Протокол доставки ключей «БУРУНДУК»

Виталий Кирюхин

ООО «СФБ Лаб»

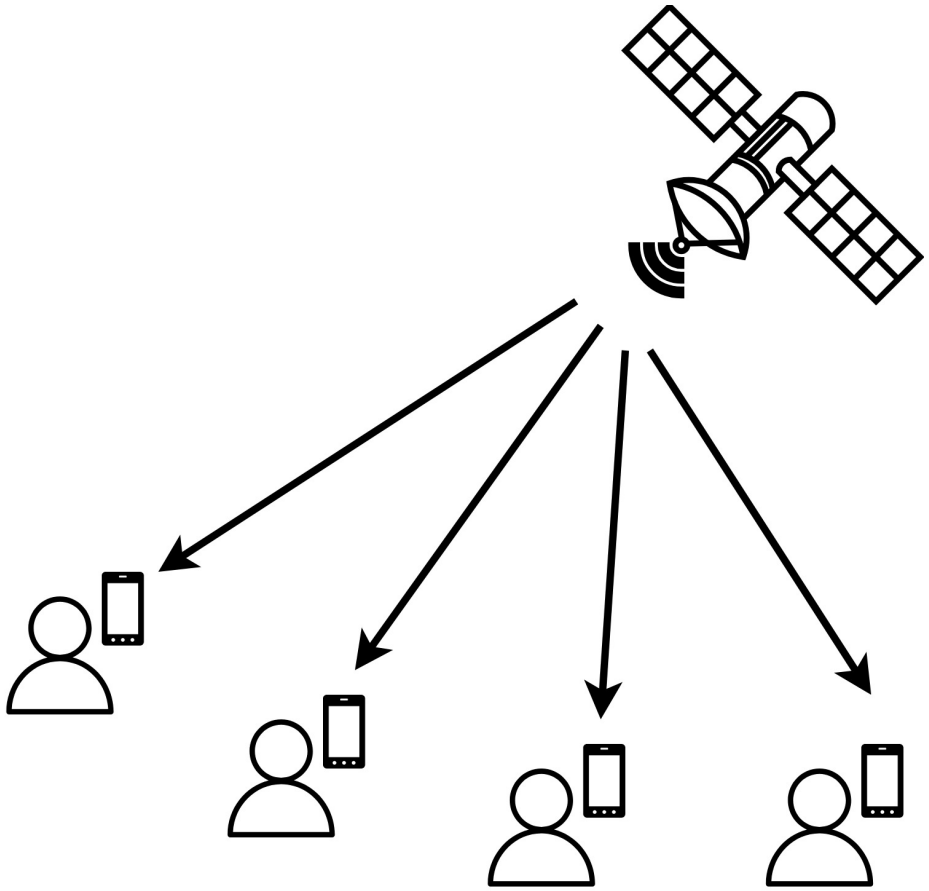
РусКрипто’2026
26 марта 2026

vitaly.kiryukhin@sfblaboratory.ru

Условия задачи

Широковещательный канал

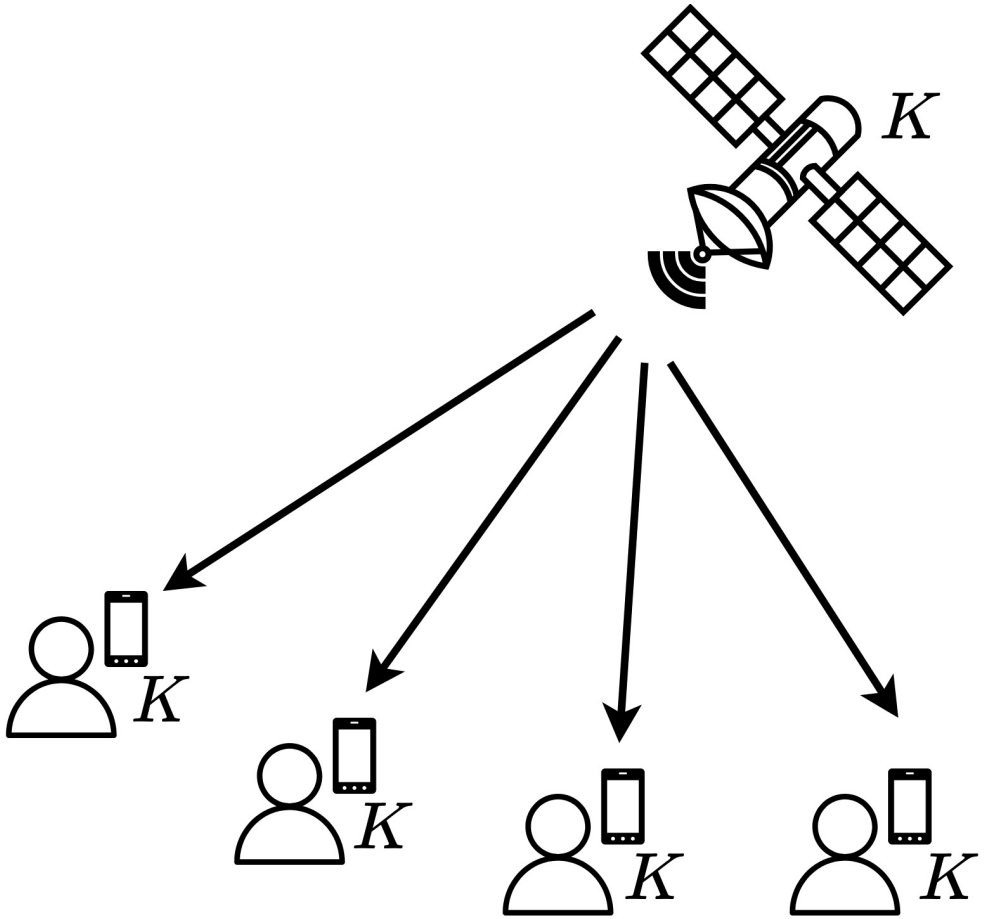
– от одного источника ко многим потребителям.



