127 018, Москва, Сущевский Вал, д.18
Телефон: (495) 9954820
Факс: (495) 995 4820
http://www.CryptoPro.ru
E-mail: info@CryptoPro.ru

КриптоПро Hyperledger Fabric

(КриптоПро HLF)

версия 1.0

2019 г.

Оглавление

1 Аннотация

Модуль КриптоПро HLF, разработанный на базе сертифицированного СКЗИ КриптоПро CSP, обеспечивает возможность использования российских криптографических алгоритмов для реализации функций создания и проверки электронной подписи, шифрования/расшифрования данных в распределённых реестрах на основе Hyperledger Fabric версии 1.4.

При сертификации решений на основе Hyperledger Fabric с встроенным модулем КриптоПро HLF требуемые исследования будут ограничиваться проверками корректности использования в решении функций модуля КриптоПро HLF для реализации целевого функционала, а также проверками выполнения требований и рекомендаций по обеспечению информационной безопасности и защиты от несанкционированного доступа.

При этом проверки корректности реализации самих криптографических алгоритмов и работы с криптогафическими ключами в процессе выполнения решением целевых функций в данном случае не требуются.

2 Описание реализации

2.1 Основные функции

Модуль КриптоПро HLF обеспечивает:

- реализацию интерфейса BCCSP;
- поддержку российских и межгосударственных стандартов в области криптографической защиты информации:
  - ГОСТ Р 34.10-2012 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи» (ГОСТ 34.10-2018)
  - ГОСТ Р 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования» (ГОСТ 34.11-2018)
  - ГОСТ 28147-89 «Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования» (ГОСТ 34.12-2018, ГОСТ 34.13-2018)
- взаимодействие с сертифицированным ФСБ России СКЗИ КриптоПро CSP.

2.2 Состав и структура КриптоПро HLF

Модуль КриптоПро HLF содержит следующие компоненты:

- плагин Golang, реализованный в виде библиотеки **cpro.so** — предоставляет интерфейс bccsp.BCCSP;
- библиотека **libcprobccsp.so** — используется для интеграции интерфейсов Golang и CryptoAPI;
- **СКЗИ КриптоПро CSP** — реализует российские криптографические алгоритмы в соответствии с интерфейсом CryptoAPI и обеспечивает управление ключевыми элементами системы;

- патч для Hyperledger Fabric, добавляющий идентификаторы алгоритмов ГОСТ Р 34.10-2012, ГОСТ Р 34.11-2012 и т.д.

Общая структура модуля КриптоПро HLF представлена на рисунке 1.

