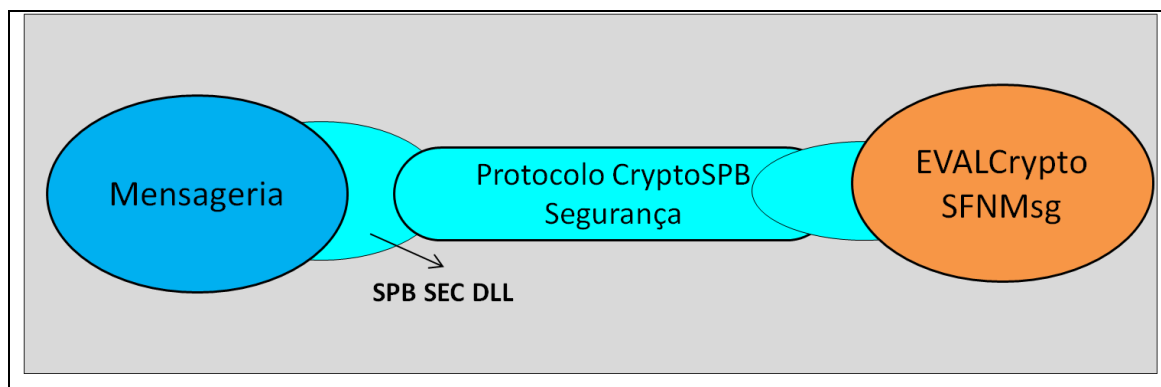


Figura 4. Biblioteca SPBSecDLL.

5.1 Funcionalidades principais

A SPBSecDLL é responsável por todo o controle da comunicação encapsulando a complexidade do gerenciamento da comunicação. As principais funcionalidades são as seguintes:

1. Comunicação com o EVALCryptoSFNMsg.
2. Redundância e tolerância a falhas.
3. Balanceamento de carga.

A Mensageria utiliza a biblioteca SPBSecDLL conforme ilustrado na Figura 5

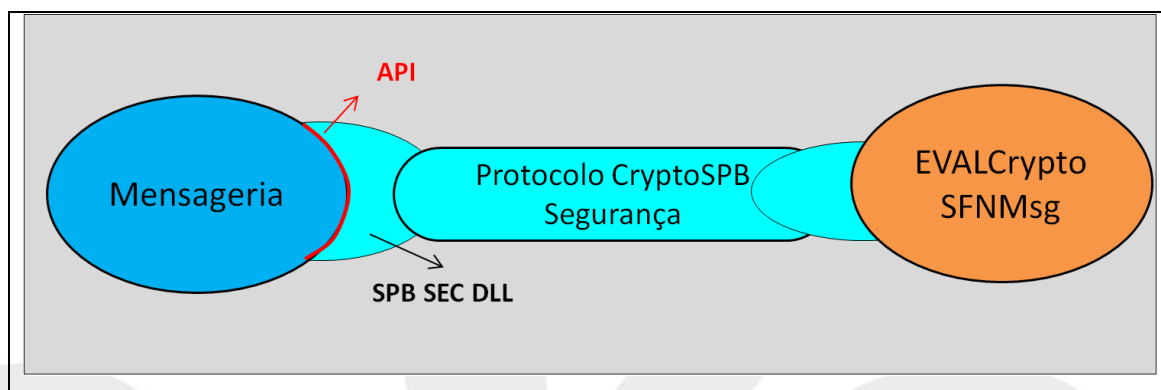


Figura 5. API da biblioteca SPBSecDLL.

5.1.1 Redundância e balanceamento de carga

Cada entidade cliente (da Mensageria) pode estabelecer conexão a mais de um EVALCryptoSFNMsg ao mesmo tempo. A Figura 6 mostra um exemplo desta situação. Isto possibilita programar funcionalidades de redundância e balanceamento de carga. O controle de como as mensagens são distribuídas entre os EVALCryptoSFNMsg é realizado de forma transparente pela biblioteca SPBSecDLL.

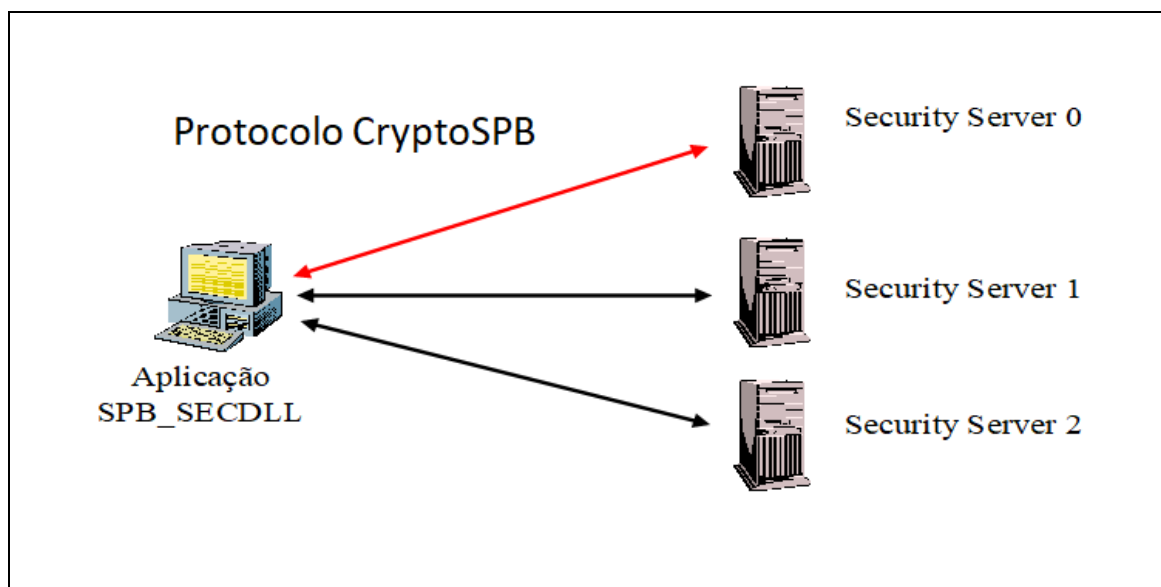


Figura 6. Mensageria conectada a diversos EVALCryptoSFNMsg.

