Лекция 11. Схема предварительного распределения ключей (схема Блома). Функция формирования ключа

Косолапов Ю.В.

ЮФУ

18 ноября 2020 г.

Содержание

Условия задачи:

Условия задачи:

- Имеется группа участников $P_1, ..., P_n$;

Условия задачи:

- Имеется группа участников $P_1, ..., P_n$;
- любой паре (P_i, P_j) может потребоваться выработать общий секретный сессионный ключ;

Условия задачи:

- Имеется группа участников $P_1, ..., P_n$;
- любой паре (P_i, P_j) может потребоваться выработать общий секретный сессионный ключ;
- можно каждому участнику сгенерировать $n - 1$ ключей для связи с остальными, но хотелось бы **уменьшить объем хранимой информации о ключах**.

Условия задачи:

- Имеется группа участников $P_1, ..., P_n$;
- любой паре (P_i, P_j) может потребоваться выработать общий секретный сессионный ключ;
- можно каждому участнику сгенерировать $n-1$ ключей для связи с остальными, но хотелось бы **уменьшить объем хранимой информации о ключах**.

Цель

Распределить среди $P_1, ..., P_n$ предварительный **ключевой материал** (малого объема) так, чтобы в дальнейшем любая пара (P_i, P_j) могла выработать общий секретный сессионный ключ. Ключевой материал распределяется участникам доверенным центром CA по защищенным каналам.

Условия задачи:

- Имеется группа участников $P_1, ..., P_n$;
- любой паре (P_i, P_j) может потребоваться выработать общий секретный сессионный ключ;
- можно каждому участнику сгенерировать $n-1$ ключей для связи с остальными, но хотелось бы **уменьшить объем хранимой информации о ключах**.

Цель

Распределить среди $P_1, ..., P_n$ предварительный **ключевой материал** (малого объема) так, чтобы в дальнейшем любая пара (P_i, P_j) могла выработать общий секретный сессионный ключ. Ключевой материал распределяется участникам доверенным центром CA по защищенным каналам.

Требования к протоколу (схеме)

Условия задачи:

- Имеется группа участников $P_1, ..., P_n$;
- любой паре (P_i, P_j) может потребоваться выработать общий секретный сессионный ключ;
- можно каждому участнику сгенерировать $n-1$ ключей для связи с остальными, но хотелось бы **уменьшить объем хранимой информации о ключах**.

Цель

Распределить среди $P_1, ..., P_n$ предварительный **ключевой материал** (малого объема) так, чтобы в дальнейшем любая пара (P_i, P_j) могла выработать общий секретный сессионный ключ. Ключевой материал распределяется участникам доверенным центром CA по защищенным каналам.

Требования к протоколу (схеме)

- должен быть **устойчивым относительно компрометации** части ключей (в результате обмана или сговора некоторых пользователей);

Условия задачи:

- Имеется группа участников $P_1, ..., P_n$;
- любой паре (P_i, P_j) может потребоваться выработать общий секретный сессионный ключ;
- можно каждому участнику сгенерировать $n-1$ ключей для связи с остальными, но хотелось бы **уменьшить объем хранимой информации о ключах**.

Цель

Распределить среди $P_1, ..., P_n$ предварительный **ключевой материал** (малого объема) так, чтобы в дальнейшем любая пара (P_i, P_j) могла выработать общий секретный сессионный ключ. Ключевой материал распределяется участникам доверенным центром CA по защищенным каналам.

Требования к протоколу (схеме)

- должен быть **устойчивым относительно компрометации** части ключей (в результате обмана или сговора некоторых пользователей);
- должен быть **гибким**: то есть быстро восстанавливаться как после частичной компрометации, так и после подключения новых пользователей.

Схема Блома. Инициализация

Схема Блома. Инициализация

- Выбирается конечное поле $\mathbb{F}_q$ достаточно большой мощности q (чем больше q, тем для большего числа участников работает схема).

