Алгоритмы шифрования с открытым ключом: Протокол Крамера-Шупа

МФТИ (ГУ), ФИВТ
artikov.akmalzhon@phystech.edu

Артыков Акмал

Март 2015

1 Вступление

Протокол Крамера-Шупа - это алгоритм шифрования с открытым ключом, придуманный в 1998 Рональдом Крамером и Виктором Шаупом как расширение уже существующей на тот момент схемы Эль-Гамаля. Особенностью этого протокола является то, что он устойчив к атакам на основе адаптивно подобранного шифротекста. Безопасность протокола основывается на предположении Диффи-Хеллмана о различении *(DDH Assumption)*. Кроме того, протокол Крамера-Шупа обладает свойством неподатливости (*non-malleability* - взломщик не может поменять шифротекст на другой шифротекст, который расшифровывается в текст, связанный с исходным, т.е. является какой-то функцией от него), в отличии той же схемы Эль-Гамаля. Считается, что протокол Крамера-Шупа - первый алгоритм шифрования с открытым ключом, удовлетворяющий этим свойствам.

2 Необходимые определения

2.1 Криптосистема с открытым ключом

Определение 2.1. *Криптографическая система с открытым ключом* — система шифрования или электронной подписи, при которой открытый ключ передаётся по открытому (то есть незащищённому, доступному для наблюдения) каналу и используется для проверки электронной подписи и для шифрования сообщения. Для генерации электронной подписи и для расшифровки сообщения используется закрытый ключ. Криптографические системы с открытым ключом в настоящее время широко применяются в

различных сетевых протоколах, в частности, в протоколах TLS и его предшественнике SSL (лежащих в основе HTTPS), в SSH.

2.2 Устойчивость к атакам на основе адаптивно подобранного шифротекста

Определение 2.2. *Атака на основе подобранного шифротекста* — криптографическая атака, при которой криптоаналитик собирает информацию о шифре путем подбора зашифрованного текста и получения его расшифровки при неизвестном ключе. Криптоаналитик может воспользоваться устройством расшифрования несколько раз для получения шифротекста в расшифрованном виде. Используя полученные данные, он может попытаться восстановить секретный ключ для расшифровки. Атака на основе подобранного шифротекста может быть адаптивной и неадаптивной.

Определение 2.3. При *неадаптивной атаке* криптоаналитик не использует результаты предыдущих расшифровок, то есть шифротексты подбираются заранее. Такие атаки называют атаками в обеденное время (*lunchtime* или *CCA1*).

Определение 2.4. В случае *адаптивной атаки* криптоаналитик адаптивно подбирает шифротекст, который зависит от результатов предыдущих расшифровок (*CCA2*).

Рассмотрим устойчивость к адаптивной атаке на основе подобранного шифротекста с помощью соответствующей игры, проводимой противником. Следует помнить, что у противника есть доступ к оракулу дешифрования:

1. Запускается алгоритм генерации ключа в схеме шифрования с соответствующей длиной ключа, подаваемой на вход.

2. Противник выполняет серию произвольных запросов к оракулу дешифрования, таким образом дешифровывая шифротексты по его выбору.

3. Противник выбирает 2 сообщения m_0, m_1 и отправляет их к оракулу шифрования.

4. Оракул шифрования случайно выбирает бит $b \in \{0, 1\}$, затем шифрует m_b, который передается противнику (подбрасывание монетки для выбора бита b скрыто от противника).

5. Противник снова выполняет серию произвольных запросов к оракулу дешифрования с одним лишь ограничением, что запрос должен отличаться от сообщения, полученного им от оракула шифрования.

6. Противник выдает бит $b' \in \{0, 1\}$ - предполагаемое значение бита b, выбранного оракулом шифрования на шаге *4*. Если $P_r(b' = b) \geq \frac{1}{2} + \epsilon$, то *превосходство* противника считается равным ϵ.