Em um determinado momento podem existir vários clientes conectados ao EVALCryptoSFNMsg, como mostrado na Figura 6.

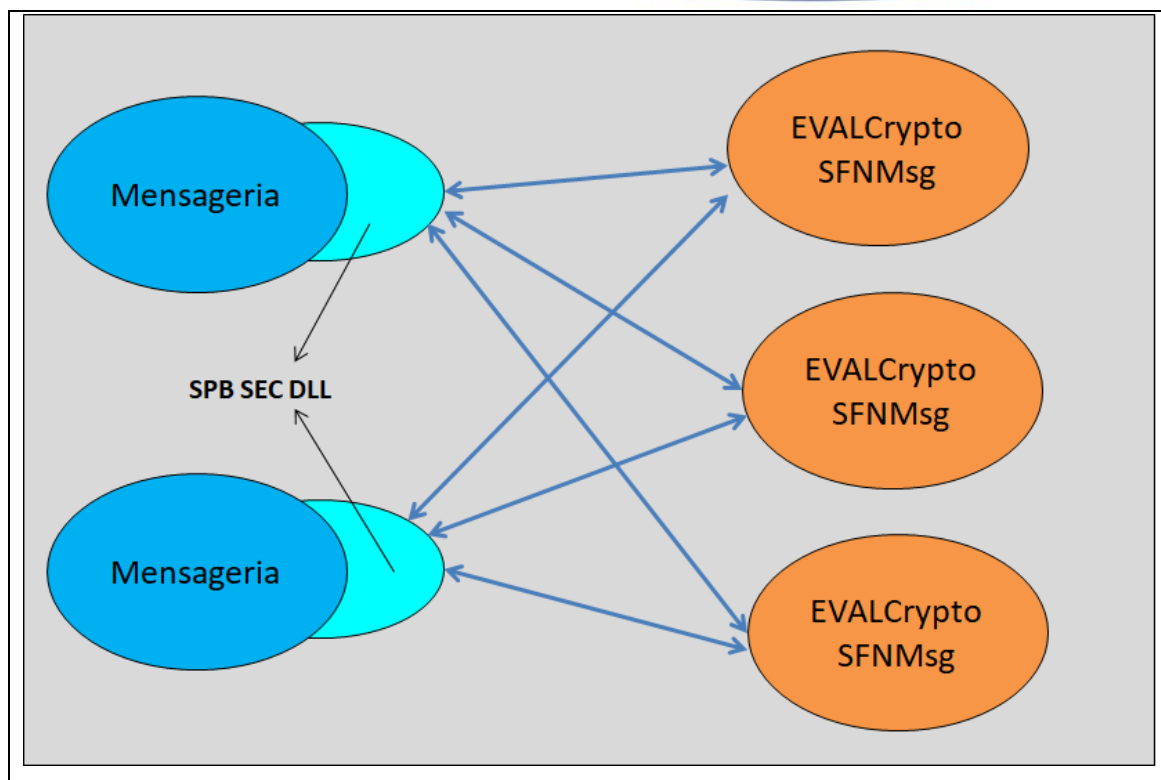


Figura 7. Diversas mensagerias conectadas a diversos EVALCryptoSFNMsg.

O EVALCryptoSFNMsg programa um mecanismo de alta disponibilidade na biblioteca de comunicação, não necessitando da utilização de outros sistemas deste tipo como aqueles baseados em *cluster*.

6 Interfaces de programação

A biblioteca SPB_SECDLL disponibiliza as seguintes funções:

LISTA DE FUNÇÕES DISPONÍVEIS		
Id	Função	Descrição
1	InitializeConnSrv	Veja seção 6.1.
2	InitializeConn	Veja seção 6.2.
3	EncryptMsg	Veja seção 6.3.
4	DecryptMsg	Veja seção 6.4.
5	TerminateAllConn	Veja seção 6.5.
6	TerminateConn	Veja seção 6.6.
7	ReConnect	Veja seção 6.7.
8	StatusConn	Veja seção 6.8.
9	GetServerPoolSize	Veja seção 6.9.
10	GetConnectedServers	Veja seção 6.10.
11	FreeMemory	Veja seção 6.11.
12	VariantToChar	Veja seção 6.12.

Tabela 3. Lista de funções disponíveis.



A função ReConnect tem por objetivo restabelecer conexões que tenham sido encerradas. A aplicação pode monitorar as conexões encerradas por meio da função StatusConn.

6.1 InitializeConnSrv()

Esta chamada é requerida durante a instanciação da DLL e irá comandar esta a executar os procedimentos de conexão SOCKET TCP com o serviço de segurança. Neste momento, a DLL estabelecerá uma conexão com cada servidor de segurança configurado no Registry. O parâmetro “nServers” retorna o número de servidores de criptografia disponíveis.