Схема Блома. Инициализация

- Выбирается конечное поле $\mathbb{F}_q$ достаточно большой мощности q (чем больше q, тем для большего числа участников работает схема).
- Каждому участнику P_i ставится в соответствие некоторый **несекретный** элемент r_i из $\mathbb{F}_q$, при этом $r_i \neq r_j$ для $i \neq j$ (r_i будет играть роль публичного ключа участника P_i);

Схема Блома. Инициализация

- Выбирается конечное поле $\mathbb{F}_q$ достаточно большой мощности q (чем больше q, тем для большего числа участников работает схема).
- Каждому участнику P_i ставится в соответствие некоторый **несекретный** элемент r_i из $\mathbb{F}_q$, при этом $r_i \neq r_j$ для $i \neq j$ (r_i будет играть роль публичного ключа участника P_i);
- Для параметра $1 \leqslant t < n$ (отвечает за безопасность схемы) выбирается случайным образом **секретный** *симметрический* многочлен (это секретный ключ доверенного центра)

$$f(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{ij} x^i y^j, a_{ij} = a_{ji}.$$

Схема Блома. Инициализация

- Выбирается конечное поле $\mathbb{F}_q$ достаточно большой мощности q (чем больше q, тем для большего числа участников работает схема).
- Каждому участнику P_i ставится в соответствие некоторый **несекретный** элемент r_i из $\mathbb{F}_q$, при этом $r_i \neq r_j$ для $i \neq j$ (r_i будет играть роль публичного ключа участника P_i);
- Для параметра $1 \leqslant t < n$ (отвечает за безопасность схемы) выбирается случайным образом **секретный** *симметрический* многочлен (это секретный ключ доверенного центра)

$$f(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{ij} x^i y^j, a_{ij} = a_{ji}.$$

- Матрица коэффициентов S должна храниться только в центре CA:

$$S = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0t} \\ a_{10} & a_{11} & \dots & a_{1t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{t0} & a_{t1} & \dots & a_{tt} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0t} \\ a_{01} & a_{11} & \dots & a_{1t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0t} & a_{1t} & \dots & a_{tt} \end{pmatrix}.$$

Схема Блома. Инициализация

- Выбирается конечное поле $\mathbb{F}_q$ достаточно большой мощности q (чем больше q, тем для большего числа участников работает схема).
- Каждому участнику P_i ставится в соответствие некоторый **несекретный** элемент r_i из $\mathbb{F}_q$, при этом $r_i \neq r_j$ для $i \neq j$ (r_i будет играть роль публичного ключа участника P_i);
- Для параметра $1 \leqslant t < n$ (отвечает за безопасность схемы) выбирается случайным образом **секретный** *симметрический* многочлен (это секретный ключ доверенного центра)

$$f(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{ij} x^i y^j, a_{ij} = a_{ji}.$$

- Матрица коэффициентов S должна храниться только в центре CA:

$$S = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0t} \\ a_{10} & a_{11} & \dots & a_{1t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{t0} & a_{t1} & \dots & a_{tt} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0t} \\ a_{01} & a_{11} & \dots & a_{1t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{0t} & a_{1t} & \dots & a_{tt} \end{pmatrix}.$$

Важно: для симметрического полинома выполняется равенство: $f(x, y) = f(y, x)$.

Схема Блома. Распределение ключевого материала

Схема Блома. Распределение ключевого материала

- Для каждого участника P_i центр CA находит полином от одной переменной

$$g_i(x) = f(x, r_i) = A_{0i} + A_{1i}x + ... + A_{ti}x^t.$$

Схема Блома. Распределение ключевого материала

- Для каждого участника P_i центр CA находит полином от одной переменной

$$g_i(x) = f(x, r_i) = A_{0i} + A_{1i}x + ... + A_{ti}x^t.$$

- Коэффициенты (это и есть ключевой материал) $A_{0i}, ..., A_{ti}$ передаются участнику i по защищенному каналу (этот набор играет роль секретного ключа участника P_i).

