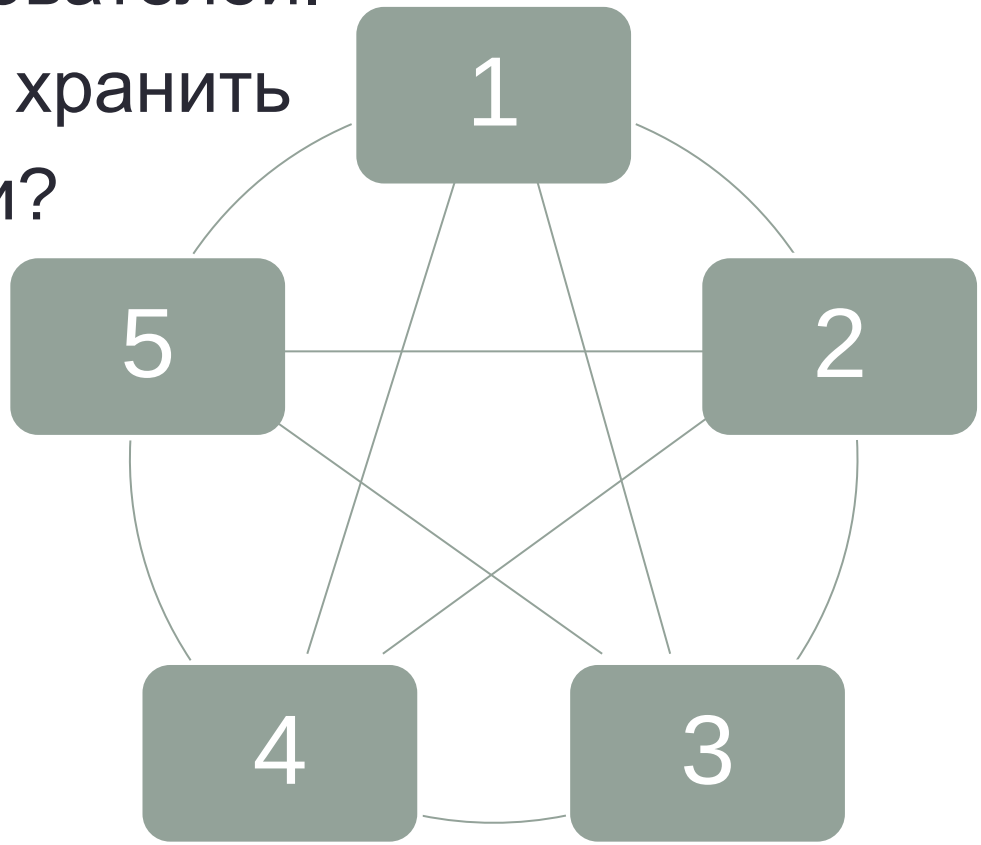
УПРАВЛЕНИЕ КЛЮЧАМИ

Протоколы

- Протокол передачи ключа (key transport): Алиса секретно передает Бобу конкретный выбранный Алисой ключ K.
- Протокол согласования ключа (key agreement): Алиса и Боб договариваются о каком-нибудь общем ключе.
- Протокол обновления ключа (key update): у Алисы и Боба уже есть секретный ключ, но они для новых сессий используют свежие разные ключи.
- Протокол раздачи ключей (key distribution): Центр раздает резидентам ключи, при помощи которых можно общаться друг с другом и с Центром.