6.1.1 Sintaxe Visual Basic

```
Private Declare Function InitializeConnSrv Lib  
    "c:\winnt\system32\spb_sec.dll"  
    (ByRef nServers As Integer) As long  
lngtRetCode = InitializeConnSrv (nServers)
```

6.1.2 Sintaxe na linguagem C

```
extern "C" __declspec(dllexport)  
    int _stdcall InitializeConnSrv (int *nServers)
```

6.1.3 Parâmetros

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Direção</i>
nServers	Integer	Número de servidores de criptografia disponíveis	Saída

6.1.4 Valores de retorno

<i>Valor de Retorno</i>	<i>Descrição</i>
0	Conexão estabelecida com sucesso
1	Erro na conexão

6.2 InitializeConn()

Esta chamada é requerida durante a instanciação da DLL e irá comandar esta a executar os procedimentos de conexão SOCKET TCP com o serviço de segurança. Neste momento, a DLL estabelecerá uma conexão com cada servidor de segurança configurado no Registry.

6.2.1 Sintaxe em Visual Basic

```
Private Declare Function InitializeConn Lib  
    "c:\winnt\system32\spb_sec.dll" () As long  
lngtRetCode = InitializeConn(nServers)
```

6.2.2 Sintaxe na linguagem C

```
extern "C" __declspec( dllexport )  
    int _stdcall InitializeConn (INT *nServers)
```

6.2.3 Valores de retorno

<i>Valor de Retorno</i>	<i>Descrição</i>
0	Conexão estabelecida com sucesso
1	Erro na conexão

6.3 EncryptMsg()

Esta chamada deve ser utilizada para requisitar uma solicitação de criação do envelope digital SPB (assinatura e criptografia) para uma mensagem. Ela é utilizada pela mensageria logo antes do envio de uma mensagem para um parceiro externo.

6.3.1 Sintaxe Visual Basic



Atenção: Esta função cifra a mensagem conforme configuração no serviço EVALCryptoSFNMsg – Veja na seção 6.3.8, a versão 5.0.9 que permite selecionar o algoritmo de cifra.

```
Private Declare Function EncryptMsg Lib "c:\winnt\system32\spb_sec.dll"
(
    ByVal strDomain As String,
    ByVal strSISPB As String,
    ByVal strDISPB As String,
    ByVal iCodMsg As Integer,
    ByVal strDados As String,
    ByRef vCipherText As Variant,
    ByRef vLog As Variant,
    ByVal iCodErro As Integer
) As long

lngRetCode = EncryptMsg
(
    strDomain,
    strSISPB,
    strDISPB,
    iCodMsg,
    strXMLMessage,
    vCipherText
)
```

6.3.2 Sintaxe na linguagem C

Esta função cifra uma mensagens alterando a codificação da mensagem para UTF16BE.



Atenção: Esta função cifra a mensagem conforme configuração no serviço EVALCryptoSFNMsg – Veja na seção 6.3.8, a versão 5.0.9 que permite selecionar o algoritmo de cifra.

```
__declspec( dllexport ) int __stdcall EncryptMsg
(
    LPSTR lpstrDomain           // (in) Domain Name
    LPSTR lpstrSISPB,           // (in) Source ISPB
    LPSTR lpstrDISPB,           // (in) Destination ISPB
    INT nCodMsg,                // (in) Message Code
    LPSTR lpstrPlainText,       // (in) XML Message
    VARIANT *pvCipherText,      // (out) SpbEnvelop
    VARIANT *pvLog,             // (out) Symmetric Key, Digital Sign, XML Msg Uni16l
    INT nCodError               // (in) Error Code
)
```

6.3.3 Sintaxe na linguagem C - Versão 5.0.2

Esta função cifra uma mensagens alterando a codificação da mensagem para UTF16BE.



Atenção: Esta função cifra a mensagem conforme configuração no serviço EVALCryptoSFNMsg – Veja na seção 6.3.8, a versão 5.0.9 que permite selecionar o algoritmo de cifra.

```
__declspec( dllexport ) int __stdcall EncryptMsgB
(
    LPCSTR lpstrDomainName, // (in) Domain Name
    LPSTR lpstrSISPB,       // (in) Source ISPB
    LPSTR lpstrDISPB,       // (in) Destination ISPB
    INT intCodMsg,          // (in) MessageCode
    LPSTR lpstrPlainText,   // (in) XML Message
    INT lpstrPlainTextLen,  // (in) XML Message Length
    BYTE **lpCipherText,    // (out)SPB Envelop (Liberar com
FreeMemory())
    INT *intCipherTextLen,  // (out)SPB Envelop Length
    BYTE **lpLog,           // (out)Symmetric Key, Digital Sign, XML
(Unicode 16) (Liberar FreeMemory())
    INT *intLogLen,         // (out)SPB Log Length
    INT intCodError); // (in) ErrorCode
```

6.3.4 Sintaxe na linguagem C - Versão 5.0.7.0

Esta função cifra uma mensagens sem alterar a codificação da mensagem, ficando a cargo da aplicação codificar a mensagem em UTF8 ou UTF16BE.



Atenção: Esta função cifra a mensagem conforme configuração no serviço EVALCryptoSFNMsg – Veja na seção 6.3.8, a versão 5.0.9 que permite selecionar o algoritmo de cifra.

```
__declspec( dllexport )int _stdcall EncryptMsgC
(
    LPCSTR lpstrDomainName, // (in) Domain Name
    LPSTR lpstrSISPB,       // (in) Source ISPB
    LPSTR lpstrDISPB,       // (in) Destination ISPB
    INT    intCodMsg,        // (in) MessageCode
    LPSTR lpstrPlainText,    // (in) XML Message
    INT lpstrPlainTextLen,   // (in) XML Message Length
    BYTE **lpCipherText,    // (out)SPB Envelop (Liberar com
FreeMemory())
    INT *intCipherTextLen,   // (out)SPB Envelop Length
    BYTE **lpLog,           // (out)Log de auditoria do BACEN (Liberar
FreeMemory())
    INT *intLogLen,         // (out)SPB Log Length
    INT    intCodError); // (in) ErrorCode
```

6.3.5 Parâmetros



Atenção: Esta função cifra a mensagem conforme configuração no serviço EVALCryptoSFNMsg – Veja na seção 6.3.8, a versão 5.0.9 que permite selecionar o algoritmo de cifra.

