Предложенный метод может быть реализован с помощью программного средства, а также экспериментально. Первый способ является более перспективным с точки зрения производительности, однако требует высокоточной математической модели КФУ и стопроцентного входного контроля элементов. Второй способ требует наличия специального тестирующего оборудования, в котором будут монтироваться элементы на время эксперимента. После этого элементы должны демонтироваться и подаваться на окончательную сборку. Использовать метод экспериментально можно в единичном и мелкосерийном производстве.

Список литературы: 1.*Фомин А. В.* Допуски и посадки в РЭА. М.: Сов. радио., 1973. 128 с. **2.** *Буловский П. И.* Автоматизация селективной сборки приборов. Л.:Машиностроение, 1978. 232 с. **3.** *Мамонтов А.В.* Оптимальная комплектация конструктивно-функциональных узлов электрорадиоэлементами // Радиотехника. 2001. Вып. 117. С. 125-126. **4.** *Левит М.Е.* Балансировка деталей и узлов. М.: Машиностроение, 1986. 248 с. **5.** *Кацман М. М.* Электрические машины. М.: Высш. школа. 1983. 432 с.

Поступила в редколлегию 20.02.2006

Мамонтов Александр Викторович, ст. пр. кафедры охраны труда ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14.

УДК 519.713

О.А.БАКАЕВ

ПРОТОКОЛ ДИНАМИЧЕСКОГО ФОРМИРОВАНИЯ ГРУППОВЫХ СЕКРЕТНЫХ КЛЮЧЕЙ В МНОГОПОЛЬЗОВАТЕЛЬСКИХ СИСТЕМАХ

Рассматривается заверенная схема обмена ключа группы для случаев динамического изменения состава группы в специальных сетях, где каждый член группы может присоединиться и/или отсоединиться в любое время, и ключ группы будет изменен без помощи центрального сервера.

1. Введение

Цель: рассмотреть распределительный протокол ID-based согласования ключа, использующий билинейное сложение. Выявить важную особенность схемы, обеспечивающую аутентификацию всех объектов одновременно, при посылке ключа распределения только с одной стороны.

Актуальность: В коммерческом секторе многочисленные инструкции секретности диктуют, что важная информация должна быть защищена от риска нарушения конфиденциальности. Для управлений критические коммуникации должны быть защищены. Практически, защищенная передача сообщений требует комбинации установления подлинности, конфиденциальности (чтобы предотвратить подслушивание) и подписи (чтобы предотвратить вмешательство в содержание сообщения). С огромными проблемами столкнулись все функционирующие системы, пытающиеся обеспечить эти услуги, особенно в вопросах формирования и управления криптографическими ключами, а также вопросах идентификации и подтверждения подлинности пользователей.

Ключевое соглашение – одно из фундаментальных криптографических элементарных вычислений. Такие протоколы позволяют двум или больше субъектам обмениваться информацией между собой по открытой опасной сети и согласовать общий ключ сеанса, который может использоваться для более поздней безопасной коммуникации среди объектов. Таким образом, безопасные ключевые протоколы соглашения служат основным стандартным блоком для того, чтобы создать безопасные, сложные, высокоуровневые протоколы.

Проектирование эффективных ключевых протоколов соглашения и в двусторонних, и в многосторонних (т.е.группа) установках с более низким вычислением и стоимостью коммуникации требует большого внимания. Первая, ведущая работа по ключевому соглашению - протокол Дифи-Хелмана, где впервые появляется шифрование с открытым ключом. Однако основной протокол Дифи-Хелмана не подтверждает подлинность двух объектов коммуникации в том смысле, что активный противник, который управляет каналом, может

предпринять атаку “человек в центре”, чтобы согласовать два отдельных ключа с пользователями.