Попарные ключи

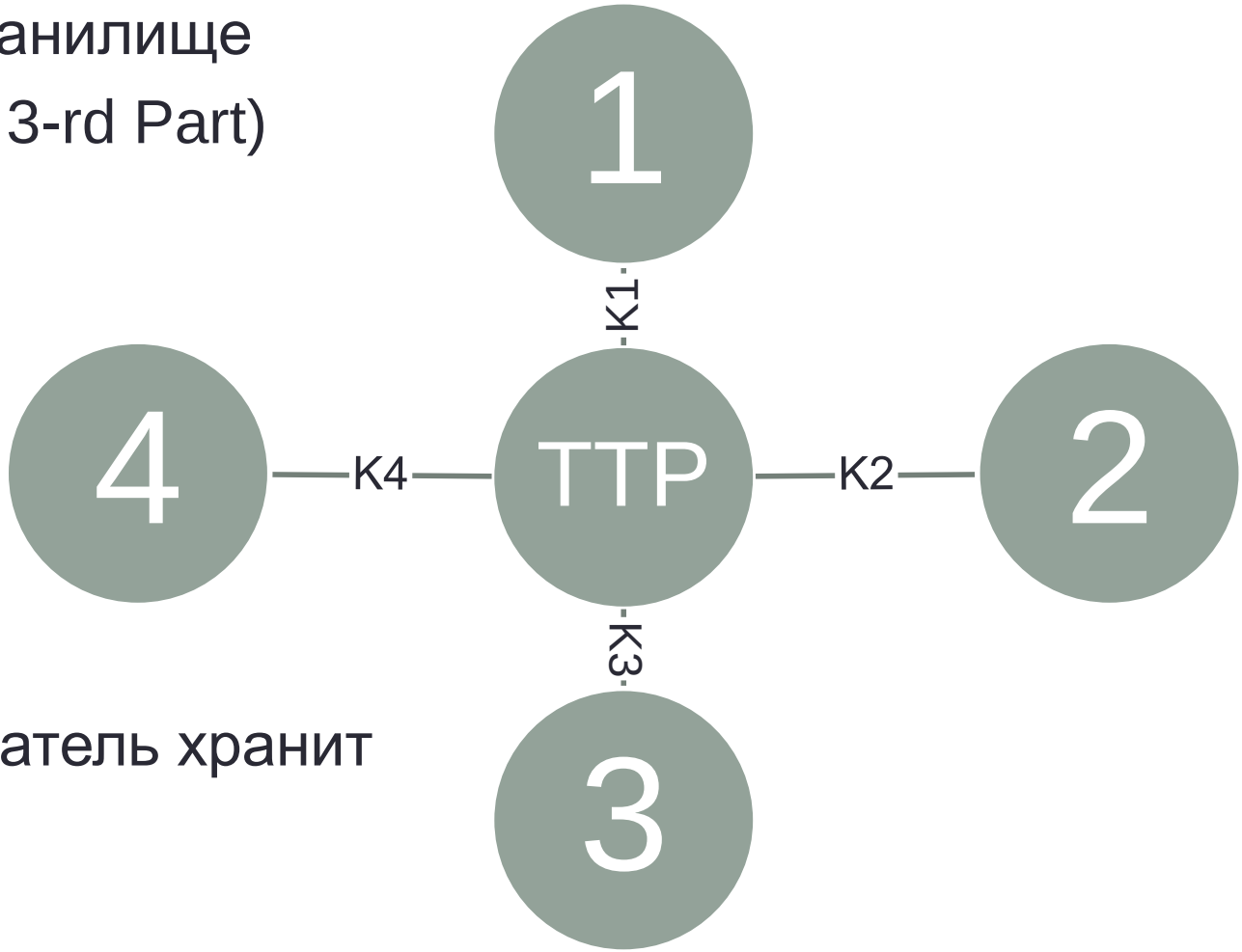
- В сети n пользователей.
- Проблема: Как хранить попарные ключи?