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Direção</i>
strDomain	String	Nome do domínio	Entrada
strSISPB	String	ISPB da instituição de origem	Entrada
strDISPB	String	ISPB da instituição de destino	Entrada
iCodMsg	Integer	Código de Mensagens Genéricas. 4 – GEN0004 (somente assinada) 6 – GEN0006 (assinada e criptografada) 0 – Demais mensagens (assinada e criptografada)	Entrada
strXMLMessage	string	Mensagem para banco central no formato XML	Entrada
strPlainTextLen	Integer	Tamanho da mensagem	Entrada
vCipherText	variant	Envelope SPB cifrado	Saída

vLog	variant	Log da operação	Saída
iCodErro	Integer	Código de erro a ser informado no caso de uma GEN0004. Zero para outras mensagens	Entrada

6.3.6 Novos parâmetros - Versão 5.0.2 e 5.0.7



Atenção: Esta função cifra a mensagem conforme configuração no serviço EVALCryptoSFNMsg – Veja na seção 6.3.8, a versão 5.0.9 que permite selecionar o algoritmo de cifra.

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Direção</i>
strDomain	String	Nome do domínio	Entrada
strSISPB	String	ISPB da instituição de origem	Entrada
strDISPB	String	ISPB da instituição de destino	Entrada
iCodMsg	Integer	Código de Mensagens Genéricas. 0 - MSG_NORMAL - (XML assinado e criptografado). 4 – GEN0004 (somente assinada). 6 – GEN0006 (assinada e criptografada).	Entrada
strXMLMessage	string	Mensagem para banco central no formato XML	Entrada
lpCipherText	byte array	Envelope SPB cifrado	Saída
intCipherTextLen	Integer	Tamanho do envelope	Saída
lpLog	byte array	Log da operação	Saída
intLogLen	Integer	Tamanho do log	Saída
iCodErro	Integer	Código de erro a ser informado no caso de uma GEN0004. Zero para outras mensagens	Entrada

6.3.7 Novos parâmetros - Versão 5.0.8.

Na versão 5.0.8 foi adicionada a opção de cifra de arquivos ZIP. Para isso, deve-se utilizar a opção ZIP_CIFRADO no parâmetro iCodMsg.



Atenção: Esta função cifra a mensagem conforme configuração no serviço EVALCryptoSFNMsg – Veja na seção 6.3.8, a versão 5.0.9 que permite selecionar o algoritmo de cifra.

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Direção</i>
strDomain	String	Nome do domínio	Entrada

strSISPB	String	ISPB da instituição de origem	Entrada
strDISPB	String	ISPB da instituição de destino	Entrada
iCodMsg	Integer	Código de Mensagens Genéricas. 0 - MSG_NORMAL - (XML assinado e criptografado). 4 – GEN0004 (somente assinada). 6 – GEN0006 (assinada e criptografada). 108 - ZIP_CIFRADO (assina e criptografa um arquivo ZIP).	Entrada
strXMLMessage	string	Mensagem para banco central no formato XML	Entrada
lpCipherText	byte array	Envelope SPB cifrado	Saída
intCipherTextLen	Integer	Tamanho do envelope	Saída
lpLog	byte array	Log da operação	Saída
intLogLen	Integer	Tamanho do log	Saída
iCodErro	Integer	Código de erro a ser informado no caso de uma GEN0004. Zero para outras mensagens	Entrada

6.3.8 Sintaxe na linguagem C – Versão 5.0.9.0

Esta função cifra uma mensagens sem alterar a codificação da mensagem, ficando a cargo da aplicação codificar a mensagem em UTF8 ou UTF16BE.

```
__declspec( dllexport ) int __stdcall EncryptMsgV3
(
    DWORD dwProtocolVersion, // (in) Versão do protocolo (H_VERSION_CONF |
H_VERSION_2 | H_VERSION_3)
    LPCSTR lpstrDomainName, // (in) Domain Name
    LPSTR lpstrSISPB,        // (in) Source ISPB
    LPSTR lpstrDISPB,        // (in) Destination ISPB
    INT  intCodMsg,          // (in) MessageCode
    LPSTR lpstrPlainText,    // (in) XML Message
    INT  lpstrPlainTextLen,   // (in) XML Message Length
    BYTE **lpCipherText,     // (out)SPB Envelop (Liberar com
FreeMemory())
    INT *intCipherTextLen,    // (out)SPB Envelop Length
    BYTE **lpLog,            // (out)Log de auditoria do BACEN (Liberar
FreeMemory())
    INT *intLogLen,          // (out)SPB Log Length
    INT  intCodError); // (in) ErrorCode
```

6.3.9 Novos parâmetros - Versão 5.0.9.

Na versão 5.0.9 foi adicionada a opção para selecionar a versão do protocolo 2 ou 3 com cifra 3DES ou AES256. Para isso, deve-se definir a versão no parâmetro version.

