Проект CLIQUES распределения ключей для групп с динамическим составом участников.

Введение

В настоящее время организация безопасной связи внутри групп абонентов с динамически меняющимся составом участников является достаточно сложной задачей, отличающейся по своему качественному составу от классических задач криптографии. Она включает в себя множество сопутствующих задач, начиная от создания основных алгоритмов и заканчивая созданием конечных приложений и коммуникационных систем. Выделяют два основных аспекта безопасности при работе в группах – секретность (т. е. все взаимодействия внутри группы остаются секретными для лиц, не являющихся участниками группы) и аутентификация.

Стандартным подходом к обеспечению безопасности для групп является получение некоторой секретной величины, известной только участникам группы. Криптографические протоколы, в которых происходят выработка и распространение этой величины внутри группы известны как *распределение ключа группы (group key establishment)*. В случае, когда это значение не вырабатывается в протоколе, а приобретается заранее кем-либо из участников, протокол носит название *протокола распространения ключей в группе(group key distribution)*. В случае, когда каждый участник группы участвует в генерации этого секретного значения, мы получаем *протокол обмена ключами (group key agreement)*. В обоих случаях только действующие участники группы имеют доступ к этому групповому секрету (действующие потому, что предполагается высокая динамичность группы). При любом присоединении нового участника или выходе участника из группы секретное значение меняется для предотвращения НСД со стороны лиц, не входящих в группу.

1 Используемые в протоколах термины и обозначения

Определим некоторые обозначения:

- n - число участников протокола;
- i, j - индексы для участников групп;
- M_i - i-ый участник группы;
- G - циклическая группа G порядка q, где q – простое;
- α, g - образующие элементы в группе G ;
- x_i - долговременный секретный ключ M_i;
- r_i - случайное (секретное) число $\in Z_q$, вырабатываемое M_i;
- S_n - групповой ключ n участников;
- $S_n(M_i)$ - вклад M_i -го участника в групповой ключ;
- K_{ij} - долговременная секретная величина, выработанная M_i и M_j , $i \neq j$.

Все вычисления проводятся в циклической группе G простого порядка q, причем $p=kq+1$ для некоторого $k \in N$.

2 Протоколы аутентичного обмена ключами

Простейшей схемой получения общего ключа является схема с доверенным сервером, в котором кто-либо посылает ему запрос на связь с другими абонентами, и сервер рассылает каждому абоненту общий ключ для связи внутри группы и список участников группы, зашифрованные ключом абонента. Но при такой схеме возникают сложности при высокой динамичности группы, обусловленные невозможностью одновременной обработки сервером большого числа запросов. Поэтому рассмотрим некоторые специально созданные протоколы для получения общего ключа участниками группы.

В рамках предварительного знакомства приведем аутентичный обмен для выработки ключа для двух сторон. Затем приведем расширение этого протокола для n сторон. Приводимые протоколы базируются на схеме Диффи-Хеллмана.

2.1 Протоколы A-DH, GDH.2 и A-GDH.2

Прежде чем привести описание протокола аутентичного обмена для двух сторон A-DH, важно подчеркнуть, что существует множество разнообразных протоколов аутентичного обмена для выработки ключа, но одни из них не поддерживают двусторонний вклад в общий ключ (как в El Gamal), другие требуют большого числа сообщений или предполагают априорный доступ к сертифицированным долговременным ключам. Необходимо также отметить, что протокол предполагает наличие у участников аутентичных открытых ключей друг друга.

Протокол A-DH. Пусть p,q,G – величины, определенные выше и пусть α – образующий элемент G.
Предварительный этап. Пусть x_1 и x_2 – два целых числа, т. ч. $1 \le x_1, x_2 \le q-1$. Пусть M_1 и M_2 – два участника, которые хотят выработать общий ключ и пусть (x_1 ,$\alpha^{x1} \bmod p$) и (x_2 ,$\alpha^{x2} \bmod p$)- секретные и открытые ключи M_1 и M_2 соответственно. Открытые величины системы: (p, q, α, α^{x1} ,α^{x2}).
Этап 1:
M_1 выбирает случайное r_1 ,
$M_1 \rightarrow M_2 : \alpha^{r_1} \bmod p$.
Этап 2:
M_2 выбирает случайное r_2 и вычисляет $K=F(\alpha^{x_1 x_2} \bmod p)$,
$M_2 \rightarrow M_1 : \alpha^{r_2 K} \bmod p$.
Когда M_1 получает $J=\alpha^{r_2 K} \bmod p$, он вычисляет $K^{-1} \bmod q$ и затем $J^{r_1 K^{-1}} \bmod p$.
Получаемый в результате ключ будет $S_2=\alpha^{r_2 r_1} \bmod p$. Функция $F(x)$ может быть $F(x)=x \bmod q$.