Условия задачи

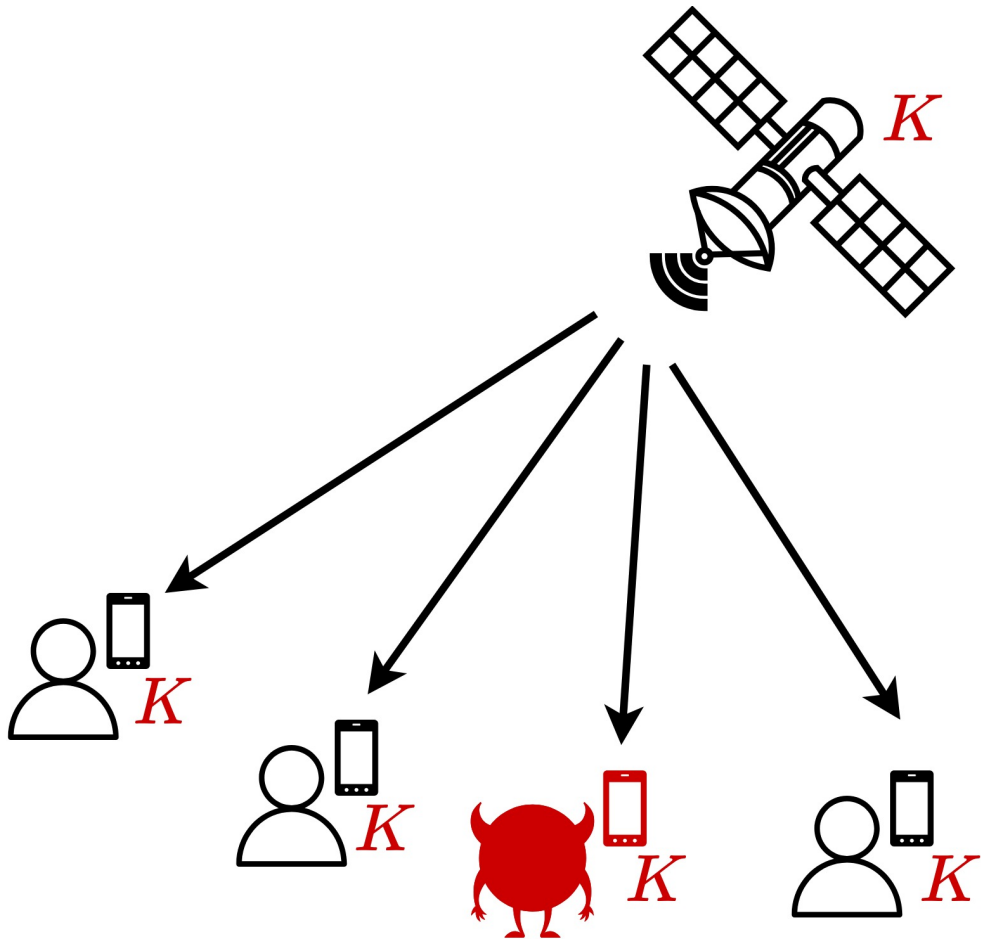
Защита **конфиденциальности** канала

требует **одного общего** ключа K.



Условия задачи

Компрометация одного пользователя – раскрытие K – компрометация всего канала.



Все авторизованные пользователи должны получить **новый** K'.

Условия задачи

СФБ ЛАБ

Альтернативная формулировка

Путём доставки действующего ключа должны дать доступ к широковещательному сигналу для произвольной *части* пользователей, а остальных наоборот отключить от него.

Ограничения

Часть пользователей может получить новый ключ с помощью:

- доверенной доставки;
- интерактивного взаимодействия с ключевым центром;

Ограничения

Часть пользователей может получить новый ключ с помощью:

- доверенной доставки;
- интерактивного взаимодействия с ключевым центром;

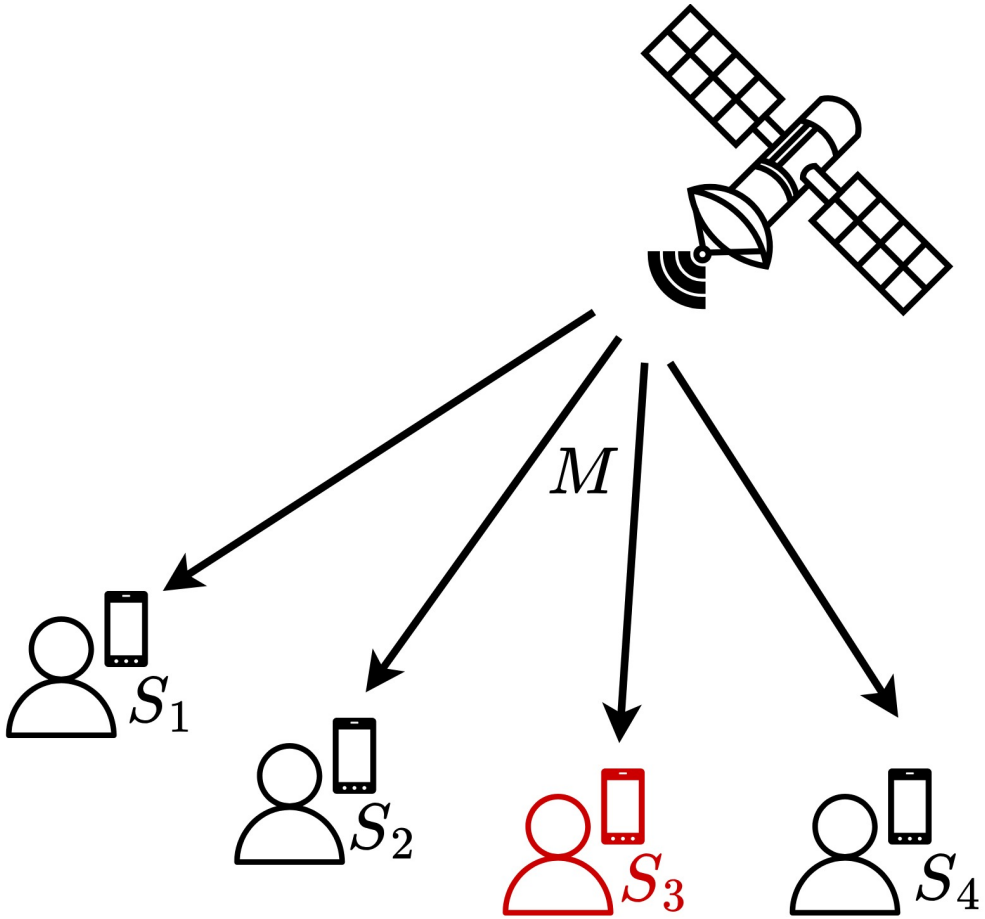
но существует множество тех, которые:

- имеют **доступ только к широковещательному каналу** и
- подключаются к нему время от времени.

Тривиальное решение

Каждое устройство исходно получает личный ключ S_i.

Раскрытие S_3 не приводит к раскрытию S_1, S_2, S_4.

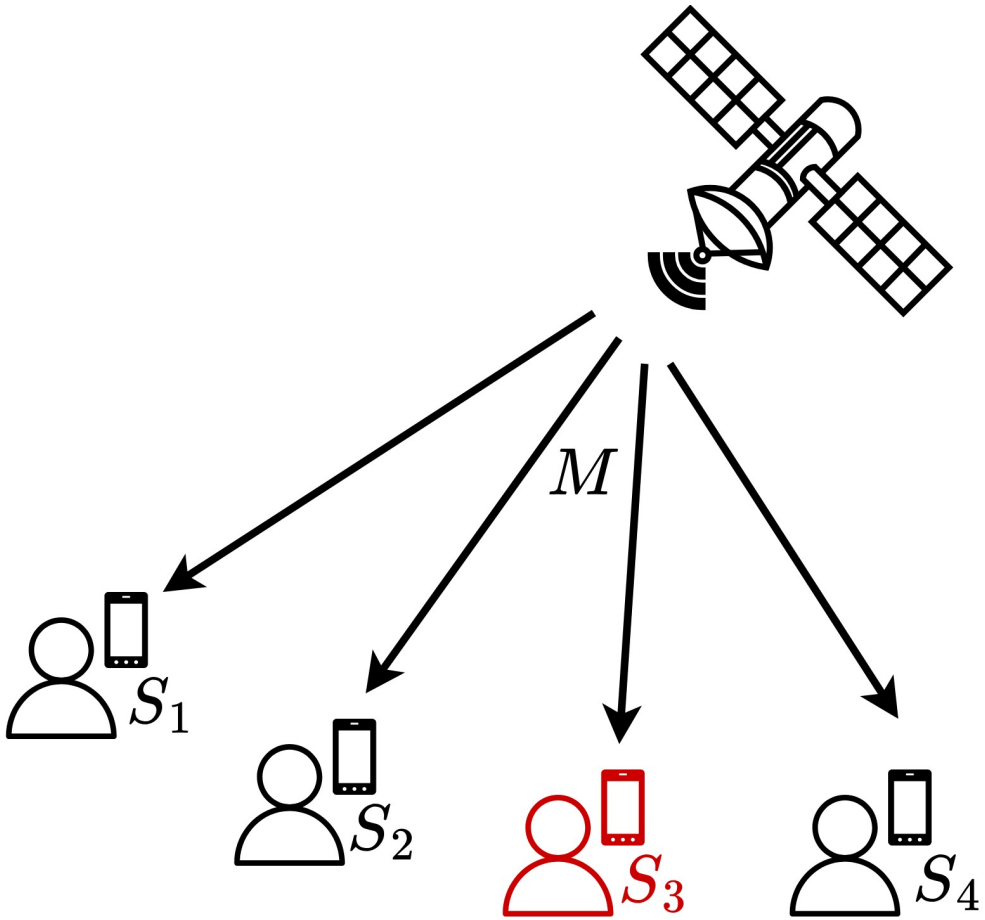


Тривиальное решение

Для доставки нового ключа K', рассылается сообщение

$$M = (1, \mathrm{enc}(S_1, K')),\ (2, \mathrm{enc}(S_2, K')),\ (4, \mathrm{enc}(S_4, K')).$$

Авторизованные пользователи получают K', а противник – нет.