Parâmetro	Tipo	Descrição	Direção
dwProtocolVersion	byte	Versão do protocolo. H_VERSION_CONF : Utiliza a versão definida na configuração do serviço. H_VERSION_2 : versão 2 do protocolo com cifra 3DES. H_VERSION_3 : versão 3 do protocolo com cifra AES256.	Entrada
strDomain	String	Nome do domínio	Entrada
strSISPB	String	ISPB da instituição de origem	Entrada
strDISPB	String	ISPB da instituição de destino	Entrada
iCodMsg	Integer	Código de Mensagens Genéricas. 0 - MSG_NORMAL - (XML assinado e criptografado). 4 – GEN0004 (somente assinada). 6 – GEN0006 (assinada e criptografada). 108 - ZIP_CIFRADO (assina e criptografa um arquivo ZIP).	Entrada
strXMLMessage	string	Mensagem para banco central no formato XML	Entrada
lpCipherText	byte array	Envelope SPB cifrado	Saída
intCipherTextLen	Integer	Tamanho do envelope	Saída
lpLog	byte array	Log da operação	Saída
intLogLen	Integer	Tamanho do log	Saída
iCodErro	Integer	Código de erro a ser informado no caso de uma GEN0004. Zero para outras mensagens	Entrada

6.3.10 Valores de retorno

O retorno do método **EncryptMsg** pode assumir os valores descritos no Anexo.

6.3.11 Parâmetro vLog

O Parâmetro **vLog** será retornado pela DLL **SPB_SECDLL** no seguinte layout:

Cabeçalho de Segurança 76 Bytes iniciais	Chave Simétrica (128 Bytes) (não cifrada)	Assinatura Digital (128 Bytes) (cifrada)	Msg XML Unicode16 (tamanho variável) (em claro)
---	--	---	--

OBS: Os campos **Cabeçalho de Segurança 76 Bytes iniciais**, **Assinatura Digital** e **Chave Simétrica** são necessários para a geração dos registros legais, obedecendo ao formato definido pelo SPB.

6.4 DecryptMsg()

Esta chamada deve ser utilizada para requisitar uma solicitação para extração do envelope digital SPB (assinatura e criptografia) para uma mensagem. Ela é utilizada pela mensageria logo após o recebimento de uma mensagem para um parceiro externo.



Atenção: Esta função cifra a mensagem conforme configuração no serviço EVALCryptoSFNMsg – Veja na seção 6.4.76.3.8, a versão 5.0.9 que permite selecionar o algoritmo de cifra.

6.4.1 Sintaxe Visual Basic

```
Private Declare Function DecryptMsg Lib "c:\winnt\system32\spb_secdll.dll"  
(  
    ByVal strDomain As String,  
    ByVal strSISPB As String,  
    ByVal strDISPB As String,  
    ByVal vEncryptTextParam As Variant,  
    ByRef vRetorno As Variant,  
    ByRef vLog As Variant,  
    ByRef iRemoteError As Integer  
) As long  
  
lngRetCode = DecryptMsg  
(  
    strDomain,  
    strSISPB,  
    strDISPB,  
    vEncryptText,  
    vPlainText,  
    vLog,  
    iRemoteError  
)
```

6.4.2 Sintaxe na Linguagem C

Esta função decifra uma mensagem alterando a codificação da mensagem para UTF16BE.

```
__declspec( dllexport ) int __stdcall DecryptMsg  
(  
    LPSTR lpstrDomain           // (in) Domain Name  
    LPSTR lpstrSISPB            // (in) Source ISPB  
    LPSTR lpstrDISPB            // (in) Destination ISPB  
    VARIANT vCipherText // (in) SPB Envelop
```

```
VARIANT *pvReturn          // (out) XML Msg
VARIANT *pvLog,             // (out)Symmetric Key, Sign, XML Msg Uni 16
INT *RemoteError           // (out) Remote Error Gen0004
)
```

6.4.3 Sintaxe na Linguagem C - versão 5.0.2

Esta função decifra uma mensagem alterando a codificação da mensagem para UTF16BE.

```
__declspec( dllexport ) int _stdcall DecryptMsgB
(
    LPCSTR lpstrDomainName,    // (in) Domain Name
    LPSTR lpstrSISPB,          // (in) Source ISPB
    LPSTR lpstrDISPB,          // (in) Destination ISPB
    BYTE *lpCipherText,        // (in) SPB Envelop
    INT intCipherTextLen,      // (in) SPB Envelop Length
    BYTE **lpReturn,           // (out)XML Message (Liberar a memória com
FreeMemory())
    INT *intReturnLen,         // (out)XML Message Length
    BYTE **lpLog,              // (out)Symmetric Key, Digital Sign, SPB Envelop +
XML Uni 16 (Liberar a memória com FreeMemory())
    INT *intLogLen,            // (out)SPB Log Lenth
    INT *RemoteError); // (out)RemoteError GEN0004
)
```

6.4.4 Sintaxe na Linguagem C - versão 5.0.7.0

Esta função decifra uma mensagens sem alterar a codificação da mensagem, ficando a cargo da aplicação codificar a mensagem em UTF8 ou UTF16BE.

```
__declspec( dllexport ) int _stdcall DecryptMsgC
(
    LPCSTR lpstrDomainName,    // (in) Domain Name
    LPSTR lpstrSISPB,          // (in) Source ISPB
    LPSTR lpstrDISPB,          // (in) Destination ISPB
    BYTE *lpCipherText,        // (in) SPB Envelop
    INT intCipherTextLen,      // (in) SPB Envelop Length
    BYTE **lpReturn,           // (out)XML Message (Liberar a memória com
FreeMemory())
    INT *intReturnLen,         // (out)XML Message Length
    BYTE **lpLog,              // (out)Log de auditoria do BACEN (Liberar a memória
com FreeMemory())
    INT *intLogLen,            // (out)SPB Log Lenth
    INT *RemoteError); // (out)RemoteError GEN0004
)
```

6.4.5 Parâmetros

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Direção</i>
<i>strDomain</i>	String	Nome do domínio	Entrada
<i>strSISPB</i>	String	ISPB da instituição de origem, segundo informações da Fila MQ	Entrada
<i>strDISPB</i>	String	ISPB da instituição de destino, segundo informações da Fila MQ	Entrada
<i>vEncryptText</i>	Byte Array	Envelope SPB enviado pelo BACEN	Entrada
<i>vPlainText</i>	variant	Mensagem descriptografada	Saída
<i>vLog</i>	variant	Log da Operação	Saída
<i>iRemoteError</i>	Integer	Retorna o erro encontrado pelo parceiro no caso de uma GEN0004	Saída

