Утверждены решением заседания
технического комитета по стандартизации
«Криптографическая защита информации»
(Протокол № 10 от 27.11.2012 г.)

Методические рекомендации

технического комитета по стандартизации
«Криптографическая защита информации»
(ТК 26)

Парольная защита с использованием алгоритмов ГОСТ

Дополнения к PKCS#5

Версия 1.0

Оглавление

1. Общие положения

Данный документ описывает расширения стандарта PKCS#5 (RFC 2898) для использования алгоритмов ГОСТ Р 34.11–94 [4] и ГОСТ 28147–89 [5] при выработке ключей из парольной информации и защиты ключей с использованием паролей.

2. Алгоритм выработки ключа из пароля

В соответствии с рекомендациями PKCS#5 для выработки ключа из пароля следует использовать функцию диверсификации PBKDF2 (P, S, c, dkLen) с использованием в качестве PRF функции HMAC_GOSTR3411 в соответствии с RFC 4357 [2].

DK = PBKDF2 (P, S, c, dkLen)

Функции:

HMAC_GOSTR3411 (K, text) – функция выработки HMAC ГОСТ Р 34.11–94 на ключе K от данных text (см. RFC 4357 [2], раздел 3).

Параметры:

P – пароль (символьная строка в кодировке Unicode UTF-8).
S - случайное значение salt [1].
c - число итераций алгоритма.
dkLen - требуемая размерность выходной последовательности в байтах.
hLen – размерность выхода псевдослучайной функции в байтах.

Выход:

DK, производный ключ длины dkLen байт.

Алгоритм:

1) Если dkLen > $(2^{32} - 1) \times 32$ алгоритм завершает работу с ошибкой (неверные параметры).
2) n = $\lceil dkLen/hLen \rceil$
3) T_1 = F (P, S, c, 1)
   T_2 = F (P, S, c, 2) ,
   …,
   T_n = F (P, S, c, n) , где

   F (P, S, c, i) = $U_1 \oplus U_2 \oplus \ldots \oplus U_c$, где $\oplus$ - поразрядное сложение по модулю 2.

   U_1 = HMAC_GOSTR3411 (P, S || Int(i)) , где Int(i) четырехбайтовое представление целого числа i, старший байт слева.
   U_2 = HMAC_GOSTR3411 (P, U_1),
   …
   U_c = HMAC_GOSTR3411 (P, U_{c-1}).

4) Ключ DK получается конкатенацией $\{T_i\}$
   DK = $T_1 \| T_2 \| \cdots \| T_n$
   и обрезанием T_n до требуемой длины dkLen выходной последовательности.

3. Шифрование данных

Шифрование данных при использовании ключа DK должно осуществляться в соответствии со схемой PBES2 с использованием ГОСТ 28147–89 [5] в режиме гаммирования с обратной связью.

Процесс шифрования в данной схеме выглядит следующим образом:

1) Выбирается случайное значение salt S размерности от 8 до 32 байт. Рекомендуемая размерность 32 байта.
2) Число итераций c выбирается в зависимости от условий применения. Минимально допустимое значение параметра 1000, рекомендуемое 2000.
3) Полагаем dkLen = 32.
4) Производится выработка последовательности DK = PBKDF2 (P, S, c, 32) .
5) Ключ шифрования данных выбирается как первые 32 байта выходной последовательности DK, CK = T_1.
6) Осуществляется шифрование данных по алгоритму ГОСТ 28147–89 [5] на ключе CK в режиме гаммирования с обратной связью со случайной синхропосылкой S’.
7) Параметры S, c, S’ и шифртекст C сохраняются в выходной структуре для последующего расшифрования.

Расшифрование данных осуществляется аналогичным образом с использованием параметров S, c и S’, использовавшихся при зашифровании.

4. Контроль целостности

Для вычисления контрольной суммы передаваемых данных следует использовать схему вычисления PBMAC1 с использованием функции HMAC_GOSTR3411 на ключе

DK = PBKDF2 (P, S, c, 32).