Тривиальное решение

СФБ ЛАБ

Проблема

Формально все требования выполнены,
но **нагрузка на канал связи огромна** –
линейно зависит от числа действующих пользователей.

Протокол «БУРУНДУК»

Протокол

СФБ ЛАБ

БУРУНДУК

Быстрая **Р**оссийская **Н**еинтерактивная **Д**оставка **К**лючей

BURUNDUK

Broadcast **R**ussian **N**oninteractive **D**elivery of **K**eys

Задачи синтеза

- Минимизация нагрузки на канал связи

Задачи синтеза

- Минимизация нагрузки на канал связи
- Приемлемые требования к памяти устройства (килобайты)

Задачи синтеза

СФБ ЛАБ

- Минимизация нагрузки на канал связи
- Приемлемые требования к памяти устройства (килобайты)
- Стойкость при компрометации любого числа пользователей

Задачи синтеза

- Минимизация нагрузки на канал связи
- Приемлемые требования к памяти устройства (килобайты)
- Стойкость при компрометации любого числа пользователей
- Произвольное количество «доставок ключей»

Задачи синтеза

- Минимизация нагрузки на канал связи
- Приемлемые требования к памяти устройства (килобайты)
- Стойкость при компрометации любого числа пользователей
- Произвольное количество «доставок ключей»
- Симметричные криптоалгоритмы в основе

Задачи синтеза

- Минимизация нагрузки на канал связи
- Приемлемые требования к памяти устройства (килобайты)
- Стойкость при компрометации любого числа пользователей
- Произвольное количество «доставок ключей»
- Симметричные криптоалгоритмы в основе
- Отсутствие требований к обновлению состояния устройств, к постоянному приёму информации из канала связи

Характеристики

Нагрузка на канал связи

Линейная зависимость от числа **r** скомпрометированных сущностей (пользователей или групп пользователей)

$$|M| = c \cdot \mathbf{r} \cdot (|K| + \log_2(N)), \quad c \leq 2.$$

Пример

$N \approx 1$ миллиард пользователей

$r \approx 100$ скомпрометированных пользователей/групп

$|K| = 128$ бит – длина доставляемого ключа

	БУРУНДУК	Тривиальный способ
$\|M\|$	2400 байт	24 гигабайта

Основа – Broadcast Encryption

СФБ ЛАБ

Naor D., Naor M., Lotspiech J.

Revocation and Tracing Schemes for Stateless Receivers

CRYPTO 2001

Bhattacherjee S., Sarkar P.

Tree Based Symmetric Key Broadcast Encryption

2013

Bhattacherjee S., Sarkar P.

Reducing Communication Overhead

of the Subset Difference Scheme

2014

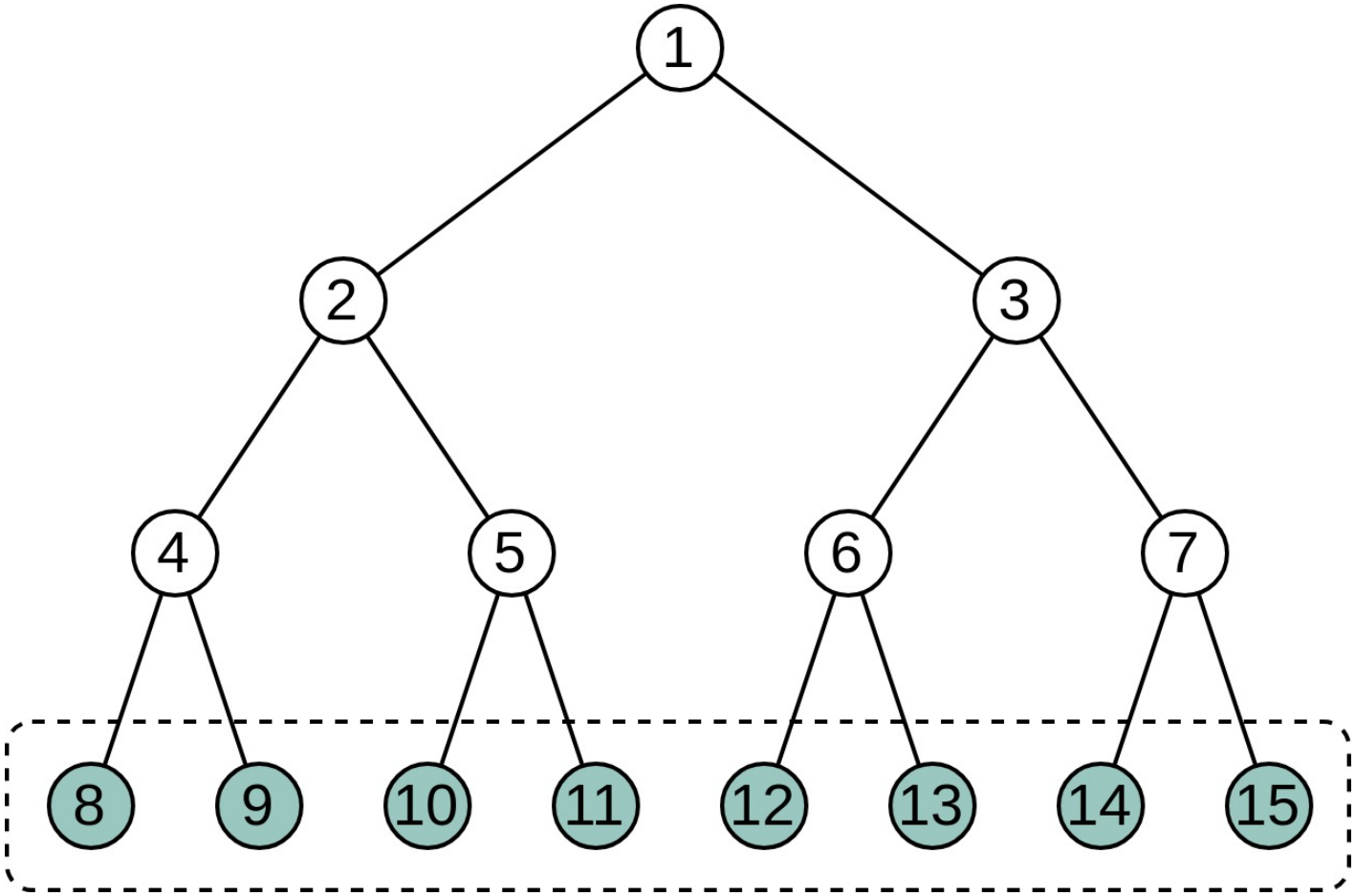
Идея

СФБ ЛАБ

- Каждое пользовательское устройство получает при изготовлении и хранит неизменяемый набор ключей.
- Наборы ключей у разных пользователей могут пересекаться.