Очевидно, что в полученном в результате ключе имеется вклад обоих сторон (т.к. r1 и r2 случайны и вырабатываются разными сторонами), т.е. протокол обладает контрибутивностью. В то же время обеспечивается

аутентификация ключа, поскольку при его формировании участвуют открытые ключи обоих абонентов, которые переданы по аутентичному каналу.

Рассмотрим теперь протокол Диффи-Хеллмана для групп .

Протокол GDH.2. Пусть $M = \{M_1, M_2 \ldots M_n\}$ – множество пользователей, которым необходимо выработать общий ключ S_n . GDH.2 протокол выполняется за n шагов. На первой стадии (n-1 этапе) идет сбор информации от отдельных участников группы, а на второй стадии (n шаге) всем рассылается материал для вычисления общего ключа.
Предварительный этап. Пусть p – простое и q – простой делитель p-1. Пусть α – образующий элемент G.
Этап *i*:
M_i выбирает случайное r_i ,
$M_i \rightarrow M_{i+1}$: $\{\alpha^{r_1 \ldots r_i / r_j} \mid j \in [1,i]\}$, $\alpha^{r_1 \ldots r_i}$.
Этап *n*:
M_n выбирает случайное r_n ,
$M_n \rightarrow$Каждому M_i : $\{\alpha^{(r_1 \ldots r_n) / r_i} \mid i \in [1,n]\}$.
Общим ключом будет значение $\alpha^{r_1 \ldots r_n}$.

Данный протокол можно модифицировать для обеспечения аутентификации ключа. Такая модификация отличается от выше приведенного только последним этапом. Предполагается, что M_n имеет с каждым M_i общий секрет $K_{in}=F(\alpha^{x_i x_n} \bmod p)$, где x_i-секретное долговременное значение M_i , $\alpha^{x_i} \bmod p$ – долговременный открытый ключ M_i .

Протокол A-GDH.2.
Этапы с 1 по *n-1* : такие же, как и в GDH.2.
Этап *n*:
M_n выбирает случайное r_n ,
$M_n \rightarrow$Каждому M_i : $\{\alpha^{r_1 \ldots r_n K_{in} / r_i} \mid i \in [1,n]\}$.
При получении M_i вычисляет $\alpha^{(r_1 \ldots r_n K_{in} / r_i) K_{in}^{-1} r_i} = \alpha^{r_1 \ldots r_n} = S_n$.

В этом протоколе каждый участник группы вырабатывает общий аутентичный ключ с M_n. Более того, если мы доверяем M_n , то каждый участник группы может быть уверен, что такой же ключ имеют и все участники группы, т.е. они выработали общий групповой ключ.
Очевидно, что протокол обладает свойством контрибутивности, поскольку в результирующем ключе S_n есть вклад i-го участника группы в виде степени r_i.

3 Проект CLIQUES

Целью данного проекта являлась разработка протокола обмена для выработки ключа для групп. Такой протокол должен поддерживать все групповые операции по удалению, включению новых участников в группу. На основе этого протокола необходимо было создать специальный прикладной программный интерфейс (CLQ-API) , позволяющий работать приложениям в неких абстрактных группах. Протокол во многом основывается на вышеописанных протоколах аутентичного обмена. Ограничимся рассмотрением только математических принципов проекта.

В качестве базового протокола обмена для выработки общего ключа был выбран протокол A-GDH.2 . Предполагается, что участники группы уже сформировали общий ключ.

Рассмотрим основные операции, которые позволяет выполнять разработанный протокол.

1. **Операции для одного участника группы:** включают в себя добавление или удаление одного участника группы. Данные ситуации появляются, когда кто-то хочет присоединиться к группе или покинуть ее. Эти операции могут проводится контролером группы или по согласию каждого участника группы (в зависимости от используемой политики безопасности).
2. **Операции для нескольких участников:** также включают в себя добавление и удаление. Однако есть отличия, обусловленные желанием проводить операции с несколькими участниками сразу, а не с каждым в отдельности:

- массовое присоединение: несколько участников хотят присоединиться к существующей группе;
- слияние групп: две или более групп желают соединиться в одну;
- массовый выход из группы: несколько участников хотят покинуть группу;
- разделение групп: группа распадается на две или более частей.