Определение 2.5. Криптосистема *безопасна к атак на основе адаптивно подобранного шифротекста*, если для любого противника, работающего полиномиальное количество времени, любого положительного полинома $p(\cdot)$ $\exists N\ \forall n \geq N$ превосходство противника меньше, чем $\frac{1}{p(n)}$.

2.3 Задача Диффи-Хеллмана о различении

Существует несколько эквивалентных формулировок задачи Диффи-Хеллмана о различении *(The Diffie-Hellman Decision Problem)*. Та, которую мы будем использовать, выглядит следующим образом:

Определение 2.6. Пусть G - группа порядка q, где q - большое простое число. Рассмотрим 2 распределения:

- Распределение R случайных четверок $(g_1, g_2, u_1, u_2) \in G^4$ (выбираются случайно и равномерно).
- Распределение D четверок $(g_1, g_2, u_1, u_2) \in G^4$, где g_1, g_2 - случайны, а $u_1 = g_1^r$ и $u_2 = g_2^r$ для случайного $r \in Z_q$.

Алгоритмом, решающим задачу Диффи-Хеллмана о различении, называется такой вероятностный алгоритм A, который может эффективно различать эти распределения. А принимает на вход x - одно из этих 2 распределений, и выдает 0 или 1. Тогда существует положительный полином $p(\cdot)$ $\forall N\ \exists n \geq N$, что:

$$|P_r(A(x) = 1 \mid x \in R) - P_r(A(x) = 1 \mid x \in D)| \geq \frac{1}{p(n)} \tag{1}$$

Задача Диффи-Хеллмана о различении трудна, если такого полиномиального вероятностного алгоритма не существует.

Определение 2.7. Другое определение, эквивалентное показанному выше состоит в следующем. Положим следующие замены в терминах предыдущего определения: $g_1 \to g,\ g_2 \to g^x,\ u_1 \to g^y,\ u_2 \to g^{xy}$. Таким образом, мы сводимся к задаче умения различать тройки Диффи-Хеллмана $(g^x,\ g^y,\ g^{xy})$ от обычных случайных троек $(g^x,\ g^y,\ g^z)$.

Предположение, что данная задача трудна (то есть, любой вероятностный полиномиальный алгоритм решает эту задачу с вероятностью меньше обратного полинома), используется в доказательстве безопасности многих криптографических протоколов, в том числе, и протокола Крамера-Шупа.

2.4 Семейство хэш-функций, устойчивых относительно коллизий

Обозначим S_n^m множество строк, которые представляют функции, переводящие n-битную строку в m-битную. Для простоты, будем считать, что

$S_n^m = \{0,1\}^{l(n,m)}$ для некоторой функции l. Кроме того, мы будем ассоциировать строки из S_n^m с функциями, которые они представляют. Обозначим H_n^m случайную величину, равномерно распределенную на S_n^m.

Определение 2.8. Будем называть S_n^m *семейством хэш-функций*, если выполнены следующие условия:

1. S_n^m – попарно независимое семейство отображений: для всех $x \neq y \in \{0,1\}^n$, случайные величины $H_n^m(x)$ и $H_n^m(y)$ независимы и равномерно распределены на $\{0,1\}^m$.
2. S_n^m имеет краткое представление: $S_n^m = \{0,1\}^{poly(n,m)}$.
3. S_n^m может быть эффективно посчитана: существует полиномиальный алгоритм, принимающий на вход представление функции h (из S_n^m) и x, и возвращающий $h(x)$.

Определение 2.9. Семейство хэш-функций называется *устойчивым относительно коллизий*, если полиномиальный противник для случайной функции H из семейства не сможет (сможет с пренебрежимо маленькой вероятностью, меньшей, чем обратный полином) найти такие x и y, что $x \neq y$ и $H(x) = H(y)$.

Определение 2.10. Более слабое определение - *универсальное семейство односторонних хеш-функций*. В этом случае полиномиальный противник выбирает x, затем для случайной функции H из семейства пытается найти такой y, что $x \neq y$ и $H(x) = H(y)$, причем сделать это ему не удастся (удастся с пренебрежимо маленькой вероятностью, меньшей, чем обратный полином).