Идея

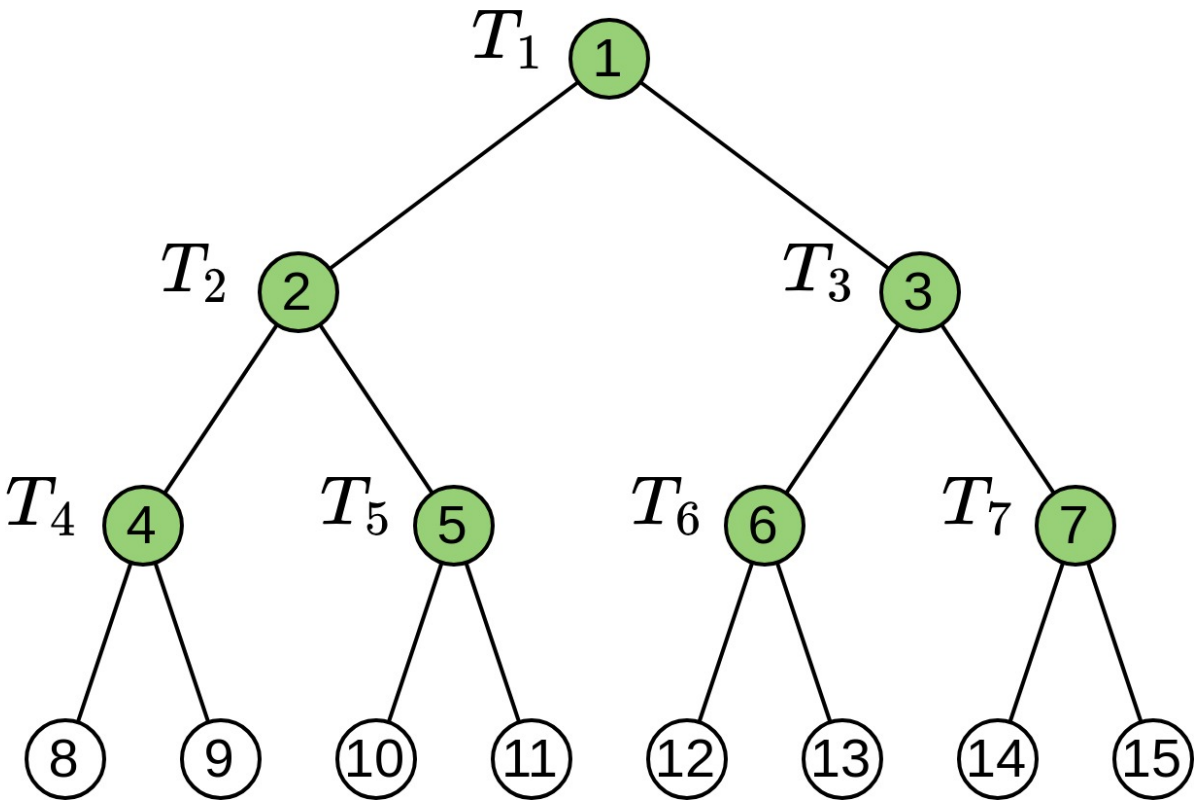
Соотнесём пользователей с листьями двоичного дерева.



Пример: $N = 8$ пользователей, дерево высоты $h = 3$ из 15 узлов.

Идея

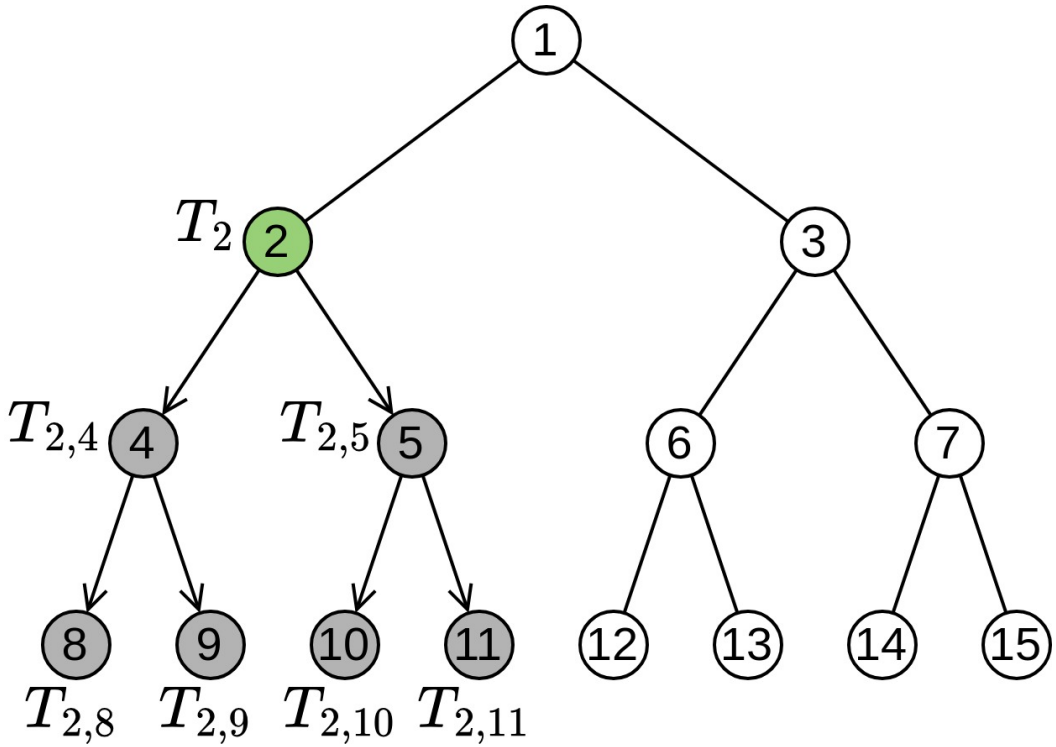
Для каждого *внутреннего* i-го узла дерева независимо сгенерируем (псевдо)случайный ключ T_i.



Пример: 7 случайных ключей.

Идея

Для каждого ключа T_i с помощью PRF
сгенерируем поддерево производных.
Ключ $T_{i,j}$ вычислен для j-го узла с помощью T_i, j – потомок i.



Пример: ключ T_2 порождает 6 производных ключей.