Схема Блома. Распределение ключевого материала

- Для каждого участника P_i центр CA находит полином от одной переменной

$$g_i(x) = f(x, r_i) = A_{0i} + A_{1i}x + ... + A_{ti}x^t.$$

- Коэффициенты (это и есть ключевой материал) $A_{0i}, ..., A_{ti}$ передаются участнику i по защищенному каналу (этот набор играет роль секретного ключа участника P_i).
- Так как $f(x, y) = f(y, x)$, то

$$g_i(r_j) = f(r_j, r_i) = f(r_i, r_j) = g_j(r_i).$$

Схема Блома. Распределение ключевого материала

- Для каждого участника P_i центр CA находит полином от одной переменной

$$g_i(x) = f(x, r_i) = A_{0i} + A_{1i}x + ... + A_{ti}x^t.$$

- Коэффициенты (это и есть ключевой материал) $A_{0i}, ..., A_{ti}$ передаются участнику i по защищенному каналу (этот набор играет роль секретного ключа участника P_i).
- Так как $f(x, y) = f(y, x)$, то

$$g_i(r_j) = f(r_j, r_i) = f(r_i, r_j) = g_j(r_i).$$

- Поэтому для генерации общего ключа участниками P_i и P_j выполняются действия:
  - $P_i : k = g_i(r_j)$
  - $P_j : k = g_j(r_i)$

Схема Блома. Распределение ключевого материала (другой взгляд на вычисление ключевого материала)

Схема Блома. Распределение ключевого материала (другой взгляд на вычисление ключевого материала)

- В общем случае:

$$f(x,y) = \sum_{i=0}^{t}\sum_{j=0}^{t} a_{ij}x^i y^j = \sum_{i=0}^{t} x^i \left(\sum_{j=0}^{t} a_{ij}y^j\right).$$

Схема Блома. Распределение ключевого материала (другой взгляд на вычисление ключевого материала)

- В общем случае:

$$f(x,y) = \sum_{i=0}^{t}\sum_{j=0}^{t} a_{ij}x^i y^j = \sum_{i=0}^{t} x^i \left(\sum_{j=0}^{t} a_{ij}y^j\right).$$

- Для $r \in \mathbb{F}_q$:

$$f(x,r) = \sum_{i=0}^{t}\sum_{j=0}^{t} a_{ij}x^i r^j = \sum_{i=0}^{t} x^i \left(\sum_{j=0}^{t} a_{ij}r^j\right) = \sum_{i=0}^{t} x^i A_i$$

Схема Блома. Распределение ключевого материала (другой взгляд на вычисление ключевого материала)

- В общем случае:

$$f(x,y) = \sum_{i=0}^{t}\sum_{j=0}^{t} a_{ij}x^i y^j = \sum_{i=0}^{t} x^i \left(\sum_{j=0}^{t} a_{ij}y^j\right).$$

- Для $r \in \mathbb{F}_q$:

$$f(x,r) = \sum_{i=0}^{t}\sum_{j=0}^{t} a_{ij}x^i r^j = \sum_{i=0}^{t} x^i \left(\sum_{j=0}^{t} a_{ij}r^j\right) = \sum_{i=0}^{t} x^i A_i$$

- Поэтому

$$\begin{pmatrix} a_{00} & a_{01} & \dots & a_{0t} \\ a_{10} & a_{11} & \dots & a_{1t} \\ \vdots & \vdots & \dots & \vdots \\ a_{t0} & a_{t1} & \dots & a_{tt} \end{pmatrix} \cdot \begin{pmatrix} r^0 \\ r^1 \\ \vdots \\ r^t \end{pmatrix} = \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_t \end{pmatrix}$$

Схема Блома. Добавление нового участника

Схема Блома. Добавление нового участника

- Для нового участника с номером N доверенный центр CA выбирает новое число r_N, не совпадающее с ранее использованными.