Количество ключей: O(n) у каждого

Решение проблемы

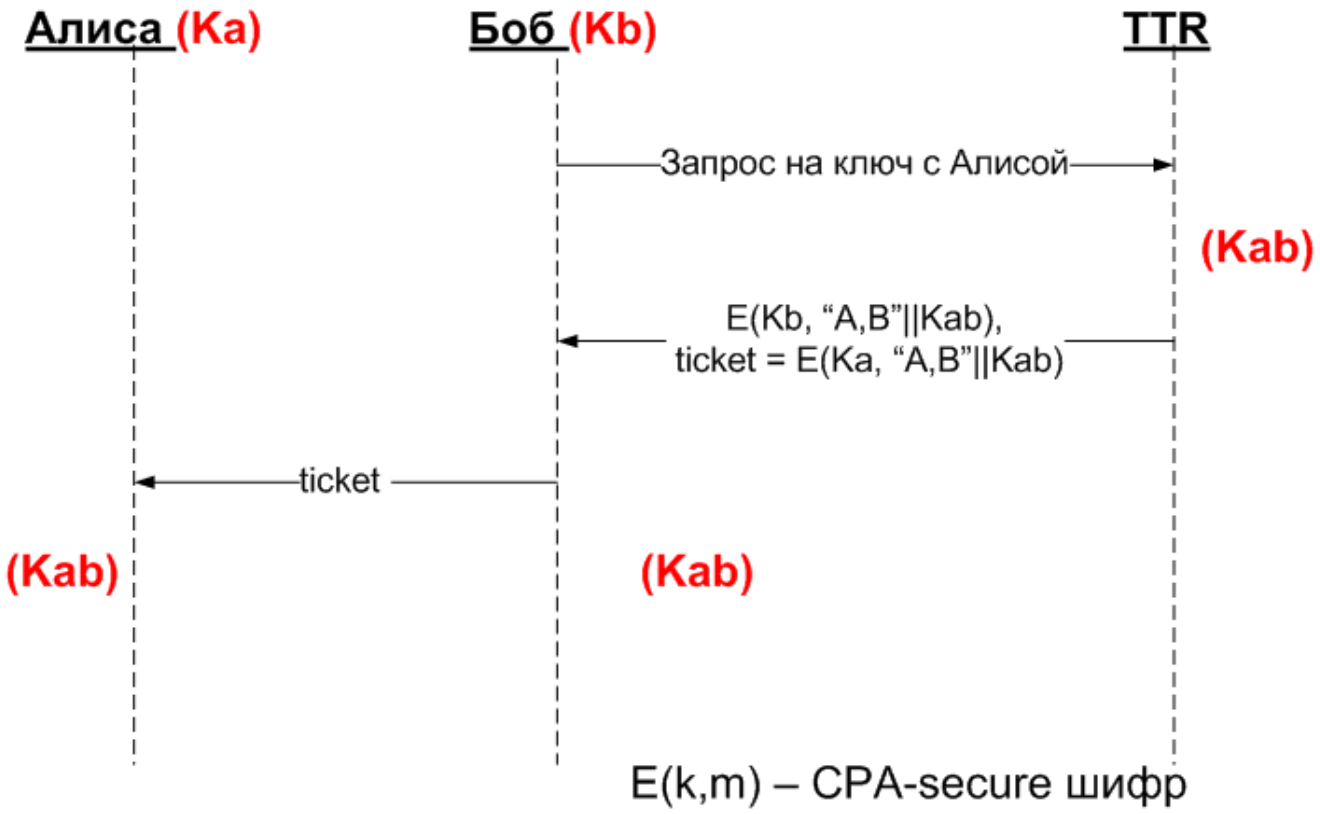
- Доверенное хранилище Ключей (Trusted 3-rd Part)



Каждый пользователь хранит только 1 ключ

Генерация ключей: игрушечный пример

- Алиса хочет получить общий ключ с Бобом.
- Рассматривается только пассивный атакующий



Анализ протокола

- Пассивный атакующий видит:

  E(Kb, “A,B”||Kab), E(Ka, “A,B”||Kab) •

- Если (E,D) CPA-secure ⟹ атакующий ничего не узнает о Kab

- Недостатки протокола:
  - TTR необходим для каждого обмена ключами
  - TTR – точка уязвимости (знает все сессионные ключи)

Случай активного атакующего

- Пример: Протокол не стоек к атаки повторением

- Атакующий сохраняет всю сессию между Алисой и Бобом
  - Например, Алиса купила книгу

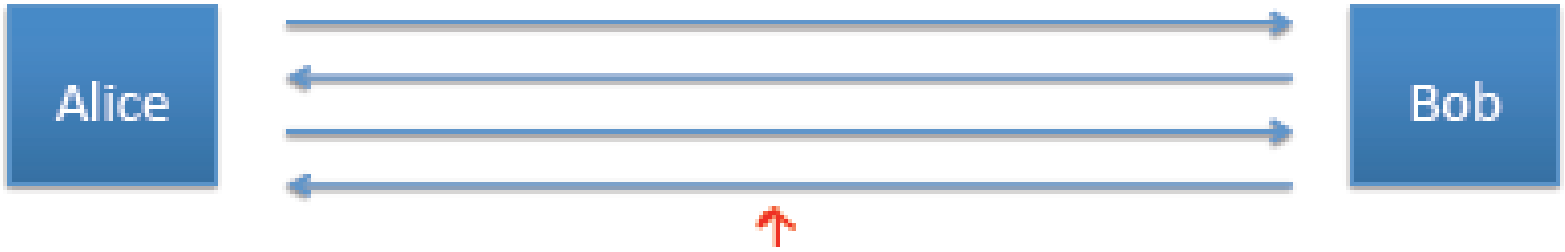
- Атакующий может повторить всю сессию от лица Алисы
  - Боб будет считать, что Алиса заказала еще одну книгу

Ключевой вопрос

- Можем ли мы обмениваться ключами без участия третьей стороны?
- Ответ: ДА!
- Основная цель создания публичной криптографии:
  - Merkle(1974), Diffie-Hellman(1976), RSA(1977)
  - Более поздние разработки: Личностная криптография (ID-Based, 2001), Функциональное шифрование(Functional encryption, 2011)

Обмен ключами без посредников 1

- Цель протокола: Обмен ключами между Алисой и Бобом
- Модель атакующего: пассивный



- Можно ли решить эту задачу при помощи симметричной криптографии?