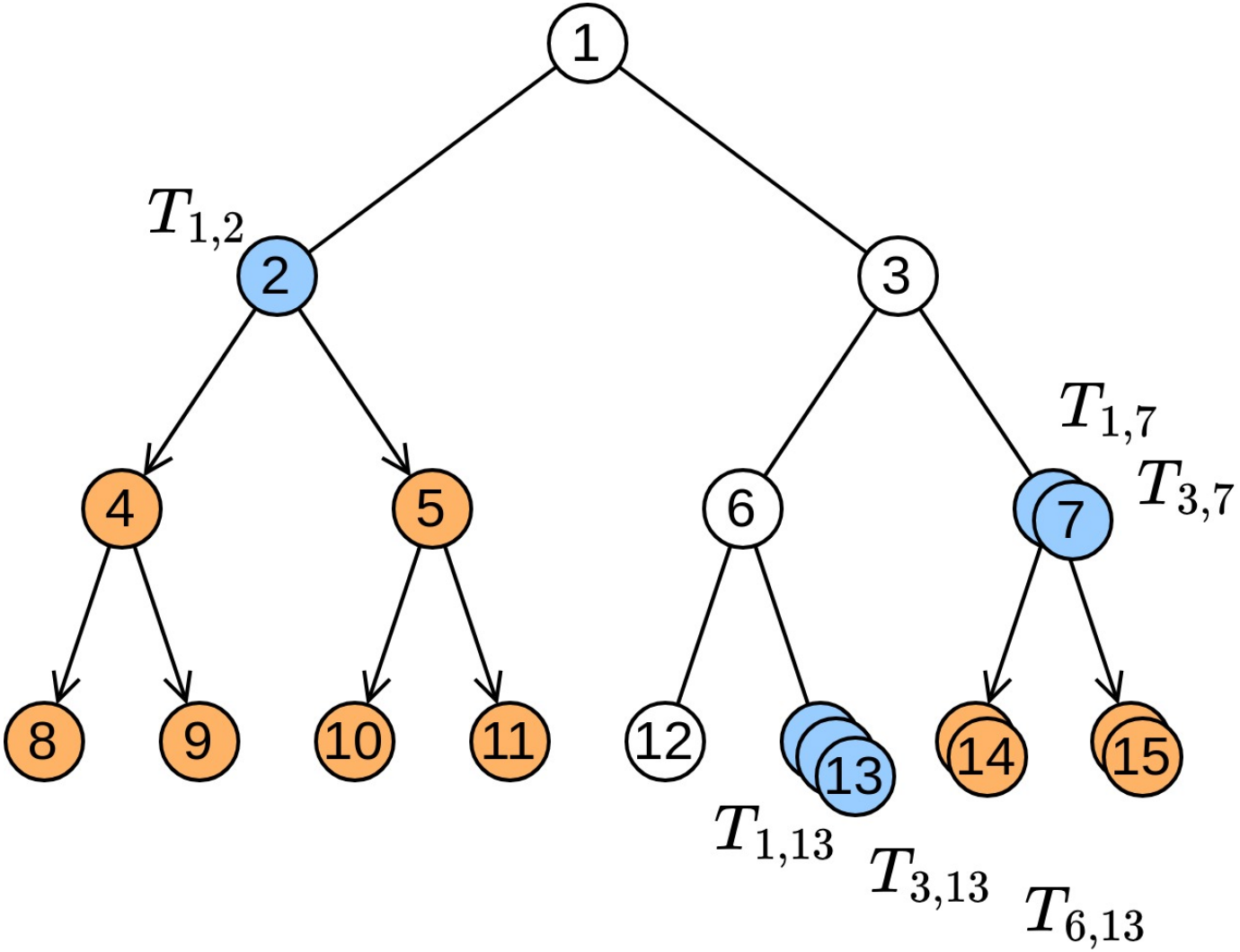
Идея

Инвертируем типичную логику присваивания ключей.

Пользователь:
– НЕ получает ключи, сопоставленные листу или узлу-предку;
– получает остальные ключи, порождённые от узлов-предков.

Идея

Пример: пользователь (лист 12) хранит 6 ключей, может вычислить ещё 10.



Идея

СФБ ЛАБ

Общее правило формирования M

- Закрашиваем листья – r отключаемых пользователей
- Закрашиваем пути от этих листьев к корню
- В закрашенном поддереве ищем цепочки (i, j):
  - i – внутренний узел, имеющий ровно одного потомка
  - j – лист или узел, имеющий ровно двух потомков
- Используем $T_{i,j}$ для защиты доставляемого ключа

Первый закрашенный лист порождает одну цепочку.

Каждый последующий – добавляет не более двух.

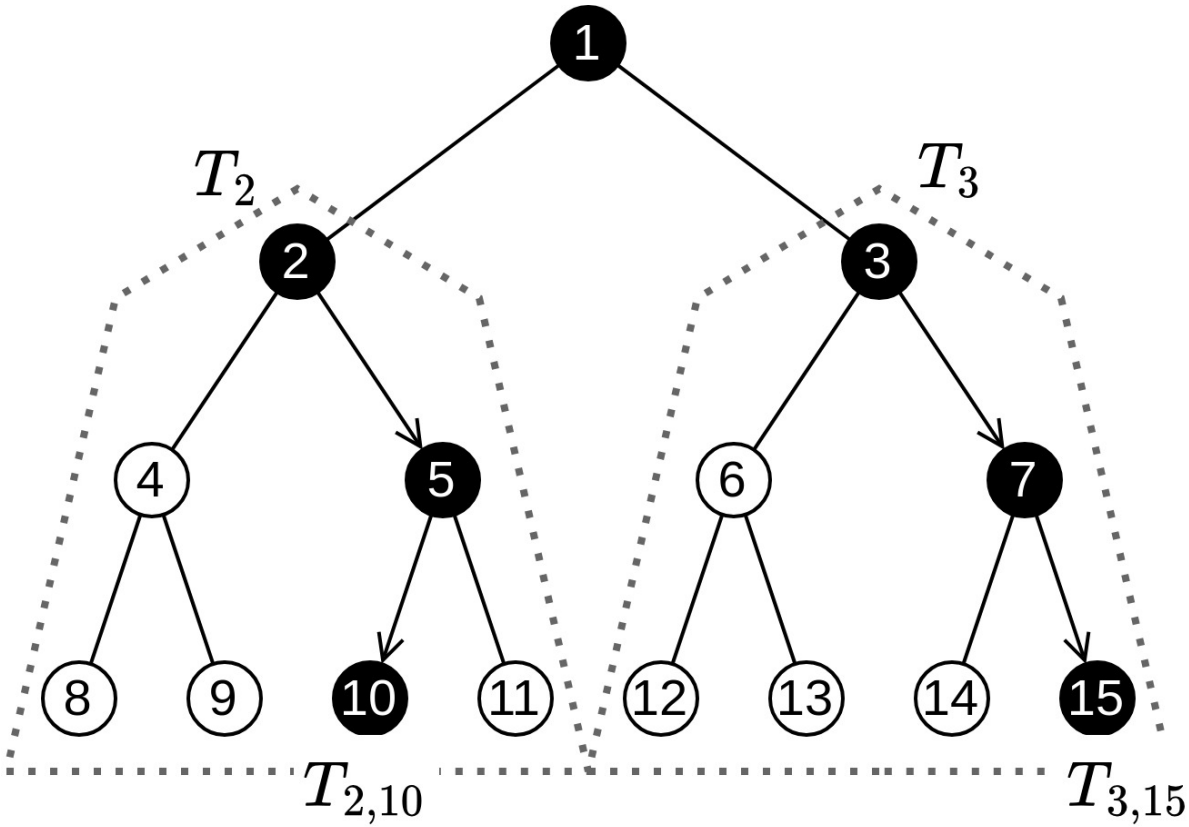
Идея

Пример: $r = 2$, скомпрометированы 10 и 15.

Для защиты доставляемого ключа используем:

– $T_{2,10}$ – его могут вычислить 8, 9, 11;

– $T_{3,15}$ – его могут вычислить 12, 13, 14.