Подтвержденное соглашение ключа Дифи-Хелмана позволяет группе пользователей в пределах большой и полностью ненадежной общественной сети устанавливать общий секретный ключ и удостоверить каждого пользователя, чтобы никакой другой, кроме конкретно установленного пользователя, не имел возможности узнать значение секретного ключа. Это неявная ключевая идентификация и протокол называется протоколом соглашения с идентифицированным ключом. В дополнение, протокол соглашения с идентифицированным ключом спроектирован так, чтобы убедить объекты в том, что они действительно совместно используют секретный ключ. Это свойство называют явной ключевой идентификацией, и протокол называется протоколом формирования секретного ключа с подтверждением аутентичности владельца. За последние годы множество свойств безопасности стало важным в ключевых протоколах соглашения, были разработаны различные подходы, чтобы решить проблему. Стандартная выработка ключей, код идентификации сообщения, цифровые подписи и т.д. являются основными средствами, используемыми для подтверждения подлинности ключей соглашения.

Очень эффективная заверенная схема обмена ключа группы (AGKE) была разработана Ким, Ли и др. для того, чтобы динамически изменить группу пользователей в специальных сетях, где член группы может присоединиться и/или отсоединиться в любое время, и ключ группы будет изменен без помощи любого центрального сервера. Предложенная ими схема безопасна и требует постоянных раундов.

2. Описание протокола согласования ключей

Установка: протокол основан на предположении CDH и безопасной схеме подписи $\sum = (K,S,V)$. Мы берем пространство ключей $\{0,1\}^l$, где l – параметр безопасности.

Пусть G будет мультипликативной группой некоторого большого главного порядка $q\,(l \le |q|)$ и g будет генератором G. Мы также рассматриваем одностороннюю хеш-функцию $H:\{0,1)^* \to \{0,1\}^l$. В фазе инициализации каждому пользователю U_i предоставляют общую ключевую пару (PK_{Ui}; SK_{Ui}) для подтверждения/подписания. Список общих ключей дан всем пользователям.

Соглашение ключей: протокол состоит из трех алгоритмов GKE.Setup (установка), GKE.Join (присоединение) и GKE.Leave (отсоединение) для начальной установки, пользовательского объединения и пользовательского отсоединения соответственно.

3. Протокол установки (GKE.Setup)

Пусть $G_0 = \{U_1,...,U_n\}$ будут начальной группой и ID_{Ui} будет идентификатором пользователя U_i. Мы рассматриваем кольцевую структуру среди членов G_0 и допускаем $I_0 = ID_{U1} | ID_{U2} | ... | ID_{Un}$.

Протокол выполняется в двух раундах следующим образом.

В раунде 1 каждый пользователь U_i беспорядочно выбирает $k_i\{0,1\}^l$ и $x_i \in Z_q^*$, вычисляет $y_i = g^{x_i}$ и сохраняет k_i тайну. Последний пользователь U_n вычисляет $H(k_n | 0)$. Каждый пользователь U_i генерирует подпись $\sigma_i^{(1)} S_{SKU_i}(M_i^{(1)} | I_0 | 0)$, где $M_i^{(1)} = y_i$ для $1 \le i \le n-1$ и $M_i^{(1)} = H(k_n | 0) | y_n$, и передает $M_i^{(1)} | \sigma_i^{(1)}$.

В раунде 2 каждый пользователь U_i, при получении $M_i^{(l)} | \sigma_i^{(l)}$, проверяет $\sigma_i^{(l)}$ на $M_i^{(l)}$ для всех $j \ne i$. Если некоторые подписи не правильны, процесс терпит неудачу и останавливается. Иначе, пользователь U_i вычисляет $t_i^L = H(y_{i-1}^{x_i} | I_0 | 0), t_i^R = H(y_{i+1}^{x_i} | I_0 | 0)$ и генерирует $T_i = t_i^L \oplus t_i^R$. Последний пользователь U_n дополнительно вычисляет $T^* = k_n \oplus t_n^R$. Каждый пользователь U_i теперь генерирует подпись $\sigma_i^{(2)} S_{SKU_i}(M_i^{(2)} | I_0 | 0)$ и передает $M^{(2)} = T^* | T_n$.