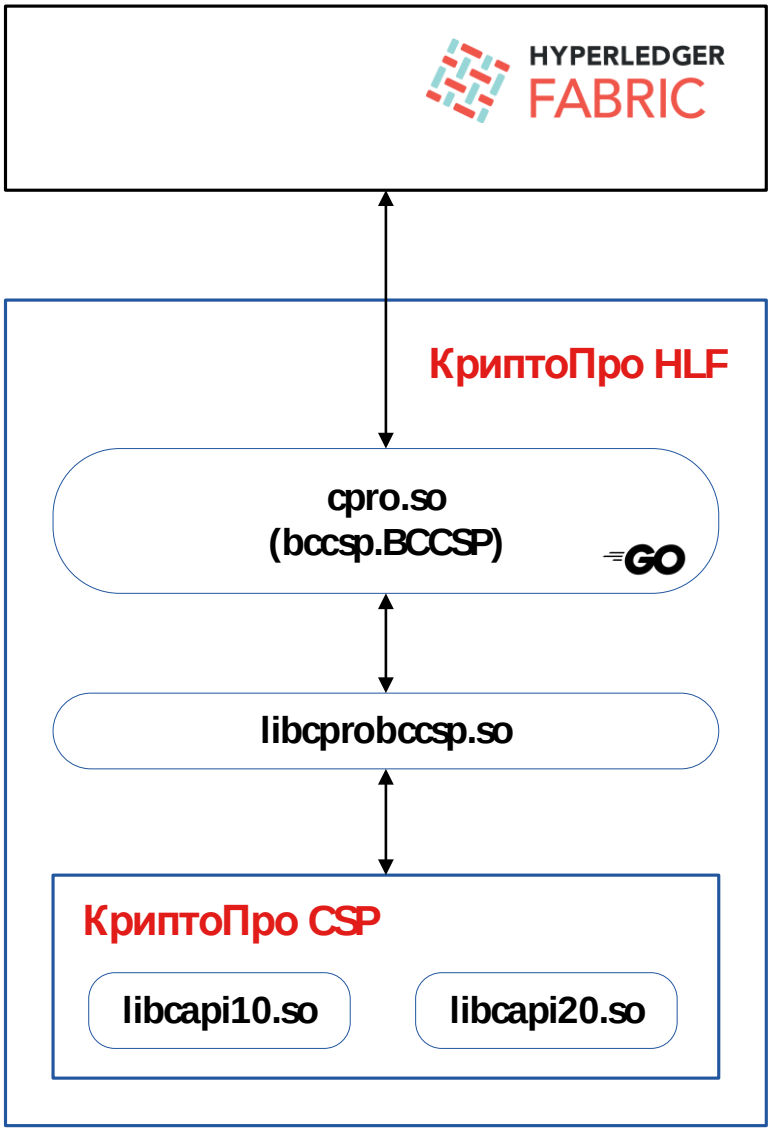

Рисунок 1 — Состав КриптоПро HLF

2.3 Операционные системы

Модуль КриптоПро HLF функционирует в ОС семейств Linux и Unix, поддерживаемых Hyperledger Fabric и СКЗИ «КриптоПро CSP» [1].

3 Использование интерфейса BCCSP

Плагин, входящий в состав КриптоПро HLF, предоставляет интерфейс bccsp.BCCSP, позволяющий использовать российские криптографические алгоритмы, реализованные в СКЗИ КриптоПро CSP.

Для возможности использования поддерживаемых российских и межгосударственных стандартов к репозиторию Hyperladger Fabric необходимо применить патч, добавляющий идентификаторы соответствующих алгоритмов.

3.1 Параметры хэширования

Определены следующие идентификаторы:

Для алгоритма ГОСТ Р 34.11-2012 с длиной хэш-кода 256 бит:

```
GOSTR3411_2012_256 = "GOSTR3411_2012_256"
```

Для алгоритма ГОСТ Р 34.11-2012 с длиной хэш-кода 512 бит:

```
GOSTR3411_2012_512 = "GOSTR3411_2012_512"
```

3.2 Параметры электронной подписи

Определены следующие идентификаторы:

Для алгоритма ГОСТ Р 34.10-2012 (256 бит):

```
GOSTR3410_2012_256 = "GOSTR3410_2012_256"
```

Для алгоритма ГОСТ Р 34.10-2012 (512 бит):

```
GOSTR3410_2012_512 = "GOSTR3410_2012_512"
```

4 Установка и настройка КриптоПро HLF

4.1 Установка КриптоПро CSP

Модуль КриптоПро HLF обеспечивает только реализацию программного интерфейса и должен работать совместно с СКЗИ «КриптоПро CSP», средствами которого реализуются криптографические алгоритмы и работа с ключевой информацией.

Поэтому для корректной работы модуля КриптоПро HLF прежде всего необходимо установить и настроить СКЗИ «КриптоПро CSP» в соответствии с эксплуатационной документацией на него, сгенерировать или экспортировать ключевую информацию.

Инструкции по установке СКЗИ «КриптоПро CSP» описаны в разделе «Установка дистрибутива ПО СКЗИ» в руководстве администратора безопасности, соответствующем используемой программно-аппаратной платформе [2, 3].