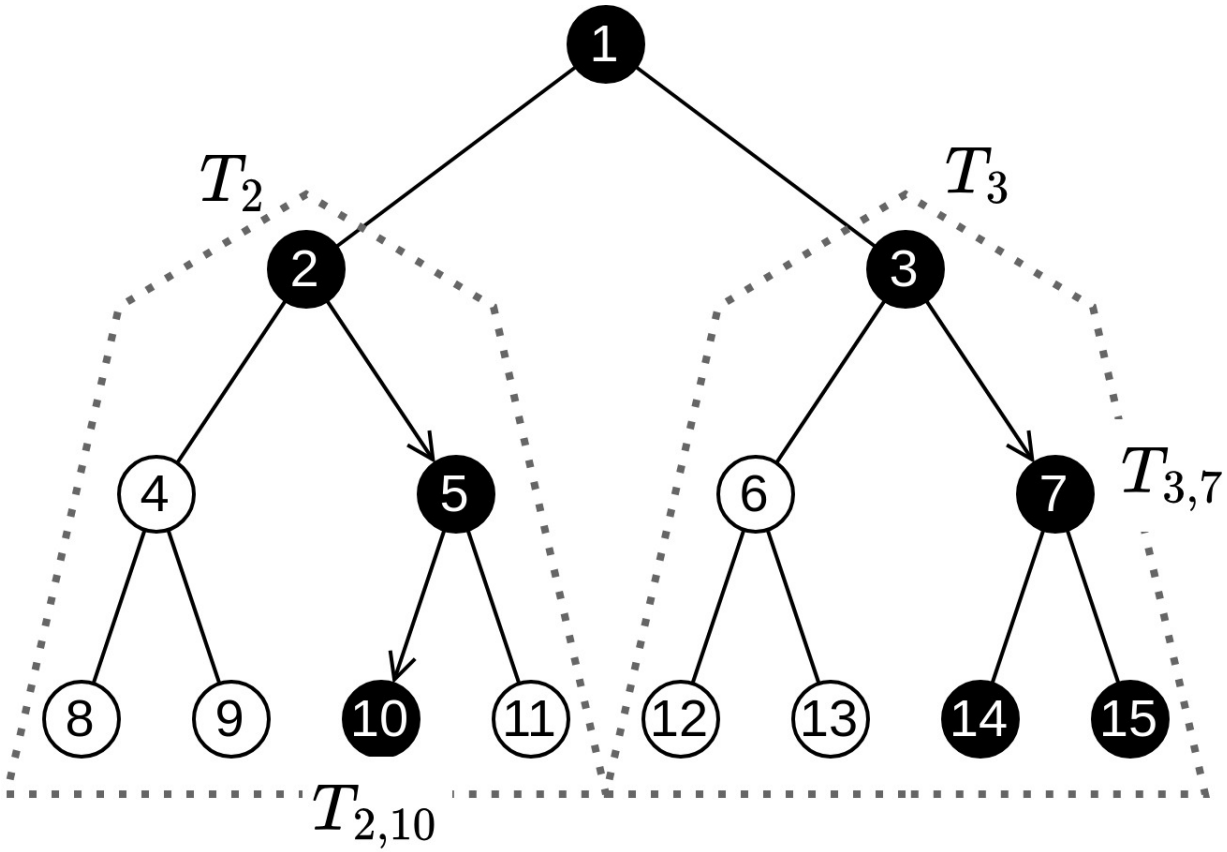
Идея

Пример: $r = 3$, скомпрометированы 10, 15, 14.

Для защиты доставляемого ключа используем:

– $T_{2,10}$ – его могут вычислить 8, 9, 11;

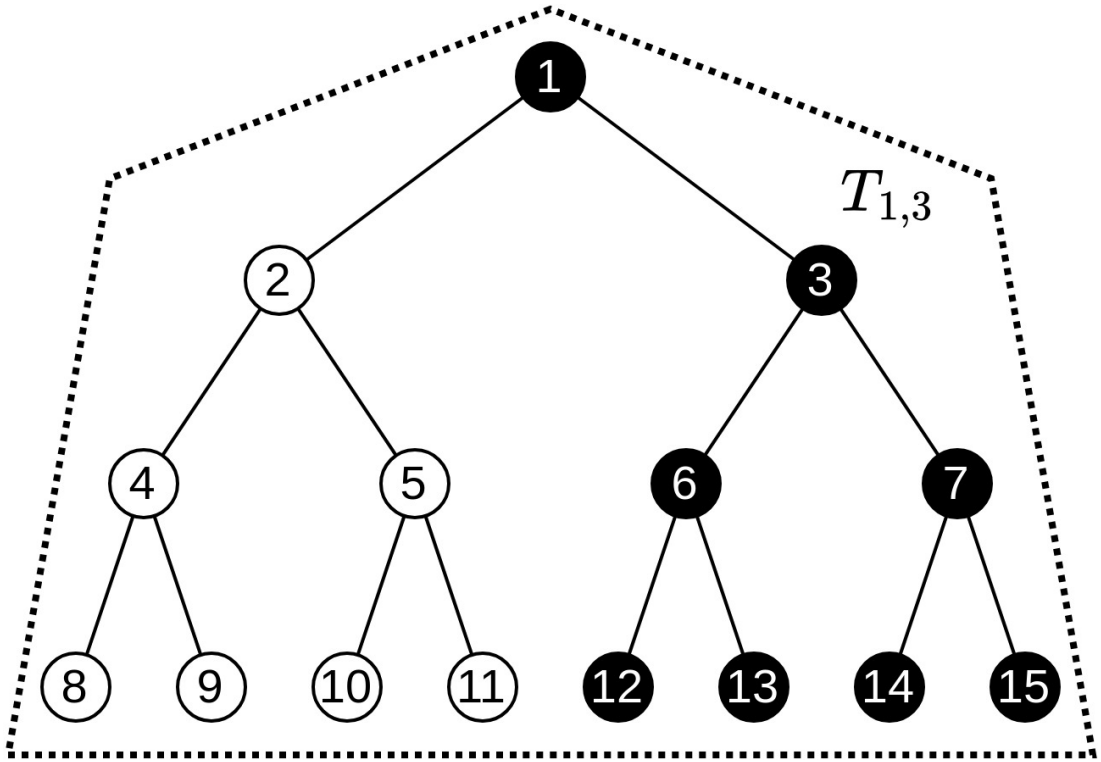
– $T_{3,7}$ – его могут вычислить 12, 13.



Полезные свойства

Пользователи и группы

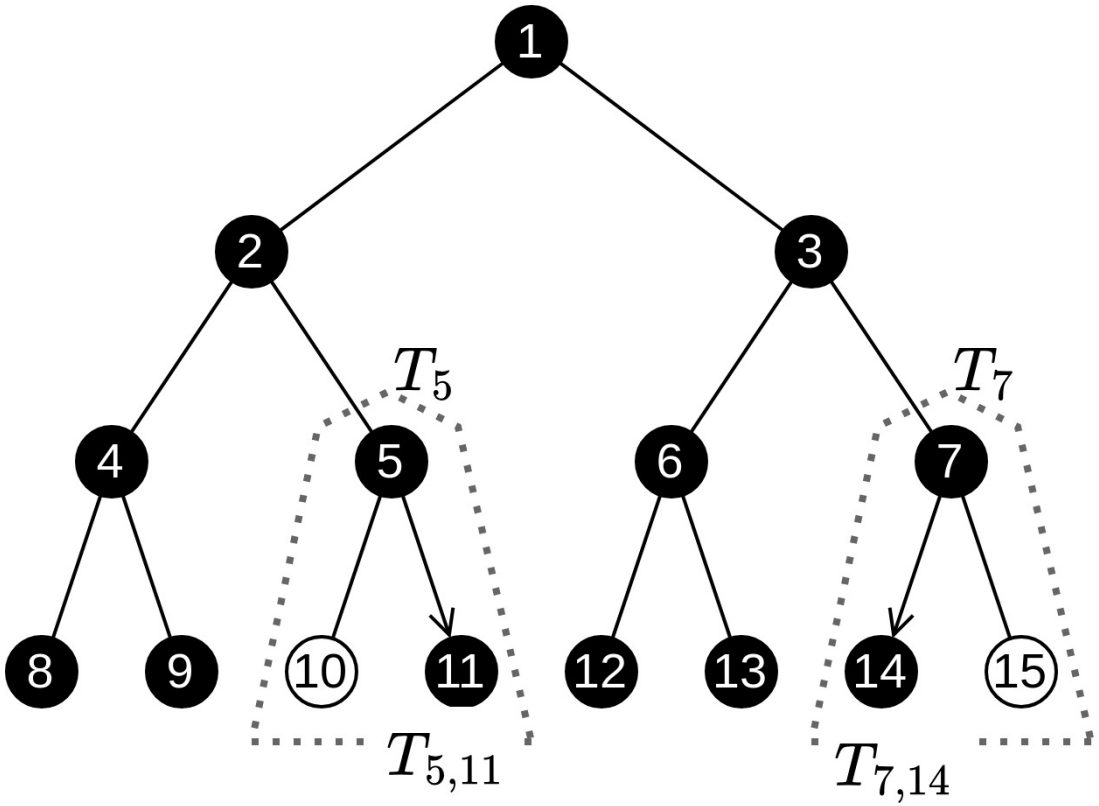
Отключение группы пользователей (полного поддерева) увеличивает нагрузку на канал связи также, как отключение одного пользователя.



Полезные свойства

Инверсия

Случаи «скомпрометировано r пользователей» и «**не**скомпрометировано r пользователей» обрабатываются с одинаковой эффективностью.