3 Протокол

Пусть у нас есть группа G порядка q, где q - большое простое число. Сообщения - это элементы из G. Также мы используем универсальное семейство односторонних хэш-функций, которое отображает длинные битовые строчки в элементы Z_q.

3.1 Генерация ключа

Алгоритм генерации ключей работает следующим образом:

1. Выбираются случайные элементы $g_1, g_2 \in G$ и случайные элементы $x_1, x_2, y_1, y_2, z \in Z_q$.
2. Затем вычисляются следующие значения.

$$c = g_1^{x_1} g_2^{x_2},\ d = g_1^{y_1} g_2^{y_2},\ h = g_1^z \tag{2}$$

3. Выбирается хэш-функция H из универсального семейства односторонних хеш-функций. Публичный ключ - (g_1, g_2, c, d, h, H), скрытый ключ - (x_1, x_2, y_1, y_2, z).

3.2 Шифрование

Дано сообщение $m \in G$. Алгоритм шифрования работает следующим образом:

1. Случайно выбирает $r \in Z_q$.
2. Вычисляются следующие значения.

$$u_1 = g_1^r,\ u_2 = g_2^r,\ e = h^r m,\ \alpha = H(u_1, u_2, e),\ v = c^r d^{r\alpha} \tag{3}$$

3. Отправляется зашифрованный текст (u_1, u_2, e, v).

3.3 Дешифрование

Получив зашифрованный текст (u_1, u_2, e, v), алгоритм дешифрования работает следующим образом:

1. Вычисляется $\alpha = H(u_1, u_2, e)$.
2. Проверяется следующее условие:

$$u_1^{x_1 + y_1\alpha} u_2^{x_2 + y_2\alpha} = v \tag{4}$$

Если условие не выполняется, то протокол завершается с отказом дешифрования. Иначе, выводится сообщение

$$m = \frac{e}{u_1^z} \tag{5}$$

4 Корректность протокола

Проверим корректность шифровальной схемы (расшифровка зашифрованного сообщения выдает это самое сообщение). Учитывая, что $u_1 = g_1^r$ и $u_2 = g_2^r$, имеем

$$u_1^{x_1} u_2^{x_2} = g_1^{rx_1} g_2^{rx_2} = c^r \tag{6}$$

Также, $u_1^{y_1} u_2^{y_2} = d^r$ и $u_1^z = h^r$. Поэтому проверка дешифровщика на шаге 2 проходит успешно и выводится исходное сообщение $\frac{e}{h^r} = m$.

5 Безопасность протокола

Теорема 5.1. *Описанная в разделе 3, криптосистема устойчива к атакам на основе адаптивно подобранного шифротекста при выполнении следующих условий.*

1. *Хэш-функция H выбирается из универсального семейства односторонних хэш-функций.*

2. *Задача Диффи-Хеллмана о различении трудна для группы G.*

Доказательство. Чтобы доказать теорему, мы предположим, что существует противник, который может взломать протокол, и покажем, что при выполнении условия *(1)*, получается противоречие с условием *(2)* (построим вероятностный полиномиальный алгоритм для решения задачи Диффи-Хеллмана о различении).

На вход нашему вероятностному алгоритму подается (g_1, g_2, u_1, u_2) из распределения R или D. Мы построим симулятор, который будет выдавать совместное распределение, состоящее из видения взломщиком криптосистемы после серии атак и скрытого бита b, генерируемым оракулом генерации (не входит в видение взломщика, скрыто от него).

Идея доказательства: мы покажем, что если на вход подается распределение из D, то симуляция пройдет успешно, а противник будет иметь нетривиальное превосходство в угадывании случайного бита b. Также, мы покажем, что если на вход подается распределение из R, то видение противника не зависит от b, а, значит, превосходство противника будет ничтожно мало (меньше обратного полинома). Отсюда можно построить различитель распределений R и D: запускаем симулятор криптосистемы (выводит b) и взломщика (выводит b') одновременно и выдаем 1, если $b = b'$ и 0, иначе.