4.2 Установка КриптоПро HLF

Порядок интеграции модуля КриптоПро HLF в fabric-peer:

1. применить патч к репозиторию hyperledger/fabric
2. собрать peer (make peer)
3. собрать новый Docker-образ на основе оригинального fabric-peer:

```
FROM hyperledger/fabric-peer:1.4.0


RUN /bin/bash -c 'apt-get update && apt-get install -y vim
net-tools git gcc libltdl-dev'


RUN /bin/bash -c 'mkdir -p /etc/hyperledger/fabric/plugin'
COPY cpro.so /etc/hyperledger/fabric/plugin/cpro.so
```

```
COPY linux-amd64_deb /root/linux-amd64_deb
RUN /bin/bash -c 'cd /root/linux-amd64_deb && ./install.sh'


COPY libcprobccsp.so /opt/cprocsp/lib/amd64/libcprobccsp.so
RUN /bin/bash -c 'ldconfig'


# peer, собранный в предыдущем пункте
COPY peer /usr/local/bin/peer
```

4. переключить в конфигурации peer (/usr/hyperledger/fabric/core.yaml) провайдер:

```
BCCSP:
   Default: PLUGIN
   PLUGIN:
         Library: /etc/hyperledger/fabric/plugin/cpro.so
         Config:
             FileKeyStore:
             KeyStore: /etc/hyperledger/fabric/msp/keystore
```

5 Литература

1. ЖТЯИ.00087-03 30 01. КриптоПро CSP. Формуляр

2. ЖТЯИ.00087-03 91 03. КриптоПро CSP. Руководство администратора безопасности. Linux

3. ЖТЯИ.00087-03 91 07. КриптоПро CSP. Руководство администратора безопасности. Mac OS X

Приложение А. Интерфейс BCCSP

А.1 Интерфейс ключа

```
type Key interface{
```

`Bytes() ([]byte, error)`	возвращает байтовое представление ключа (если операция разрешена)
`GetSKI() []byte`	возвращает идентификатор ключа (SKI)
`Symmetric() (bool)`	возвращает *true*, если ключ симметричный, иначе – *false*
`Private() (bool)`	возвращает *true*, если ключ закрытый, иначе – *false*
`PublicKey() (Key, error)`	возвращает открытый ключ, соответствующий указанной ключевой паре; возвращает ошибку в случае симметричных алгоритмов

```
}
```

А.2 Параметры функций

`type KeyGenOpts interface{`	Параметры генерации ключа
`Algorithm() string`	возвращает идентификатор алгоритма генерации ключа

Ephemeral() bool	возвращает *true*, если генерируемый ключ должен быть эфемерным, иначе – *false*

}

type KeyDerivOpts interface{	Параметры формирования ключа
Algorithm() string	возвращает идентификатор алгоритма формирования ключа
Ephemeral() bool	возвращает *true*, если генерируемый ключ должен быть эфемерным, иначе – *false*

}

type KeyImportOpts interface{	Параметры импорта ключа
Algorithm() string	возвращает идентификатор алгоритма импорта ключа
Ephemeral() bool	возвращает *true*, если генерируемый ключ должен быть эфемерным, иначе – *false*

}

type HashOpts interface{	Параметры хеширования
Algorithm() string	возвращает идентификатор алгоритма хэширования

}

type SignerOpts interface{}	Параметры подписи
type EncrypterOpts interface{}	Параметры зашифрования
type DecrypterOpts interface{}	Параметры расшифрования

A.3 Функции работы с ключами

KeyGen(opts KeyGenOpts) (k Key, err error)

Генерирует ключ с заданными параметрами

Аргументы

opts	Параметры генерируемого ключа

Возвращаемые значения

k	Сгенерированный ключ
err	Код ошибки (при успешном завершении функции равен 0)

KeyDeriv(k Key, opts KeyDerivOpts) (dk Key, err error)

Формирует ключ с заданными параметрами из базового ключа

Примечание: Аргумент opts должен соответствовать используемому примитиву.