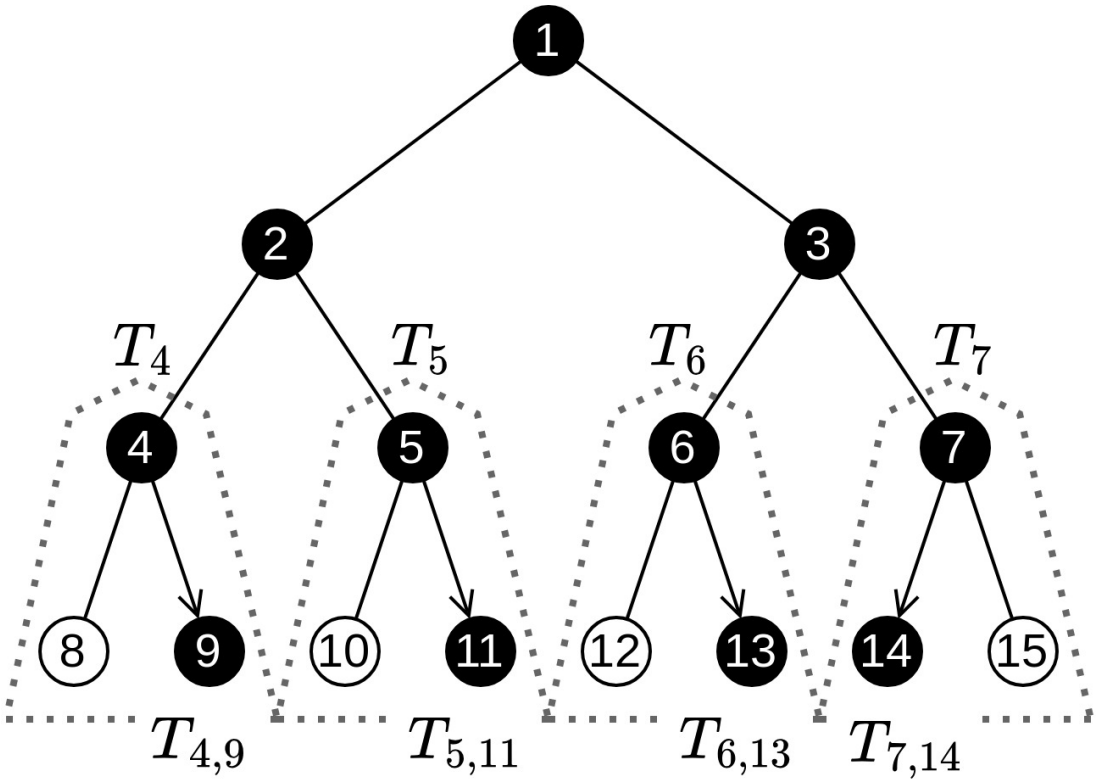


Полезные свойства

СФБ ЛАБ

Худший случай (двоичное дерево)

Скомпрометировано $r = N/2$ случайных пользователей.
Нагрузка на канал почти как у тривиального способа.



Уходим от худшего случая: q-арные деревья

СФБ ЛАБ

Критерий

По умолчанию $q = 2$ и предполагается, что $r \ll N$.

С ростом доли r/N скомпрометированных лучше выбрать $q > 2$.

Пример. $q = 8$ лучше, чем $q = 2$, начиная с $r/N \approx 5\%$.

Снижается предельная нагрузка на канал связи

$$|M| \leq (N/q) \cdot (|K| + \log_2(N) + \log_2(q)).$$

Растут требования к памяти

$N \approx 1$ миллиард пользователей.

При $q = 2$ в памяти устройства 465 ключей (15 килобайт).

При $q = 8$ в памяти устройства 28105 ключей (1 мегабайт).

Уходим от худшего случая: q-арные деревья

СФБ ЛАБ

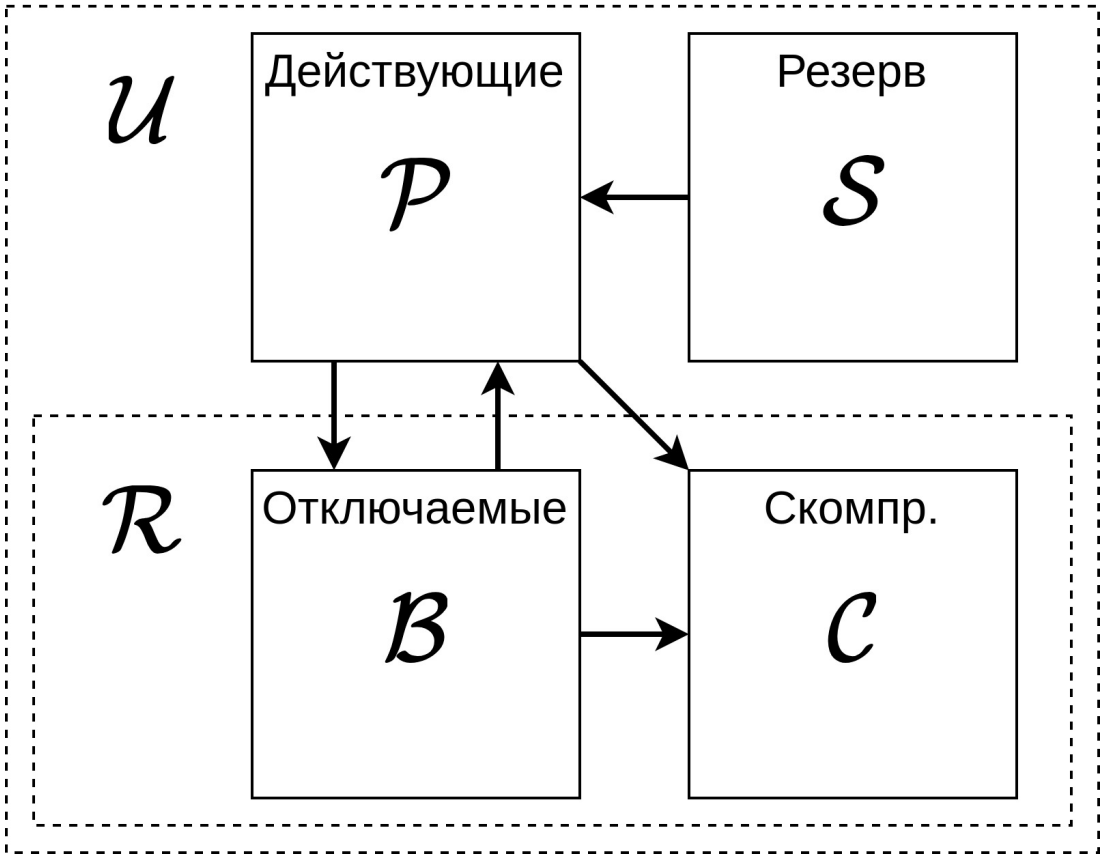
Гибридный подход

При

- наличии памяти для хранения ключей и
- ожидаемой высокой доле скомпрометированных/отключаемых пользователей

храним двоичное дерево и q-арное дерево $q \in \{8, 16\}$.
Для доставки ключа выбираем $q \in \{2, 8, 16\}$,
минимизирующее нагрузку на канал связи.

Статусы пользователей



Протокол должен обеспечить следующее:

- *действующие* получают ключ;
- *отключаемые* и *скомпромет.* не получают ключ;
- *зарезервированные* получают или не получают ключ.

Добавление пользователей

СФБ ЛАБ

Исходное состояние

Одно дерево – 2^h листов

– 2^h зарезервированных пользователей.

Что делать при исчерпании?

Добавление пользователей

СФБ ЛАБ

Исходное состояние

Одно дерево – 2^h листов

– 2^h зарезервированных пользователей.

Что делать при исчерпании?

Способ 1

Сразу выбираем высоту дерева h с большим запасом.

- Незначительно увеличиваем требования к памяти.
- Не увеличиваем нагрузку на канал связи.