Итак, список операций, выполняемых протоколом, выглядит следующим образом:

- присоединение (JOIN): новый участник добавляется в группу;
- слияние (MERGE): один или более участников добавляются в группу;
- выход из группы (LEAVE): один или более участников покидают группу;
- обновление ключа (KEY REFRESH): генерация нового ключа для группы.

Для простоты, считается, что последний участник группы является контролирующим группы (это может быть легко исправлено и не является критическим требованием).

Присоединение

Операция добавляет нового участника M_{n+1} к группе из n участников. Во время операции вычисляется новый групповой ключ S_{n+1} , и M_{n+1} становится новым контролирующим группы. Предполагая, что M_n является текущим контролирующим группы, протокол выглядит следующим образом:

1. M_n вырабатывает новое значение r_n' и получает множество[1] чисел
$$M=\{g^{r1\ldots r_n'/r_i} \mid i\in[1,n\text{-}1]\} \cup \{ g^{r1\ldots r_{n-1}} \} \cup \{ g^{r1\ldots r_n'} \}$$
Затем M посылается M_{n+1}.
2. После получения сообщения M_{n+1} вырабатывает число r_{n+1} и вычисляет значение $g^{Ki,n+1r1\ldots r_n'r_{n+1}/r_i}$ для всех i из $[1,n]$. Затем это множество рассылается всей группе.
3. При получении каждый M_i вычисляет групповой ключ как $(g^{Ki,n+1r1\ldots r_n'r_{n+1}/r_i})^{K\text{-}1}{}_{i,n+1}{}^{ri} = g^{r1\ldots r_n'r_{n+1}} = S_{n+1}$. А M_{n+1} вычисляет ключ, используя сообщение из шага (1).

Шаги (1) и (2) требуют n экспоненцирований, шаг (3) требует одно экспоненцирование для каждого участника группы. Общее число экспоненцирований для получения ключа равно $2n+1$ (считается, что на третьем шаге экспоненцирования происходят одновременно и по времени равны одному).

Слияние

Операция используется для добавления $k>0$ участников к существующей группе из $n>1$ участников. Пусть $m=n+k$. Во время операции вырабатывается новый групповой ключ S_m, и M_m становится новым контролирующим группы. Предполагая, что M_n является текущим контролирующим группы, протокол выглядит следующим образом:

1. M_n вырабатывает новое значение r_n' и вычисляет[2] $g^{r1\ldots r_{n-1}r_n'}$. Затем это сообщение отправляется к M_{n+1}.
2. Каждый участник M_j , $j=n+1,\ldots,m\text{-}1$ вырабатывает число r_j и вычисляет $g^{r1\ldots r_n'\ldots r_j}$. Это сообщение посылается M_{j+1}.
3. После получения сообщения, M_m рассылает полученное значение всей группе
4. После получения сообщения каждый участник M_i, $i=1,2,\ldots,m\text{-}1$ группы вычисляет $g^{(r1\ldots,r_n'\ldots r_{m-1})/r_i}$ и посылает его M_m.
5. M_m вырабатывает r_m и получает множество
$M=\{ g^{Ki,m\,r1\ldots r_n'\cdots r_m/r_i} \mid i\in[1,m\text{-}1]\}$.
Затем оно посылается группе.

---

[1] Необходимые данные для вычисления множества M_n берет из последнего этапа протокола A-GDH.2 и возводя затем нужные элементы в степень $r_n'(K_{in}^{-1} mod\ p)$ получает необходимые значения.

[2] Значение $g^{r1\ldots r_{n-1}}$ M_n может получить из предыдущего ключа путем возведения в степень r_n^{-1}.

6. При получении сообщения шага (5) каждый M_i , i=1,2…m-1 вычисляет групповой ключ как $(g^{r1\ldots rn'\ldots rmKim/ri})^{Kim^{-1}ri} = g^{r1\ldots rn'\ldots rm} = S_m$. Аналогично, M_m вычисляет ключ, используя сообщение из шага (3).

   Если k=2, то шаг (2) не нужен, в остальном протокол выглядит также.

Шаги требуют всего k модульных экспоненцирований. Также, как и ранее, шаги (4) и (6) требуют по одному для каждого участника. Шаг (5) требует n+k-1 экспоненцирований. Число экспоненцирований для присоединения k участников равно n+2k+1.