Схема Блома. Добавление нового участника

- Для нового участника с номером N доверенный центр CA выбирает новое число r_N, не совпадающее с ранее использованными.
- Вычисляет ключевой материал

$$S \cdot \begin{pmatrix} r_N^0 \\ r_N^1 \\ \vdots \\ r_N^t \end{pmatrix} = \begin{pmatrix} A_{N,0} \\ A_{N,1} \\ \vdots \\ A_{N,t} \end{pmatrix}$$

Схема Блома. Добавление нового участника

- Для нового участника с номером N доверенный центр CA выбирает новое число r_N, не совпадающее с ранее использованными.
- Вычисляет ключевой материал

$$S \cdot \begin{pmatrix} r_N^0 \\ r_N^1 \\ \vdots \\ r_N^t \end{pmatrix} = \begin{pmatrix} A_{N,0} \\ A_{N,1} \\ \vdots \\ A_{N,t} \end{pmatrix}$$

- CA ключевой материал $(A_{N,0}, ..., A_{N,t})$ по защищенному каналу передает участнику P_N.

О безопасности схемы Блома

Утверждение

Коалиция из $r \leqslant t$ участников не получает какой-либо информации (в смысле энтропии Шеннона) об общих ключах оставшихся участников.

Доказательство.

(Схема) Представить множество всех пар ключей в виде

$$K = (G \cdot S)^T \cdot G,$$

где

$$G = \begin{pmatrix} r_1^0 & \dots & r_n^0 \\ r_1^1 & \dots & r_n^1 \\ \vdots & \ddots & \vdots \\ r_1^t & \dots & r_n^t \end{pmatrix}.$$

Применение протокола Блома

В технологии HDCP

HDCP (англ. High-bandwidth Digital Content Protection — «защита широкополосного цифрового содержимого») — технология защиты медиаконтента, разработанная корпорацией Intel и предназначенная для предотвращения незаконного копирования высококачественного видеосигнала. Защищённый видеосигнал может быть воспроизведён только на оборудовании, поддерживающем **HDCP**.

Функция формирования ключа

Функция формирования ключа (KDF - key derivation function)

Функция формирования ключа – функция, формирующая один или несколько секретных ключей на основе данных **источника начального ключевого материала** с помощью псевдослучайной функции. Для формирования ключей к секретным данным источника начального ключевого материала могут добавляться несекретные.

Функция формирования ключа

Функция формирования ключа (KDF - key derivation function)

Функция формирования ключа – функция, формирующая один или несколько секретных ключей на основе данных **источника начального ключевого материала** с помощью псевдослучайной функции. Для формирования ключей к секретным данным источника начального ключевого материала могут добавляться несекретные.

Назначение

Функция формирования ключа

Функция формирования ключа (KDF - key derivation function)

Функция формирования ключа – функция, формирующая один или несколько секретных ключей на основе данных **источника начального ключевого материала** с помощью псевдослучайной функции. Для формирования ключей к секретным данным источника начального ключевого материала могут добавляться несекретные.

Назначение

- для создания ключа необходимой длины;

Функция формирования ключа

Функция формирования ключа (KDF - key derivation function)

Функция формирования ключа – функция, формирующая один или несколько секретных ключей на основе данных **источника начального ключевого материала** с помощью псевдослучайной функции. Для формирования ключей к секретным данным источника начального ключевого материала могут добавляться несекретные.

Назначение

- для создания ключа необходимой длины;
- для создания ключа заданного формата.

Функция формирования ключа

Функция формирования ключа (KDF - key derivation function)

Функция формирования ключа – функция, формирующая один или несколько секретных ключей на основе данных **источника начального ключевого материала** с помощью псевдослучайной функции. Для формирования ключей к секретным данным источника начального ключевого материала могут добавляться несекретные.

Назначение

- для создания ключа необходимой длины;
- для создания ключа заданного формата.