В фазе вычисления ключа каждый пользователь U_i определяет подпись $\sigma_j^{(2)}$, $j \ne i$. Если все подписи правильны, U_i вычисляет $t_{i+1}^R, t_{i+2}^R, ..., t_{i+(n-1)} (= t_i^L)$, используя t_i^R следующим образом:

$$t_{i+1}^{R} = T_{i+1} \oplus t_{i}^{R}, t_{i+2}^{R} = T_{i+2} \oplus t_{i+1}^{R}, ..., t_{i+(n-1)}^{R} = T_{i+(n-1)} \oplus t_{i+(n-2)}^{R}.$$

Тогда U_i проверяет, если $t_i^L = t_i^{-L}$. Отметим, что упомянутая выше проверка дает возможность честным пользователям обратить внимание на ошибки, вызванные системными ошибками или неправильными сообщениями, переданными незаконными пользователями, и останавливает протокол. Однако это не просто для поиска, кто передал незаконные сообщения. Наконец, каждый пользователь получает k_n от T и проверяет, если $H('k_n \mid 0) = H(k_n \mid 0)$. Это значение проверки гарантирует управление ключом. Все пользователи тогда вычисляют ключ сеанса $sk_0 = H(k_1 \mid k_2 \mid ... \mid k_{n-1} \mid k_n \mid 0)$.

Каждый пользователь U_i дополнительно вычисляет $h_i^L = H(y_{i-1}^{x_i} \mid sk_0 \mid 0), h_i^R = H(y_{i+1}^{x_i} \mid sk_0 \mid 0)$ и $X = H(k_n \mid sk_0 \mid 0)$, сохраняет (h_i^L, h_i^R, X, sk_0) тайно и стирает другие эфемерные данные. Вычисленные значения этого отправления будут использоваться для последующего объединения или операции отсоединения.

4. Протокол присоединения (GKE.Join)

Пусть $G_{V-1} = \{U_1, ..., U_n\}(v \geq 1)$ будет текущей группой и $J = \{U_{n+1}, ..., U_{n+n'}\}(n' \geq 1)$ будет рядом новых пользователей, желающих присоединяться к группе G_{v-1}. Группа G_{v-1} разделена на три части: $\{U_1\}, \{U_2, ..., U_{n-1}\}$ и $\{U_n\}$. Мы рассматриваем $\{U_2\}$ как представитель $\{U_2, ..., U_{n-1}\}$ и для удобства объяснения допускаем, что $U_{n+n'+1}, U_{n+n'+2}$ и $U_{n+n'+3}$ обозначают U_1, U_2, U_3 соответственно. В этом алгоритме кольцевую структуру рассматривают среди пользователей $U_{n+1}, ..., U_{n+n'+3}$. Пусть G будет набором $\{U_{n+1}, ..., U_{n+n'+3}\}$ и $I_y = ID_{U1} \mid ... \mid ID_{U_{n+n'}}$. Пользователи в G действуют следующим образом.

В раунде 1 каждый пользователь $U_{n+i}G$ беспорядочно выбирает $k_{n+i} \in \{0,1\}^l$ и $x_{n+i} \in Z_q^*$, вычисляет $y_{n+i} = g^{x_{n+i}}$ и сохраняет секрет k_{n+i}. Пользователь $U_{n+n'+2}(= U_2)$ вычисляет $y_{n+n'+2} = g^X$ при использовании секретного значения X вместо $x_{n+n'+2}$, и пользователь $U_{n+n'+3}(= U_3)$ вычисляет $H(k_{n+n'+3} \mid v)$. Каждый пользователь U_{n+i} генерирует подпись $\sigma_{n+i}^{(1)} = S_{SKU_{n+i}}(M_{n+i}^{(1)} \mid I_y \mid v)$, где $M_{n+i}^{(1)} = y_{n+i}$ для $1 \leq i \leq n'+2$ и $M_{n+n'+3}^{(1)} = H(k_{n+n'+3} \mid v \mid y_{n+n'3}$ и передает $M_{n+i}^{(1)} \mid \sigma_{n+i}^{(1)}$.