Операция присоединения также может быть использована для добавления k участников к группе. Это потребует повторить операцию присоединения k раз – соответственно возрастает трудоемкость операции. Таким образом для массового добавления участников группы лучше использовать операцию слияния. Если использовать операцию слияния для добавления одного участника к группе, то получается на два экспоненцирования больше, чем для операции присоединения. Итак, присоединение используется для добавления одного участника к группе, а слияние – нескольких.

Выход из группы

Операция выхода из группы удаляет k участников из n участников текущей группы. Во время операции вычисляется новый групповой ключ S_{n-k}. M_{n-k} становится новым контролирующим группы, если M_n покидает группу. Для простоты предположим, что только один участник M_d выходит из состава группы. Протокол выглядит следующим образом:

1. M_n вырабатывает новое r_n' и получает множество
   $M=\{ g^{Kin\, r1\ldots rn'/ri} \mid i\in[1,n\text{-}1]$ и $i\neq d\}$
   Затем M рассылается всем.
2. При получении сообщения M_i вычисляет
   $(g^{Kin\, r1\ldots rn'/ri})^{Kin^{-1}ri} = g^{r1\ldots rn'} = S_n$. M_n вычисляет новый групповой ключ $g^{r1\ldots rn'}=S_n$ используя старое значение.

Участник M_d не может вычислить новый групповой ключ, т.к. контролирующий группы не вычислил вспомогательный ключ $g^{Kdnr1\ldots rn'/rd}$ для M_d. Если несколько участников покидают группу, то контролирующий группы не вычисляет нужные значения для выходящих из группы участников на шаге (1).
Если из группы выходит контролирующий, то вышеописанные операции выполняет предпоследний участник группы M_{n-1}. Более того, поскольку новый контролирующий группы не может удалить из ключа долговременные ключи (они были у прошлого контролирующего группы), каждый участник M_i должен заново вычислить свой случайный сеансовый ключ как $r_i = r_i*(K_{in}^{-1} \bmod q)$ перед выполнением шага (2).
Шаг (1) требует n-k экспоненцирований. Шаг (2) требует одно экспоненцирование от каждого участника группы. Таким образом, операция выхода из группы требует n-k+1 модульных экспоненцирований.

Обновление ключа

Операция обновления ключа выполняет замену группового ключа на новый. Использование этой операции зависит от используемой политики приложения, использующего CLIQUES или политики работы с ключами для предприятия. Эта операция выглядит также, как и операция выхода из группы с k=0, т.е. на первом шаге M_n вырабатывает множество

$$M=\{ g^{Kin\, r1 \ldots rn'/ri} \mid i \in [1,n-1]\}$$

для всех участников группы.

Приведенный протокол является протоколом аутентичного обмена и обладает свойством контрибутивности, что гарантирует независимость ключа (так как в его формировании участвуют все участники группы), может обеспечивать подтверждение ключа (как это было описано выше), устойчив к атакам по известному ключу (многие свойства следуют из рассмотренного ранее протокола A-GDH.2).

Таким образом, с использованием приведенных выше операций достигается полноценная работы группы. На основе приведенного протокола был разработан интерфейс прикладного программирования (*Application Programming Interface - API*). Он принят как проект стандарта для Internet. Программные реализации математических принципов, используемых в протоколах, можно найти в крипто-библиотеках Crypto++[3] и RSAREF[4]. Между тем, приведенная схема работы не лишена недостатков. Во-первых – и это, наверное, самый главный из них – приходится менять ключ для всей группы при изменении ее состояния. Это может подходить для небольших групп, но при большом числе участников становится серьезной трудностью. В этом случае все будет зависеть от динамики группы. Также решающую роль играет пропускная способность каналов связи между участниками, поскольку в случае появления участника со слабым каналом (тем более, если это контролирующий группы) встает вопрос о временных факторах формирования ключа. Возможна ситуация непроизвольного «выкидывания» (в случае отсутствия необходимых механизмов) участника, использующего канал с низкой пропускной способностью в случае высокой динамики группы. Он просто не будет успевать получать новые данные для формирования ключа. Во-вторых – довольно высокое число экспоненцирований в операциях протокола.

Используемые работы:

[1] G. Ateniese, M. Steiner, G. Tsudik “Authenticated Group Key Agreement and Friends”,
[2] M. Steiner, G. Tsudik, M. Waidner “Diffie-Hellman key distribution extended to groups”, in *ACM Conference on Computer and Communications Security*, pp.31-37, ACM Press, Mar. 1996.
[3] W. Dai “Crypto++”, 05.1999, http://www.eskimo.com/~wedai/cryptolib.html
[4] RSA Laboratories, http://www.rsalab.com