Примеры источников начального ключевого материала

Функция формирования ключа

Функция формирования ключа (KDF - key derivation function)

Функция формирования ключа – функция, формирующая один или несколько секретных ключей на основе данных **источника начального ключевого материала** с помощью псевдослучайной функции. Для формирования ключей к секретным данным источника начального ключевого материала могут добавляться несекретные.

Назначение

- для создания ключа необходимой длины;
- для создания ключа заданного формата.

Примеры источников начального ключевого материала

- Несовершенный генератор случайных чисел.

Функция формирования ключа

Функция формирования ключа (KDF - key derivation function)

Функция формирования ключа – функция, формирующая один или несколько секретных ключей на основе данных **источника начального ключевого материала** с помощью псевдослучайной функции. Для формирования ключей к секретным данным источника начального ключевого материала могут добавляться несекретные.

Назначение

- для создания ключа необходимой длины;
- для создания ключа заданного формата.

Примеры источников начального ключевого материала

- Несовершенный генератор случайных чисел.
- Движение мышью, задержка между нажатиями клавиш конкретного пользователя.

Функция формирования ключа

Функция формирования ключа (KDF - key derivation function)

Функция формирования ключа – функция, формирующая один или несколько секретных ключей на основе данных **источника начального ключевого материала** с помощью псевдослучайной функции. Для формирования ключей к секретным данным источника начального ключевого материала могут добавляться несекретные.

Назначение

- для создания ключа необходимой длины;
- для создания ключа заданного формата.

Примеры источников начального ключевого материала

- Несовершенный генератор случайных чисел.
- Движение мышью, задержка между нажатиями клавиш конкретного пользователя.
- Ключи, сгенерированные в протоколе Диффи-Хэллмана.

Функция формирования ключа. Схема Extract-then-Expand

Общая схема функции формирования ключа по принципу Extract-then-Expand

$$PRK = \text{XTR}(XTS, SKM) \leftarrow Extract$$
$$KM = \text{PRF*}(PRK, CTXInfo, L) \leftarrow Expand$$

- SKM — начальный ключевой материал;
- XTS — «соль», может отсутствовать;
- XTR — функция получения начальных «почти идеальных случайных данных» PRK;
- PRF* — псевдослучайная функция с варьируемой длиной (типа ГПСЧ);
- CTXInfo — информация о контексте использования ключа: алгоритм, приложение, NONCE, время и т.п.
- L – длина ключа, который формируется.

HKDF

$$\text{HKDF}(XTS, SKM, CTXInfo, L) = K(1) \parallel K(2) \parallel ... \parallel K(t),$$

где

$$\begin{aligned} PRK &= \text{HMAC}(XTS, SKM) \\ K(1) &= \text{HMAC}(PRK, CTXInfo \parallel 0) \\ K(i+1) &= \text{HMAC}(PRK, K(i) \parallel CTXInfo \parallel i), 1 \leqslant i \leqslant t \\ t &= \lceil L/k \rceil, \\ k &= |\text{HMAC}|(\text{длина хэш-значения}) \end{aligned}$$

Из блока $K(t)$ берется столько битов, чтобы совокупная длина выхода была L.

HKDF

$$\text{HKDF}(XTS, SKM, CTXInfo, L) = K(1) \parallel K(2) \parallel ... \parallel K(t),$$

где

$$\begin{aligned} PRK &= \text{HMAC}(XTS, SKM) \\ K(1) &= \text{HMAC}(PRK, CTXInfo \parallel 0) \\ K(i+1) &= \text{HMAC}(PRK, K(i) \parallel CTXInfo \parallel i), 1 \leqslant i \leqslant t \\ t &= \lceil L/k \rceil, \\ k &= |\text{HMAC}|(\text{длина хэш-значения}) \end{aligned}$$

Из блока $K(t)$ берется столько битов, чтобы совокупная длина выхода была L.

Напоминание

$$\text{HMAC}(K_{out} \parallel K_{in}, M) = H(K_{out} \parallel H(K_{in} \parallel M)),$$

H – любая криптографическая хэш-функция.

Заключение

Спасибо за внимание!