В раунде 2 все пользователи при получении $M_{n+i}^{(1)} \mid \sigma_{n+i}^{(1)}$ определяют $\sigma_{n+i}^{(1)}$'s. Каждый U_{n+i} вычисляет $t_{n+i}^L = H(y_{L_{(n+i)}}^{x_{(n+i)}} \mid I_v \mid v), t_{n+i}^R = H(y_{R_{n+i)}}^{x_{n+i}} \mid I_v \mid v)$ и генерирует $T_{n+i} = t_{n+i}^L \oplus t_{n+i}^R$, где $L_{(n+i)}$ и $R_{(n+i)}$ соответственно означает левый и правый индекс $n+i$ на кольце, поскольку $i \in \{1, ..., n'+3\}$. Пользователь $U_{n+n'+3}$ дополнительно вычисляет $T' = k_{n+n'+3} \oplus t_{n+n'+3}^R$. Каждый пользователь U_{n+i} генерирует подпись $\sigma_{n+i}^{(2)} = S_{SKU_{n+i}}(M_{n+i}^{(2)} \mid I_y \mid v)$ и передает $M_{n+i}^{(2)} \mid \sigma_{n+i}^{(2)}$, где $M_{n+i}^{(2)} = k_{n+i} \mid T_{n+i}$ для $1 \leq i \leq n'+2$ и $M_{n+n'+2}^{(2)} = T' \mid T_{n+n'+2} = T' \mid T_{n+n'+2}$. Все пользователи $\{U_3, ..., U_{n-1}\}$ вычисляют $t_{n+n'+2}^L$ и $t_{n+n'+2}^R$ при использовании X.

В фазе вычисления ключа все пользователи проверяют подписи $\sigma_{n+i}^{(2)}$. Если все подписи являются правильными, каждый пользователь U_{n+i} вычисляет $'t_{n+i+1}^R, ..., 't_{n+i+n'-1}(='t_{n+i}^L)$,

при использовании $'t^R_{n+i}$ проверяет, если $t^R_{n+i} = 't^R_{n+i}$ держится. Кроме того, пользователи $U_3,...,U_{n-1}$ могут проверить это, используя $t^L_{n+n'+2}$ и $t^R_{n+n'+2}$. Наконец, все пользователи возвращают $k_{n+n'+3}$ для $'T$ и вычисляют новый ключ сеанса $sk_y = H(k_{n+1} | ... | k_{n+n'+3} | v)$.

Каждый новый пользователь $U_{n+i} (1 \le i \le n')$ вычисляет $h^L_{n+i} = H(^{x_{n+i}}_{L(n+i)} | sk_y | v)$ и $h^R_{n+i} = H(^{x_{n+i}}_{R(n+i)} | sk_y | v)$. Пользователи U_1 и U_n соответственно вычисляют $h^L_1 = H(y^{x_1}_{n+n'} | sk_y | v)$ и $h^R_n = H(y^{x_n}_{n+1} | sk_v | v)$ вместо предыдущего значения $h^L_i (= h^R_n)$. Все пользователи вычисляют новое значение $X = H(k_n | sk_v | v)$. Каждый пользователь U_i тогда сохраняет h^L_i, h^R_i, X и sk_y тайно и удаляет все другие эфемерные данные. Они отправляют сообщение, вычисление требуется для любого последующего объединения или операции отсоединения.

