Расширение PKCS#11 для поддержки протокола CRISP

1. Аннотация

Данный документ определяет расширение спецификаций PKCS#11 для обеспечения возможности выполнения криптографических преобразований, описанных в спецификации Протокола защищенного обмена для индустриальных систем (CRISP) – Р 1323565.1.029—2019.

2. Список ссылок

В настоящем документе использованы ссылки на следующие стандарты и рекомендации:

- Р 1323565.1.029-2019 – “Информационная технология. КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Протокол защищенного обмена для индустриальных систем”, CRISP, Рекомендации по стандартизации, Федеральное агентство по техническому регулированию и метрологии, Стандартинформ, 2020.
- **ТК26P11RUS18** – Федеральное агентство по техническому регулированию и метрологии (РОССТАНДАРТ), Технический комитет №26, “Криптографическая защита информации”, “Информационная технология. Методические рекомендации. Расширение PKCS#11 для использования стандартов ГОСТ 34.12-2018 и ГОСТ 34.13-2018” (распространение документа ограничено).

3. Замечания

3.1 Об использовании численных идентификаторов.

До момента включения данного дополнения в официальные методические рекомендации по расширению PKCS#11 и назначения «национальных» значений для всех приведенных здесь определений численные значения для них выбираются в соответствии со следующими правилами:

1. признаком нестандартного значения (определяемого производителем) является взведенный старший бит (0x80000000);
2. в каждом из самостоятельных «пространств имен» определений значения выбираются произвольно.

4. Механизм диверсификации ключа CKM_VENDOR_KDF_CRISP_CMAC

В документе Р1323565.1.029-2019 описан алгоритм выработки производного ключа на основе алгоритма шифрования ГОСТ 34.12-2018 длиной 256 бит (Магма), работающего в режиме имитовставки в соответствии с ГОСТ 34.13-2018.

Для реализации этого алгоритма вводится механизм

`CKM_VENDOR_KDF_CRISP_CMAC`

Он используется в функции C_DeriveKey.

Механизм может применяться к ключам шифрования CKK_MAGMA.

Механизм использует один параметр, структуру, которая задает все параметры механизма.

```
typedef struct CK_VENDOR_KDF_CRISP_CMAC_PARAMS {
    CK_ULONG ulSeqNumLen;
    CK_BYTE_PTR pSeqNum;
    CK_ULONG ulSourceIdentifierLen;
    CK_BYTE_PTR pSourceIdentifier;
    CK_BYTE cs;
} CK_VENDOR_KDF_CRISP_CMAC_PARAMS;
```

`ulSeqNumLen` – байтовая длина поля pSeqNum.

`pSeqNum` – байтовая строка, параметр, задаваемый протоколом, представление целого числа SeqNum в формате Big-Endian.

`ulSourceIdentifierLen` – байтовая длина поля pSourceIdentifier.

`pSourceIdentifier` – байтовая строка, параметр, задаваемый протоколом, соответствует SourceIdentifier в описании протокола.

`cs` – идентификатор криптонабора, с возможными значениями

```
#define CRISP_CS_1 1U
#define CRISP_CS_2 2U
#define CRISP_CS_3 3U
#define CRISP_CS_4 4U
```

Результатом работы этого механизма могут быть ключ типа CKK_MAGMA, если используются идентификаторы криптонабора CRISP_CS_2 или CRISP_CS_4, или ключ типа CKK_MAGMA_TWIN_KEY, если используются идентификаторы криптонабора CRISP_CS_1 или CRISP_CS_3.

При выработке ключа типа CKK_MAGMA_TWIN_KEY при помощи данного механизма допускается установка атрибута ключа CKA_ENCRYPT в значение CK_TRUE.

5. Аутентифицированное шифрование на двойственных ключах Магма в соответствии с протоколом CRISP

В документе P1323565.1.029-2019 описан алгоритм защиты сообщения, включающий в себя шифрование и вычисление имитовставки от зашифрованных данных и дополнительных незашифрованных данных.

Для реализации этого алгоритма вводится механизм

`CKM_VENDOR_CRISP_AEAD`

Этот механизм используется в функциях C_EncryptInit и C_DecryptInit.

Использование механизма CKM_VENDOR_CRISP_AEAD соответствует использованию механизма CKM_MAGMA_MGM. Механизм в функции C_Encrypt дописывает значение имитовставки в конец зашифрованных данных. При расшифровании функцией C_Decrypt механизм проверяет корректность имитовставки перед выдачей расшифрованного текста.

Механизм использует один параметр, структуру, которая задает все параметры механизма.

```
typedef struct CK_VENDOR_CRISP_AEAD_PARAMS {
    CK_MECHANISM_PTR  pEncryptMechanism;
    CK_BYTE_BTR       pAAD;
    CK_ULONG          ulAADLen;
    CK_ULONG          ulMACLen;
} CK_VENDOR_CRISP_AEAD_PARAMS;
```

`pEncryptMechanism` – механизм шифрования данных.

`pAAD` – байтовый массив, содержащий дополнительные имитозащищаемые данные, которые в P1323565.1.029-2019 обозначены как заголовок CRISP-сообщения.

`ulAADLen` – байтовая длина поля `pAAD`.

`ulMACLen` – байтовая длина имитовставки.

В качестве ключа используется ключ типа `CKK_MAGMA_TWIN_KEY`.

В качестве механизма шифрования данных используется механизм шифрования алгоритмом Магма в режиме гаммирования `CKM_MAGMA_CTR_ACPKM` (см. **TK26P11RUS18**, раздел 13).

Для параметров механизма вводятся ограничения. Допустимое значение периода смены ключа: 0. Допустимые значения байтовой длины имитовставки: 4 и 8. Допустимая длина дополнительных имитозащищаемых данных: от 10 до 137 байт (включительно).

Приложение 1. Добавления в публичный заголовочный файл PKCS#11

```
typedef struct CK_VENDOR_KDF_CRISP_CMAC_PARAMS {
    CK_ULONG ulSeqNumLen;
    CK_BYTE_PTR pSeqNum;
    CK_ULONG ulSourceIdentifierLen;
    CK_BYTE_PTR pSourceIdentifier;
    CK_BYTE cs;
} CK_VENDOR_KDF_CRISP_CMAC_PARAMS;

typedef CK_VENDOR_KDF_CRISP_CMAC_PARAMS CK_PTR CK_VENDOR_KDF_CRISP_CMAC_PARAMS_PTR;

typedef struct CK_VENDOR_CRISP_AEAD_PARAMS {
    CK_MECHANISM_PTR  pEncryptMechanism;
    CK_BYTE_BTR       pAAD;
    CK_ULONG          ulAADLen;
    CK_ULONG          ulMACLen;
} CK_VENDOR_CRISP_AEAD_PARAMS;

typedef CK_VENDOR_CRISP_AEAD_PARAMS CK_PTR CK_VENDOR_CRISP_AEAD_PARAMS_PTR;

#define CRISP_CS_1 1U
#define CRISP_CS_2 2U
#define CRISP_CS_3 3U
#define CRISP_CS_4 4U

#define CKM_VENDOR_KDF_CRISP_CMAC   (CKM_VENDOR_DEFINED + 4UL)
#define CKM_VENDOR_CRISP_AEAD       (CKM_VENDOR_DEFINED + 5UL)
```