Теперь займемся конструированием симулятора.

Симуляция генерации ключа:

1. На вход алгоритму поступает (g_1, g_2, u_1, u_2).

2. Выбираются случайные элементы $x_1, x_2, y_1, y_2, z_1, z_2 \in Z_q$.

3. Вычисляются следующие величины:

$$c = g_1^{x_1} g_2^{x_2},\ d = g_1^{y_1} g_2^{y_2},\ h = g_1^{z_1} g_2^{z_2} \tag{7}$$

4. Симулятор выбирает случайную хэш-функцию H из семейства и выдает публичный ключ (g_1, g_2, c, d, h, H). Скрытый ключ симулятора: $(x_1, x_2, y_1, y_2, z_1, z_2)$.

Можно заметить, что генерация ключа симулятора отличается от генерации ключа в протоколе (там $z_2 = 0$).

Симуляция дешифрования: Происходит так же, как и в протоколе, кроме того, что $m = \frac{e}{u_1^{z_1} u_2^{z_2}}$.

Симуляция шифрования: Получая на вход $m_0,\ m_1$, симулятор выбирает случайный $b \in \{0, 1\}$, вычисляет

$$e = u_1^{z_1} u_2^{z_2} m_b,\ \alpha = H(u_1, u_2, e),\ v = u_1^{x_1 + y_1 \alpha} u_2^{x_2 + y_2 \alpha} \tag{8}$$

и выводит (u_1, u_2, e, v).

Теперь доказательство теоремы будет следовать из следующих 2 лемм.

□

Лемма 5.2. *Если на вход симулятору подается распределение из D, то совместное распределение видения взломщиком криптосистемы и скрытого бита b статистически неразличимо от настоящей атаки криптосистемы.*

Доказательство. На вход симулятору подается распределение из D, поэтому положим $u_1 = g_1^r$ и $u_2 = g_2^r$. Очевидно, что в таком случае распределение шифровального оракула будет верным, поскольку $u_1^{x_1} u_2^{x_2} = c^r,\ u_1^{y_1} u_2^{y_2} = d^r,\ u_1^{z_1} u_2^{z_2} = h^r$. В самом деле, эти уравнения подразумевают, что $e = m_b h^r$ и $v = c^r d^{r\alpha}$, где α уже в правильной форме.

Чтобы завершить доказательство, нужно доказать, что выход дешифровального оракула тоже имеет правильное распределение. Будем называть $(u'_1, u'_2, e', v') \in G^4$ *валидным шифротекстом*, если $\log_{g_1} u'_1 = \log_{g_2} u'_2$. Отметим, что если шифротекст валидный и $u'_1 = g_1^{r'}$ и $u'_2 = g_2^{r'}$, то $h^{r'} = (u'_1)^{z_1} (u'_2)^{z_2}$, а значит, дешифровальный оракул выведет $\frac{e}{h^{r'}}$, как и должен. Таким образом доказательство теоремы вытекает из следующего утверждения. □

Утверждение 5.3. *Дешифровальный оракул в обоих случаях (обычная атака криптосистемы и атака симулятора) отвергает все невалидные шифротексты, кроме, разве что, чрезмерно малого количества случаев (вероятность меньше обратного полинома).*

Доказательство. Рассмотрим распределение точки $P = (x_1, x_2, y_1, y_2) \in Z_q^4$, обусловленное видением взломщика. Для него точка образуется путем случайного выбора из плоскости Q пересечения 2 гиперплоскостей:

$$\log c = x_1 + w x_2 \tag{9}$$

$$\log d = y_1 + w y_2 \tag{10}$$

Эти 2 гиперплоскости можно взять из открытого ключа.