5. Идентификация и параметры

```
rsadsi OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840) 113549}
pkcs OBJECT IDENTIFIER ::= {rsadsi 1}
pkcs-5 OBJECT IDENTIFIER ::= {pkcs 5}
```

5.1. PBKDF2

Объектный идентификатор для схемы PBKDF2 в соответствии с PKCS#5:

```
id-PBKDF2 OBJECT IDENTIFIER ::= {pkcs-5 12}
```

Параметры алгоритма должны быть представлены в следующем виде:

```
PBKDF2-params ::= SEQUENCE {
        salt CHOICE {
                specified OCTET STRING,
                otherSource AlgorithmIdentifier {{PBKDF2-SaltSources}}
        },
        iterationCount INTEGER (1..MAX),
        keyLength INTEGER (1..MAX) OPTIONAL,
        prf AlgorithmIdentifier {{PBKDF2-PRFs}}
}
```

salt - значение случайной синхропосылки размерности от 8 до 32 байт, представленное в виде **OCTET STRING**.

iterationCount - число итераций выбирается в зависимости от условий применения. Минимально допустимое значение параметра 1000, рекомендуемое 2000.

keyLength - размер ключа в байтах. В рамках схемы PBES2 должен отсутствовать, так как всегда равен 32. В рамках схемы PBMAC1 должен быть равен 32, но должен всегда присутствовать, так как функция HMAC_GOSTR3411 имеет переменный размер ключа.

prf - идентификатор алгоритма, используемого в качестве PRF, должен быть указан идентификатор алгоритма HMAC_GOSTR3411, в соответствии с RFC 4490 [3]:

id-HMACGostR3411-94 OBJECT IDENTIFIER ::= {iso(1) member-body(2) ru(643) rans(2) cryptopro(2) hmacgostr3411(10) }

Алгоритм ГОСТ Р 34.11-94 [4] при вычислении HMAC должен использоваться только с параметрами **id-GostR3411-94-CryptoProParamSet**, параметры HMAC_GOSTR3411 не указываются:

prf.parameters = NULL.

5.2. PBES2

Идентификатор данной схемы в соответствии с PKCS#5:

id-PBES2 OBJECT IDENTIFIER ::= {pkcs-5 13}

Параметры алгоритма:

```
PBES2-params ::= SEQUENCE {
        keyDerivationFunc AlgorithmIdentifier {{PBES2-KDFs}},
        encryptionScheme AlgorithmIdentifier {{PBES2-Encs}}
}
```

keyDerivationFunc - идентификатор и параметры алгоритма выработки парольного ключа PBKDF2 с параметрами в соответствии с предыдущим пунктом.

encryptionScheme - идентификатор и параметры алгоритма шифрования ГОСТ 28147–89 [5] в соответствии с RFC 4490 [3].

id-Gost28147-89 OBJECT IDENTIFIER ::= { iso(1) member-body(2) ru(643) rans(2) cryptopro(2) gost28147-89(21) }

Параметры алгоритма:

```
Gost28147-89-Parameters ::= SEQUENCE {
        iv          Gost28147-89-IV,
        encryptionParamSet   OBJECT IDENTIFIER
}

Gost28147-89-IV ::= OCTET STRING (SIZE (8))
```

5.3. PBMAC1

Объектный идентификатор для схемы PBKDF2 в соответствии с PKCS#5:

id-PBMAC1 OBJECT IDENTIFIER ::= {pkcs-5 14}

Параметры должны быть представлены в следующем виде:

```
PBMAC1-params ::= SEQUENCE {
        keyDerivationFunc AlgorithmIdentifier {{PBMAC1-KDFs}},
        messageAuthScheme AlgorithmIdentifier {{PBMAC1-MACs}}
}
```

keyDerivationFunc - идентификатор и параметры алгоритма выработки парольного ключа в соответствии с разделом 5.1. PBKDF2.

messageAuthScheme - Идентификатор алгоритма HMAC_GOSTR3411 в соответствии с RFC 4490 [3].