6.4.6 Novos parâmetros - versão 5.0.2 e 5.0.6

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Direção</i>
strDomain	String	Nome do domínio	Entrada
strSISPB	String	ISPB da instituição de origem, segundo informações da Fila MQ	Entrada
strDISPB	String	ISPB da instituição de destino, segundo informações da Fila MQ	Entrada
vEncryptText	Byte Array	Envelope SPB enviado pelo BACEN	Entrada
iEncryotLen	Integer	Tamanho da mensagem	Entrada
lpReturn	Byte Array	Mensagem descryptografada	Saída
intReturnLen	Integer	Tamanho da mensagem	Saída
lpLog	Byte Array	Log da Operação	Saída
intLogLen	Integer	Tamanho do Log	Saída
iRemoteError	Integer	Retorna o erro encontrado pelo parceiro no caso de uma GEN0004	Saída

6.4.7 Sintaxe na Linguagem C - versão 5.0.9.0

Esta função decifra uma mensagens sem alterar a codificação da mensagem, ficando a cargo da aplicação codificar a mensagem em UTF8 ou UTF16BE. Nesta codemos definir a versão do protocolo utilizada.

```
__declspec( dllexport ) int __stdcall DecryptMsgV3
(
    DWORD dwProtocolVersion, // (in) Versão do protocolo (H_VERSION_CONF |
H_VERSION_2 | H_VERSION_3)
    LPCSTR lpstrDomainName, // (in) Domain Name
    LPSTR lpstrSISPB, // (in) Source ISPB
    LPSTR lpstrDISPB, // (in) Destination ISPB
    BYTE *lpCipherText, // (in) SPB Envelop
    INT intCipherTextLen, // (in) SPB Envelop Length
    BYTE **lpReturn, // (out)XML Message (Liberar a memória com
FreeMemory())
    INT *intReturnLen, // (out)XML Message Length
    BYTE **lpLog, // (out)Log de auditoria do BACEN (Liberar a memória
com FreeMemory())
    INT *intLogLen, // (out)SPB Log Lenth
    INT *RemoteError); // (out)RemoteError GEN0004
)
```

6.4.8 Parâmetros

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Direção</i>
<i>dwProtocolVersion</i>	DWORD	Versão do protocolo. H_VERSION_CONF : Utilizar a versão definida na configuração do serviço. H_VERSION_V2 : Versão 2 com 3DES. H_VERSION_V3 : Versão 3 com AES256.	Entrada
<i>strDomain</i>	String	Nome do domínio	Entrada
<i>strSISPB</i>	String	ISPB da instituição de origem, segundo informações da Fila MQ	Entrada
<i>strDISPB</i>	String	ISPB da instituição de destino, segundo informações da Fila MQ	Entrada
<i>vEncryptText</i>	Byte Array	Envelope SPB enviado pelo BACEN	Entrada
<i>vPlainText</i>	Variant	Mensagem descriptografada	Saída
<i>vLog</i>	Variant	Log da Operação	Saída
<i>iRemoteError</i>	Integer	Retorna o erro encontrado pelo parceiro no caso de uma GEN0004	Saída

6.4.9 Parâmetro vlog

O Parâmetro **vLog** será retornado pela DLL **SPB_SECDLL** no seguinte layout:

Cabeçalho de Segurança 76 Bytes iniciais	Chave Simétrica (128 Bytes) (não cifrada)	Assinatura Digital (128 Bytes) (cifrada)	Mensagem XML Unicode 16 (não cifrada)
---	--	---	--

OBS: Os campos **Cabeçalho de segurança 76 bytes iniciais**, **Assinatura Digital** e **Chave Simétrica** serão utilizados para a gravação do LOG e a Mensagem XML contém uma das mensagens definidas pelo BACEN.

6.4.10 Valores de retorno

O retorno do método **DecryptMsg** pode assumir os valores descritos no Anexo.

6.5 TerminateAllConn()

Esta chamada é utilizada para o encerramento da comunicação com os EVALCryptoSFNMsg disponíveis. Ela não é obrigatória, dado que no término da utilização da DLL, a comunicação com o EVALCryptoSFNMsg é automaticamente encerrada e os recursos liberados.

6.5.1 Sintaxe Visual Basic

```
Private Declare Function TerminateAllConn Lib  
"c:\winnt\system32\spb_secdll.dll"  
(  
    ) As long  
  
lngtRetCode = TerminateAllConn()
```

6.5.2 Sintaxe na Linguagem C

```
extern "C" __declspec( dllexport ) int stdcall TerminateAllConn ()
```

6.5.3 Valores de retorno

<i>Valor de Retorno</i>	<i>Descrição</i>
0	Conexões encerradas com sucesso
1	Erro no encerramento de uma ou mais conexões

6.6 TerminateConn()

Esta chamada é utilizada para o encerramento da comunicação com um EVALCryptoSFNMsg. Ela não é obrigatória, dado que no término da utilização da DLL, a comunicação com o EVALCryptoSFNMsg é automaticamente encerrada e os recursos liberados.