5. Протокол отсоединения (GKE.Leave)

Пусть $G_{v-1} = \{U_1,...,U_n\}$ будет текущей группой и пусть $R = \{U_1,...,U_{ln}\}$, где $\{l,...,l_{n''}\} \subset \{1,...,n\}$, будут рядом отсоединяющихся пользователей. Мы обозначаем $N(R)$ набор всех левых/правых соседей отсоединяющихся пользователей на кольце, т.е.:

$$N(R) = \{U_{L(l1)}, U_{R(l1)},..., U_{L(ln'')}, U_{R(ln'')}\}.$$

Для простоты обсуждения, пусть $U_{L(li)} = U_{li-1}$ и $U_{R(li)} = U_{li+1}$. Тогда $N(R) = \{U_{li-1}, U_{li+1},..., U_{ln''-1}, U_{ln''+1}\}$. Чтобы генерировать новую группу $G_v = G_{v-1}R$ с новым ключом сеанса sk_v, новое значение Diffie-Hellman должно быть разделено между левыми и правыми соседями U_{lj-1} и $U_{lj+1} (1 \le j \le n'')$ в соответствии с отсоединяющимся пользователем U_{lj} на кольце. В этом алгоритме отсоединения мы рассматриваем кольцевую структуру среди членов G_v и заново индексируем членов как $G_v = \{U_1,...,U_{n-n''}\}$. Пусть $I_v = ID_{nl} | ... | ID_{n-n''}$. Алгоритм выполняется в двух раундах следующим образом.

В раунде 1 каждый пользователь U_w из $N(R)$ беспорядочно выбирает $k_w \in \{0,1\}^l$ и $x_w \in Z^*_q$, вычисляет $y_w = g^{x_w}$ и сохраняет k_w тайно. Тогда пользователь $U_{ln''+1}$ вычисляет $H(k_{ln''+1} | v)$. Пользователь U_w генерирует подпись $\sigma^{(1)}_w = S_{SKU_W}(M^{(1)}_w | I_v | v)$, где $M^{(1)}_w = y_w$, с $w \in \{l_1 - 1, l_1 + 1,..., l_{n''-1}\}$ и $M^{(1)}_{ln''+1} = H(k_{ln''+1} | v | y_{ln''+1})$ и передает $M^{(1)}_w | \sigma^{(1)}_w$.

Все пользователи G_v проверяют подписи $\sigma^{(1)}_w 's$ во втором раунде. Если все подписи правильны, каждый пользователь U_{lj-1} (соответственно, U_{lj+1}) $N(R)$ регенерирует $h^R_{lj-1} = y^{x_{lj-1}}_{lj+1}$ (соответственно, $h^L_{lj+1} = y^{x_{lj+1}}_{lj-1}$). Тогда каждый пользователь U_i из $G_v t^L_i = H(h^L_i | I_v | v)$, $t^R_i = H(h^R_i | I_v | v)$ и $T_i = t^L_i \oplus t^R_i$. Пользователь $U_{ln''+1}$ генерирует подпись $'T = k_{ln''+1} \oplus t^R_{ln''+1}$. Каждый пользователь $U^{ln''+1}_i$ генерирует подпись $\sigma^{(2)}_i = S_{SKU_i}(M^{(2)}_i | I_v | v)$ и передает $M^{(2)}_i | \sigma^{(2)}_i$, где $M^{(2)}_{ln''1} = 'T | T_{ln''+1}, M^{(2)}_i = k_i | T_i$, для других пользователей, кроме $U_{ln''+1} N(R)$ и $M^{(2)}_i = T_i$ для пользователей $G_v | N(R)$.

В фазе вычисления ключа сеанса все пользователи проверяют подписи $\sigma^{(2)}_i 's$. Если все подписи правильны, каждый пользователь U_i вычисляет $'t^R_{i+1}, 't_{i+2},..., t_{i+(n-n''-1)} (= 't^L_i)$ при

использовании t_i^R и проверяет, если $t_i^L = {}' t_i^L$. Наконец, все пользователи вычисляют ключ сеанса $sk_v = H(k_{1l-1} | k_{1l+1} | ... | k_{ln''-1} | k_{ln''+1} | v)$.