5.4. Рекомендуемые криптографические параметры

При практической реализации парольной защиты с использованием алгоритмов ГОСТ рекомендуется использовать следующие параметры: id-Gost28147-89-CryptoPro-A-ParamSet для шифрования данных в рамках схемы PBES2 с использованием алгоритма ГОСТ 28147–89 [5] и id-GostR3411-94-CryptoProParamSet для контроля целостности в рамках схемы PBMAC1 с использованием алгоритма ГОСТ Р34.11–94 [4].

6. Тестовые примеры

Данные тестовые вектора сформированы по аналогии с тестовыми векторами из RFC 6070 [6] для алгоритма хэширования ГОСТ Р34.11–94 [4].

```
Input:
  P = "password" (8 octets)
  S = "salt" (4 octets)
  c = 1
  dkLen = 32

Output:
   DK =    73 14 e7 c0 4f b2 e6 62 c5 43 67 42 53 f6 8b d0
           b7 34 45 d0 7f 24 1b ed 87 28 82 da 21 66 2d 58             (32 octets)

Input:
  P = "password" (8 octets)
  S = "salt" (4 octets)
  c = 2
  dkLen = 32

Output:
```

```
   DK =     99 0d fa 2b d9 65 63 9b a4 8b 07 b7 92 77 5d f7
            9f 2d b3 4f ef 25 f2 74 37 88 72 fe d7 ed 1b b3          (32 octets)
```

```
Input:
  P = "password" (8 octets)
  S = "salt" (4 octets)
  c = 4096
  dkLen = 32

Output:
 DK =  1f 18 29 a9 4b df f5 be 10 d0 ae b3 6a f4 98 e7
       a9 74 67 f3 b3 11 16 a5 a7 c1 af ff 9d ea da fe        (32 octets)
```

```
Input:
  P = "password" (8 octets)
  S = "salt" (4 octets)
  c = 16777216
  dkLen = 32

Output:
 DK =  a5 7a e5 a6 08 83 96 d1 20 85 0c 5c 09 de 0a 52
       51 00 93 8a 59 b1 b5 c3 f7 81 09 10 d0 5f cd 97        (32 octets)
```

```
Input:
  P = "passwordPASSWORDpassword" (24 octets)
  S = "saltSALTsaltSALTsaltSALTsaltSALTsalt" (36 octets)
  c = 4096
  dkLen = 40

Output:
 DK =  78 83 58 c6 9c b2 db e2 51 a7 bb 17 d5 f4 24 1f
       26 5a 79 2a 35 be cd e8 d5 6f 32 6b 49 c8 50 47
       b7 63 8a cb 47 64 b1 fd                                  (40 octets)
```

```
Input:
  P = "pass\0word" (9 octets)
  S = "sa\0lt" (5 octets)
  c = 4096
  dkLen = 20

Output:
 DK =  43 e0 6c 55 90 b0 8c 02 25 24 23 73 12 7e df 9c
       8e 9c 32 91                                              (20 octets)
```

Литература

1. PKCS #5 v2.1: Password-Based Cryptography Standard RSA Laboratories October 5, 2006.
2. RFC 4357, Additional Cryptographic Algorithms for Use with GOST 28147–89, GOST R 34.10–94, GOST R 34.10–2001, and GOST R 34.11–94 Algorithms.
3. RFC 4490, Using the GOST 28147-89, GOST R 34.11–94, GOST R 34.10–94, and GOST R 34.10–2001 Algorithms with Cryptographic Message Syntax (CMS).
4. ГОСТ Р 34.11–94 Информационная технология. Криптографическая защита информации. Функция хэширования
5. ГОСТ 28147–89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования
6. RFC 6070. PKCS #5: Password-Based Key Derivation Function 2 (PBKDF2). Test Vectors.