6.6.1 Sintaxe Visual Basic

```
Private Declare Function TerminateConn Lib "c:\winnt\system32\spb_sec.dll"
    ( ByVal Server_Id As Integer ) As long
lngtRetCode = TerminateConn(Server_Id)
```

6.6.2 Sintaxe na Linguagem C

```
extern "C" __declspec( dllexport ) int stdcall TerminateConn (INT Server_Id)
```

6.6.3 Parâmetros

Parâmetro	Tipo	Descrição	Direção
Server_Id	Integer	Identificador do Servidor a ser desconectado. Os identificadores dos servidores seguem a ordem da lista cadastrada no Registry: 0 → Server_IP0 1 → Server_IP1 ...	Entrada

6.6.4 Valores de retorno

O retorno do método **TerminateConn** obedece aos valores da tabela abaixo:

Valor de Retorno	Descrição
0	Conexão encerrada com sucesso
1	Erro no encerramento da conexão

6.7 Reconnect()

Esta chamada deve ser utilizada quando algum servidor de segurança apresentou algum tipo de problema e teve a conexão com a DLL encerrada. Esta função percorrerá as conexões com os servidores cadastrados no Registry e para aquelas que estiverem inválidas será executada a re-conexão.

6.7.1 Sintaxe Visual Basic

A chamada executada pela Mensageria em código Visual Basic 6.0 é:

```
Private Declare Function ReConnect Lib "c:\winnt\system32\spb_sec.dll"
(
    ByRef Servers_OK As Integer
) As long

lngtRetCode = TerminateConn(Servers_OK)
```

6.7.2 Sintaxe na Linguagem C

```
extern "C" __declspec( dllexport ) int stdcall ReConnect (INT *Servers_OK)
```

6.7.3 Parâmetros

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Direção</i>
Servers_OK	Integer	Quantidade de servidores prontos para realizar operações de criptografia	Saída

6.7.4 Valores de retorno

<i>Valor de Retorno</i>	<i>Descrição</i>
0	Conexão encerrada com sucesso
1	Erro no encerramento da conexão

6.8 Status das conexões com os Servidores de Segurança

Esta chamada é utilizada para apresentar o estado atual das conexões com os servidores de segurança. Esta verificação retornará a lista de servidores disponíveis para criptografia e a lista de servidores cujas conexões foram encerradas. Uma conexão é encerrada quando o servidor não responde a uma requisição antes do valor de TIMEOUT cadastrado no Registry ou quando o serviço de segurança é desativado no servidor.

6.8.1 Sintaxe Visual Basic

```
Private Declare Function StatusConn Lib "c:\winnt\system32\spb_sec.dll"
(
    ByRef Servers_OK As Integer,
    ByVal IP_Servers_OK As String, ByRef IP_Servers_OK_LEN As Integer,
    ByRef Fault_Servers As Integer,
    ByVal IP_Fault_Servers As String, ByRef IP_Fault_Servers As Integer
) As long
```

6.8.2 Sintaxe na linguagem C

```
extern "C" __declspec( dllexport ) int _stdcall StatusConn
(
    INT *servers_ok,
    LPSTR ip_servers_ok,
    INT *ip_servers_ok_len,
    INT *fault_servers,
    LPSTR ip_fault_servers,
    INT *ip_fault_servers_len
);
```

6.8.3 Parâmetros

<i>Parâmetro</i>	<i>Tipo</i>	<i>Descrição</i>	<i>Direção</i>
Servers_OK	Integer	Quantidade de servidores prontos para realizar operações de criptografia	Saída
IP_Servers_OK	String	Endereços IP's dos servidores disponíveis para criptografia	Saída
IP_Servers_OK_Len	String	Tamanho do buffer	Entrada (255) Saída – tamanho utilizado
Fault_Servers	Integer	Quantidade de servidores com problemas	Saída
IP_Fault_Servers	String	Endereços IP's dos servidores com problemas	Saída
IP_Fault_Servers_Len	String	Tamanho do <i>buffer</i>	Entrada (255) Saída – tamanho utilizado

6.8.4 Valores de retorno

<i>Valor de Retorno</i>	<i>Descrição</i>
0	Função realizada com sucesso
1	Erro na execução da função

6.9 Servidores disponíveis

Esta chamada é utilizada para apresentar o número de servidores configurados para conexões com os servidores de segurança. Disponível a partir da versão 5.0.5.

6.9.1 Sintaxe na linguagem C

```
extern "C" __declspec( dllexport )  int _stdcall GetServerPoolSize();
```

6.9.2 Valores de retorno

<i>Valor de Retorno</i>	<i>Descrição</i>
n	Número de servidores

6.10 Conexões ativas com os Servidores de Segurança

Esta chamada é utilizada para apresentar o número de conexões ativas com os servidores de segurança. Disponível a partir da versão 5.0.5.

6.10.1 Sintaxe na linguagem C

```
extern "C" __declspec( dllexport ) int _stdcall GetConnectedServers ();
```

6.10.2 Valores de retorno

<i>Valor de Retorno</i>	<i>Descrição</i>
n	Número de servidores ativos

6.11 Libera a memória

Esta chamada é utilizada para liberar a memória das array retornadas pelas funções de cifra e decifra quando a biblioteca é utilizada com C#. Disponível a partir da versão 5.0.5.

6.11.1 Sintaxe na linguagem C

```
extern "C" __declspec( dllexport )  int _stdcall FreeMemory(BYTE *pbMalloc);
```

6.11.2 Valores de retorno

<i>Valor de Retorno</i>	<i>Descrição</i>
pbMalloc	Ponteiro da memória a ser liberada

6.12 Conversão de Variant para Char *

Esta chamada é realizada para converter uma variável do tipo VARIANT para char *. Como parâmetros desta chamada são passados os índices que contém as posições do vetor da variável VARIANT que deverão ser copiados para o CHAR *.

Antes da cópia dos dados é verificado se o buffer alocado é suficiente para receber os dados da variável do tipo VARIANT. Portanto, na chamada da função a variável **msgLen** deve conter o tamanho total do buffer apontado pelo ponteiro **msg**. No retorno da função a variável **msgLen** conterá o valor efetivamente utilizado.