Добавление пользователей

СФБ ЛАБ

Наблюдение

Пусть в каждом поддереве из 2^u узлов
есть отключаемый/скомпр. пользователь.
Тогда одинаково эффективны:
– одно дерево из 2^{v+u} элементов;
– 2^v разных деревьев из 2^u элементов.

Добавление пользователей

СФБ ЛАБ

Способ 2. Несколько деревьев

По исчерпании первого дерева,
создаём ещё одно и увеличиваем резерв.

Добавление пользователей

СФБ ЛАБ

Способ 2. Несколько деревьев

По исчерпании первого дерева,
создаём ещё одно и увеличиваем резерв.

Способ 3. Непосредственное резервирование

На каждом уровне, высота которого кратна h', оставляем долю β поддеревьев в качестве зарезервированных.
Можем добавлять пользователей, поддерживая иерархичный принцип группировки.

Аутентификация ключей

Имитозащита

Пользователи не получают дополнительных ключей.

Каждая часть сообщения имитозащищается,

$$(i, j), \mathrm{encmac}(T_{i,j}, K')$$

Нестойкое решение!

Противник может скомпрометировать $T_{i,j}$
и навязать K' пользователям, которым известен $T_{i,j}$.

Аутентификация ключей

СФБ ЛАБ

Имитозащита

Пользователи не получают дополнительных ключей.

Каждая часть сообщения имитозащищается,

$$(i, j), \mathrm{encmac}(T_{i,j}, K')$$

Нестойкое решение!

Противник может скомпрометировать $T_{i,j}$
и навязать K' пользователям, которым известен $T_{i,j}$.

Электронная подпись

Каждый пользователь получает ключ проверки подписи.

Передаётся пара «сообщение M, подпись $\mathrm{sig}(M)$».

Простое и гибкое решение, но рост нагрузки на канал связи.

Аутентификация ключей

СФБ ЛАБ

Цепочки хэшей

Доставляемые ключи генерируются заранее

$$K_\ell \xrightarrow{H} K_{\ell-1} \xrightarrow{H} \ldots \to K_2 \xrightarrow{H} K_1.$$

Пользователи исходно получают ключ K_1.
Доставляемый ключ аутентифицируется путём проверки принадлежности к цепочке, $H(K_2) \stackrel{?}{=} K_1$.

- Требуется регулярная передача сообщений.
- Срок жизни устройств ограничен числом/длиной цепочки.
- Не увеличивается нагрузка на канал связи.

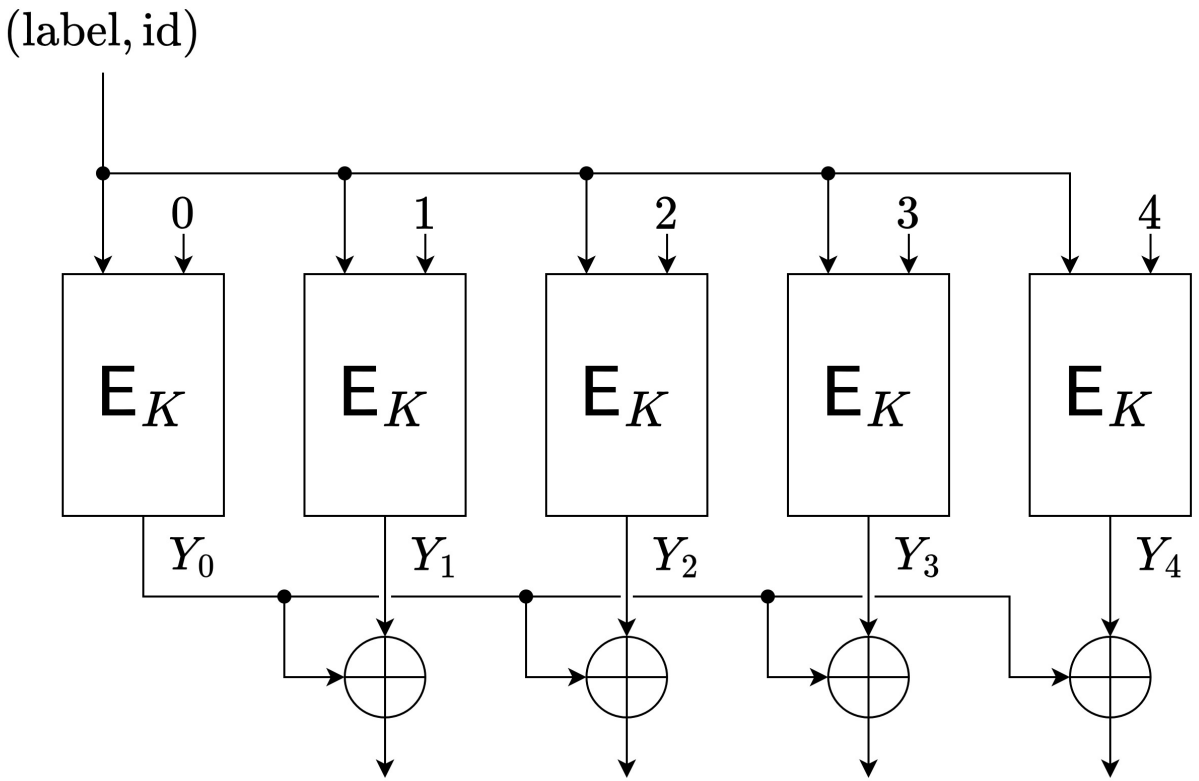
Замечания об аутентификации ключей

СФБ ЛАБ

- Доставляемый ключ предназначен для использования в некоторый период времени.
- Пользователь должен с достаточной точностью определять корректное время.
- При сбое часов пользователя противник получает возможность навязывать ранее рассылаемые сообщения – «старые» ключи.

Инстанцирование

Для ключевых деревьев и шифрования доставляемых ключей нужна PRF – используем шифр $E : V^k \times V^n \to V^n$ по схеме XORP.



XORP позволяет формировать
на одном ключе $\approx 2^n$, а не $2^{n/2}$ блоков выхода.

Инстанцирование

(label, id):

1) "gen", i – ключ T_i из мастер-ключа;
2) "left", (i, j) – «левый» производный ключ из $T_{i,j}$;
3) "right", (i, j) – «правый» производный ключ из $T_{i,j}$;
4) "enc", id(K') – шифрование ключа K' на $T_{i,j}$;
5) "chain", id(K') – формирование цепочек хэшей.

Замечание 1: $T_{i,j}$ используется
и для формирования производных в дереве,
и для шифрования доставляемого ключа K'.

Замечание 2: Для аутентификации с помощью цепочек хэшей недостаточно PRF-стойкости. Достаточной является стойкость к «поиску эквивалентного ключа» или в модели «идеального шифра».

Заключение

Заключение

СФБ ЛАБ

1. Протокол «БУРУНДУК» – эффективная доставка ключей по широковещательному каналу связи.

Заключение

1. Протокол «БУРУНДУК» – эффективная доставка ключей по широковещательному каналу связи.
2. Нагрузка на канал – линейная по числу скомпрометированных пользователей/групп.

Заключение

1. Протокол «БУРУНДУК» – эффективная доставка ключей по широковещательному каналу связи.
2. Нагрузка на канал – линейная по числу скомпрометированных пользователей/групп.
3. Требования к памяти устройств – логарифмические по числу пользователей.

Заключение

1. Протокол «БУРУНДУК» – эффективная доставка ключей по широковещательному каналу связи.
2. Нагрузка на канал – линейная по числу скомпрометированных пользователей/групп.
3. Требования к памяти устройств – логарифмические по числу пользователей.
4. Основа – PRF – блочный шифр в схеме XORP.
5. Аутентификация ключей – цепочки хэшей или подпись.

Благодарю за внимание!

Виталий Кирюхин

ООО «СФБ Лаб»

РусКрипто’2026

26 марта 2026

vitaly.kiryukhin@sfblaboratory.ru