Каждый пользователь U_i регенерирует $h_i^L = H(h_i^L | sk_v | v), h_i^R = H(h_i^R | sk_v | v)$ и $X = H(k_{ln''+1} | sk_v | v)$ и сохраняет h_i^L, h_i^R, X и ключ сеанса sk_v тайно. Они отправляют сообщение, вычисление требуется для любого последующего объединения или операции отсоединения.

Предположение: Проблема CDH трудна, и схема подписи безопасна.

Безопасность: Протокол доказал безопасность в случайной модели оракула.

Эффективность: Предположим, что n - число членов группы, j и l - соответственно число соединений и оставлений пользователей, h, x, e, v, s - соответственно стоимость операции хеш-функции, операции XOR, модульного возведения в степень, определения подписей и генерации подписи.

Коммуникация: Для каждого из алгоритмов GKE.Setup, GKE.Join и GKE. Leave требуемые раунды – 2, максимальные биты, посланные каждому пользователю -$| q | +3 | h | +2 | \sigma |$, где $| q |, | h |, | \sigma |$ соответственно обозначают длину q, порядок циклической группы *G*, длину хеш-функции h и длину подписи.

Вычисление: Вычисления, требуемые для алгоритмов GKE.Setup, GKE.Join и GKE.Leave - соответственно

$$3e + 4h + (n+1)x + 2s + Q(n)v, 3e + 4h + (j+1)x + 2s + Q(j)v \text{ и } 3e + 4h + (n-1)x + 2s + (1+n)v .$$

6. Вывод

Практическая значимость: Сегодня требуется надежная связь с широким множеством групп пользователей; динамические группы пользователей быстро формируются и стремительно изменяются или аннулируются. Здесь рассмотрен распределительный протокол ID-based согласования ключа, использующий билинейное сложение.

Научная новизна: Наиболее важная особенность схемы – это то, что она может обеспечивать аутентификацию всех объектов одновременно, хотя ключ распределения посылается только с одной стороны. К тому же, он удобен в применении, так как имеет небольшой передаваемый размер данных (потоки сообщений в новом протоколе состоят только из одного параметра). В этой статье новое безопасное свойство определено случайным числом, компрометирующим безопасность.

Список литературы: 1. *Al-Riyami S., Paterson K.G.* Tripartite Authenticated Key Agreement Protocols from Pairings. In proceedings of IMA Conference of Cryptography and Coding, LNCS 2898. P. 332-359. Also available at http://eprint.iacr.org/2002/035. **2.** *Asokan N., Ginzboorg P.* Key Agreement in Ad-hoc Networks. In Computer Communications, 23(18). 2000. P. 1627-1637. **3.** *Ateniese G., Steiner M., Tsudik G.* Authenticated Group Key Agreement and Friends. In proceedings of ACM CCS 1998[1]. P. 17-26, ACM Press, 1998. **4.** *Ateniese G., Steiner M., Tsudik G.* New Multi-party Authenticated Services and Key Agreement Protocols. In Journal of Selected Areas in Communications, 18(4). P. 1-13, IEEE, 2000. **5.** *Barreto P. S. L. M.* Pairing Based Crypto Lounge. Available at http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html. **6.** *Barreto P. S. L. M., Kim H. Y., Scott M.* E_cient Algorithms for Pairing Based Cryptosystems. In proceedings of Crypto 2002, LNCS 2442, P. 354-368, Springer-Verlag, 2002. Also available at http://www.iacr.org/2002/008. **7.** *Barua R., Dutta R., Sarkar P.* Extending Joux Protocol to Multi Party Key Agreement. In proceedings of Indocrypt 2003, LNCS 2904. P. 205-217, Springer-Verlag, 2003. Also available at http://eprint.iacr.org/2003/062.

Бакаев Олег Александрович, магистрант ХНУРЭ по специальности «Защита информации в компьютерных сетях и системах». Адрес: Украина, Харьковская обл., пгт. Песочин, ул. Квартальная, 6, кв. 86, тел. +380 (67) 578-48-79.