6.12.1 Sintaxe na linguagem Visual Basic

```
Private Declare Function VariantToChar Lib "c:\winnt\system32\spb_sec.dll"
(
    ByVal var AsVariant,          // (in) variável do tipo VARIANT
    ByVal begin AsInteger,        // (in) índice do início do vetor
    ByVal end AsInteger,          // (in) índice final do vetor
    ByVal msg AsString,           // (in) ponteiro para buffer
    ByRef msgLen AsInteger        // (in/out) tam. do buffer
    disponível/utilizado
)
```

6.12.2 Sintaxe na linguagem C

```
__declspec( dllexport ) int __stdcall VariantToChar
(
    VARIANT var,    // (in) variável do tipo VARIANT
    INT begin,      // (in) índice do início do vetor
    INT end,        // (in) índice final do vetor
    LPSTR msg,      // (in) ponteiro para buffer
    INT *msgLen     // (in/out) tamanho do buffer disponível/utilizado
)
```

6.12.3 Valores de retorno

Valor de Retorno	Descrição
0	Conversão realizada com sucesso
-1	Erro na conversão. Deve-se verificar: tamanho do buffer alocado e índices informados.

6.13 Parâmetros de configuração da DLL SPB_SECDLL

A DLL é configurada através de parâmetros definidos no Registry do Windows. O caminho dos parâmetros varia de acordo com o modo de operação do sistema operacional, 32 ou 64 bits.

Os caminhos e parâmetros são descritos a seguir:

HKLM\SOFTWARE\Robo\SPB (SO 32 bits); ou HKLM\SOFTWARE\Wow6432Node\Robo\SPB (SO 64 bits)			
Chave	Tipo	Descrição	Valores
BUFFER_SIZE	DWORD	Tamanho máximo das mensagens SPB	0 - 65536
PROTOCOL_VERSION	DWORD	Versão do protocolo utilizado no serviço de segurança	0 – 1 0 => não efetua criptografia/descriptografia
TIMEOUT	DWORD	Tempo máximo de espera por uma resposta do serviço de segurança. Valor em milissegundos	0 – valor
HKLM\SOFTWARE\Robo\SPB\Log (SO 32 bits); ou HKLM\SOFTWARE\Wow6432Node\Robo\SPB\Log (SO 64 bits)			
Chave	Tipo	Descrição	Valores
LOG_SIZE	DWORD	Tamanho máximo dos registros de log	0- 2048
OPTION	STRING	Flag de ativação/desativação do LOG	TRUE - FALSE
LEVEL	STRING	Nível do log. INFO gera o mínimo de registros e ALL o máximo.	OFF, INFO, FATAL, ERROR, DEBUG, ALL
PATH	STRING	Path do arquivo de log da DLL	Path

HKLM\SOFTWARE\Robo\SPB\SECURITY SERVICE (SO 32 bits); ou HKLM\SOFTWARE\Wow6432Node\Robo\SPB\SECURITY SERVICE (SO 64 bits)			
Chave	Tipo	Descrição	Valores
LOAD_BALANCE	DWORD	Habilita balanceamento de carga entre os servidores de segurança	0 – Desabilitado 1- Habilitado
SECURITY_SERVERS	DWORD	Quantidade de Servidores de Segurança disponíveis	1 - 15
IP_SERVER0	STRING	Endereço IP de um servidor 0 onde está instalado o serviço de segurança	endereço
PORT_SERVER0	DWORD	Número da porta	1-65535
IP_SERVER1	STRING	Endereço IP do servidor 1 onde está instalado o serviço de segurança	ENDEREÇO
PORT_SERVER1	DWORD	Número da porta	1-65535
IP_SERVERn	STRING	Endereço IP do servidor onde está instalado o serviço de segurança	Endereço
PORT_SERVERn	DWORD	Número da porta	1-65535



As entradas IP_SERVERn e PORT_SERVERn repetem-se conforme o número de servidores cadastrados em SECURITY_SERVERS.

ANEXO A: CÓDIGOS DE RETORNO

DETECÇÃO DE ERROS NA MENSAGEM

<i>Valor de Retorno</i>	<i>Descrição</i>
-1	Erro na comunicação com o EVALCryptoSFNMsg
00	Sem erros
01	Tamanho do header de segurança zerado ou incompatível com os possíveis
02	Versão inválida ou incompatível com o tamanho e/ou conexão
03	Algoritmo da chave do destinatário inválido ou divergente do certificado
04	Algoritmo simétrico inválido
05	Algoritmo da chave de assinatura (local) inválido ou divergente do certificado
06	Algoritmo de hash não corresponde aos indicados
07	Código da AC do Certificado do destinatário inválido
08	Número de série do certificado destino inválido ou divergente do Connect
09	Código da AC do Certificado local inválido
10	Número de série do certificado local inválido ou divergente do Connect
11	Assinatura da mensagem inválida ou com erro
12	Certificado não é o do emissor da mensagem
13	Erro na extração da chave simétrica
14	Erro gerado pelo algoritmo simétrico
15	Tamanho da mensagem não é múltiplo de 8 bytes
16	Certificado usado não está ativado
17	Certificado usado está revogado, vencido ou excluído pela instituição

ERROS ESPECÍFICOS DO EVALCRYPTOSFNMSG

<i>Valor de Retorno</i>	<i>Descrição</i>
101	Erro ao extrair mensagem
103	Erro ao gerar assinatura
105	Erro ao executar o RSA
106	Tamanho do pacote inválido
107	Erro ao alocar memória
108	Erro na transmissão dos dados
111	Comando Inválido
113	Erro ao atualizar certificado proveniente de GEN0007 ou GEN0008