Аргументы

k	Базовый ключ
opts	Параметры формируемого ключа

Возвращаемые значения

dk	Сформированный ключ
err	Код ошибки (при успешном завершении функции равен 0)

KeyImport(raw interface{}, opts KeyImportOpts) (k Key, err error)

Импортирует ключ из необработанного представления (из ключевого блоба?) с заданными параметрами

Примечание: Аргумент opts должен соответствовать используемому примитиву.

Аргументы

raw	Необработанный ключ
opts	Параметры импортируемого ключа

Возвращаемые значения

k	Импортированный ключа
err	Код ошибки (при успешном завершении функции равен 0)

GetKey(ski []byte) (k Key, err error)

Возвращает ключ, который CSP ассоциирует с указанным идентификатором ключа субъекта (SKI).

Аргументы

ski	Идентификатор ключа субъекта

Возвращаемые значения

k	Ключ

err	Код ошибки (при успешном завершении функции равен 0)

A.4 Функции хэширования данных

Hash(msg []byte, opts HashOpts) (hash []byte, err error)

Вычисляет хэш-код сообщения с заданными параметрами

Примечание: Если параметры не заданы, используется хэш-функция по умолчанию

Аргументы

msg	Хэшируемое сообщение
opts	Параметры хэширования

Возвращаемые значения

hash	Хэш-код сообщения
err	Код ошибки (при успешном завершении функции равен 0)

GetHash(opts HashOpts) (h hash.Hash, err error)

Возвращает экземпляр hash.Hash с заданными параметрами

Примечание: Если параметры не заданы, используется хэш-функция по умолчанию

Аргументы

opts	Параметры хэширования

Возвращаемые значения

h	Экземпляр hash.Hash

err	Код ошибки (при успешном завершении функции равен 0)

A.5 Функции создания и проверки ЭП

Sign(k Key, digest []byte, opts SignerOpts) (signature []byte, err error)

Подписывает хэш-код сообщения, используя заданные ключ и параметры

Примечание: Алгоритм, используемый для подписи, определяется ключом.

Если требуется подпись хэш-кода сообщения большего размера, вызывающая сторона отвечает за хэширование сообщения и передачу хэш-кода в качестве digest.

Аргументы

k	Ключ
digest	Хэш-код сообщения
opts	Параметры подписи

Возвращаемые значения

signature	Значение подписи
err	Код ошибки (при успешном завершении функции равен 0)

Verify(k Key, signature, digest []byte, opts SignerOpts) (valid bool, err error)

Осуществляет проверку подписи хэш-кода сообщения, используя заданные ключ и параметры

Примечание:	Аргумент opts должен соответствовать используемому алгоритму.

Аргументы

k	Ключ
signature	Значение проверяемой подписи
opts	Параметры подписи

Возвращаемые значения

valid	Результат проверки подписи
err	Код ошибки (при успешном завершении функции равен 0)

A.6 Функции шифрования/расшифрования

Encrypt(k Key, plaintext []byte, opts EncrypterOpts) (ciphertext []byte, err error)

	Зашифровывает отрытый текст, используя заданные ключ и параметры
Примечание:	Алгоритм, используемый для зашифрования, определяется ключом и параметрами.
	Аргумент opts должен соответствовать используемому примитиву.

Аргументы

k	Ключ
plaintext	Открытый текст
opts	Параметры шифрования

Возвращаемые значения

ciphertext	Шифртекст

err	Код ошибки (при успешном завершении функции равен 0)

Decrypt(k Key, ciphertext []byte, opts DecrypterOpts) (plaintext []byte, err error)

Расшифровывает данные, предварительно зашифрованные функцией Encrypt

Примечание: Алгоритм, используемый для расшифрования, определяется ключом и параметрами.

Аргумент opts должен соответствовать используемому примитиву.

Аргументы

k	Ключ
ciphertext	Шифртекст
opts	Параметры шифрования

Возвращаемые значения

plaintext	Открытый текст
err	Код ошибки (при успешном завершении функции равен 0)