Предположим, что взломщик подает невалидный шифротекст ($\log u'_1 = r'_1, \log u'_2 = r'_2$, а $r'_1 \neq r'_2$). Тогда дешифровальный оракул будет выдавать отказ, пока P не попадет на гиперплоскость M, определяемую как:

$$\log v' = r'_1 x_1 + w r'_2 x_2 + \alpha' r'_1 y_1 + \alpha' r'_2 w y_2 \tag{11}$$

где $\alpha' = H(u'_1, u'_2, e')$. Но видно, что уравнения (9), (10) и (11) линейно независимы, поэтому M пересекает Q по линии. Из этого следует, что даже при многочисленных попытках отправить невалидный шифротекст, он будет отвергаться с большой вероятностью (будет ошибаться не больше, чем обратный полином). Ведь при 1 попытке отправить невалидный шифротекст, дешифровальный оракул отвергнет его с вероятностью $1 - \frac{1}{q}$, которая обусловлена попаданием точки P на линию в гиперплоскости M. Повторив это i раз, мы получим вероятность, как минимум равную $1 - \frac{1}{(q-i+1)}$. □

Лемма 5.4. *Если на вход симулятору подается распределение из R, то распределение скрытого бита b не зависит от распределения видения взломщика.*

Доказательство. Пусть $u_1 = g_1^{r_1}, u_2 = g_2^{wr_2}$. Мы можем считать, что $r_1 \neq r_2$, т.к. вероятность этого пренебрежимо мала. Доказательство леммы будет следовать из следующих 2 утверждений. □

Утверждение 5.5. *Если дешифровальный оракул отвергает все невалидные шифротексты во время атаки, то распределение скрытого бита b не зависит от видения взломщика.*

Идея доказательства. Доказывается аналогично утверждению 1 (в терминах гиперплоскостей, их линейной независимости). □

Утверждение 5.6. *Дешифровальный оракул отвергает все невалидные шифротексты, кроме случаев, вероятность которых пренебрежимо мала (меньше обратного полинома).*

Идея доказательства. Как и в лемме 1, мы рассматриваем распределение $P = (x_1, x_2, y_1, y_2) \in Z_q^4$, обусловленное видением взломщика, в терминах гиперплоскостей. Предположим, что взломщик отправляет невалидный шифротекст $(u'_1, u'_2, e', v') \neq (u_1, u_2, e, v)$, где $\log u'_1 = r'_1$ и $\log u'_2 = wr'_2$, а $r'_1 \neq r'_2$. Положим $\alpha = H(u'_1, u'_2, e')$ и рассмотрим 3 случая.

Случай 1: $(u'_1, u'_2, e') = (u_1, u_2, e)$. В этом случае значения хэш-функции одинаковы, но $v \neq v'$, следовательно, дешифровальный оракул отвергнет этот шифротекст.

Случай 2: $(u'_1, u'_2, e') \neq (u_1, u_2, e)$ и $\alpha' \neq \alpha$. Дешифровальный оракул отвергнет этот шифротекст. Соображения такие же, как и в прошлые разы (нужно доказать линейную независимость соответствующих гиперплоскостей. В этом случае это сделать несколько сложнее, поэтому надо будет рассмотреть детерминант соответствующей матрицы и доказать, что он не равен 0).

Случай 3: $(u'_1, u'_2, e') \neq (u_1, u_2, e)$ и $\alpha' = \alpha$. Доказываем, что если это происходит с нетривиальной вероятностью, то данное семейство хэш-функций не является универсальным односторонним (противоречие). □

Заметим, что для достижения безопасности к неадаптивным атакам (и только им!), можно значительно упростить протокол, не использовав d, y_1 y_2 и хэш-функцию H. При шифровании мы используем $v = c^r$, а при дешифровании проверяем, что $v = {u_1}^{x_1} {u_2}^{x_2}$.

6 Заключение

В данном обзоре были приведены основные определения, необходимые для понимания работы протокола Крамера-Шупа. Так же была доказана корректность и безопасность протокола к атакам на основе адаптивно подобранного шифротекста.

7 Список использованной литературы

1. *Ronald Cramer, Victor Shoup "A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack"; 1998*
2. *Henk van Tilborg "Encyclopedia of cryptography and security"; 2005*