Пазл Меркла (1974)

- Ответ: ДА, но ОЧЕНЬ неэффективно.

- **Основная идея:** использование пазлов

  - Пазл – задача, которая может быть решена с заданной сложностью

  - Пример: E(k,m) – симметричный шифр с $k \in \{0,1\}^{128}$

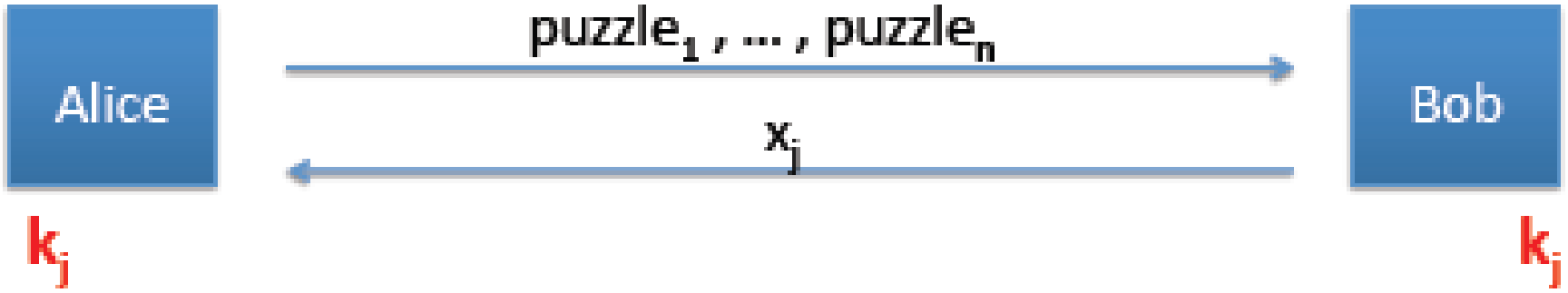
    - Puzzle(P) = E(P, “message”), где $P = 0^{96}||b_1 \ldots b_{32}$

    - Цель: найти ключ P полным перебором

Протокол: Пазлы Меркла

- **Алиса:** готовит 2^{32} пазлов
  - Случайно выбирает $P_i \in \{0,1\}^{32}, x_i, k_i \in \{0,1\}^{128}$ и вычисляет $puzzle_i \leftarrow E(0^{96} \,||P_i, \text{Puzzle \# } x_i||\, k_i)$
  - Отправить все $puzzle_i$ Бобу
- **Боб:** выбирает любой $puzzle_i$ и решает его. Получает пару (x_i, k_i)
  - Посылает x_i Алисе
- **Алиса:** находит k_i по x_i

Схема протокола



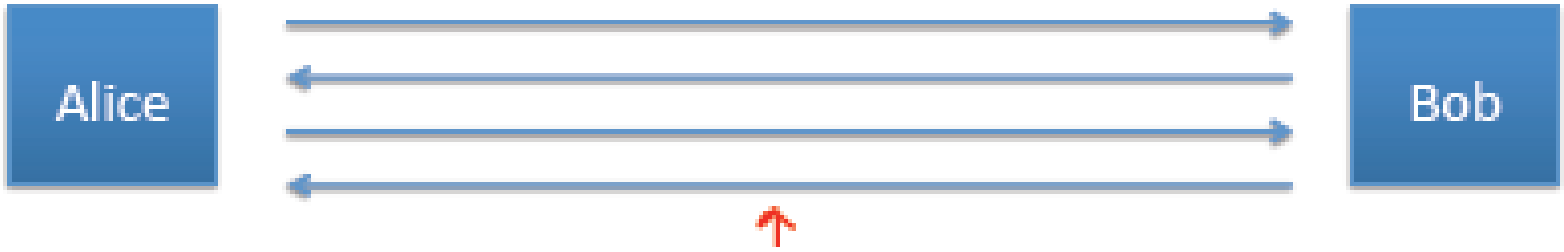
- Сложность для Алисы O(n)
- Сложность для Боба O(n)
- Сложность атаки $O(n^2)$

Верхняя граница

- Можем ли мы получить большее преимущество над атакующим?
- Неизвестно
- НО: известно,
- Выигрыш в квадрат операций наилучший возможный, если рассматривать процедуру шифрования как случайного оракула

Обмен ключами без посредников 2

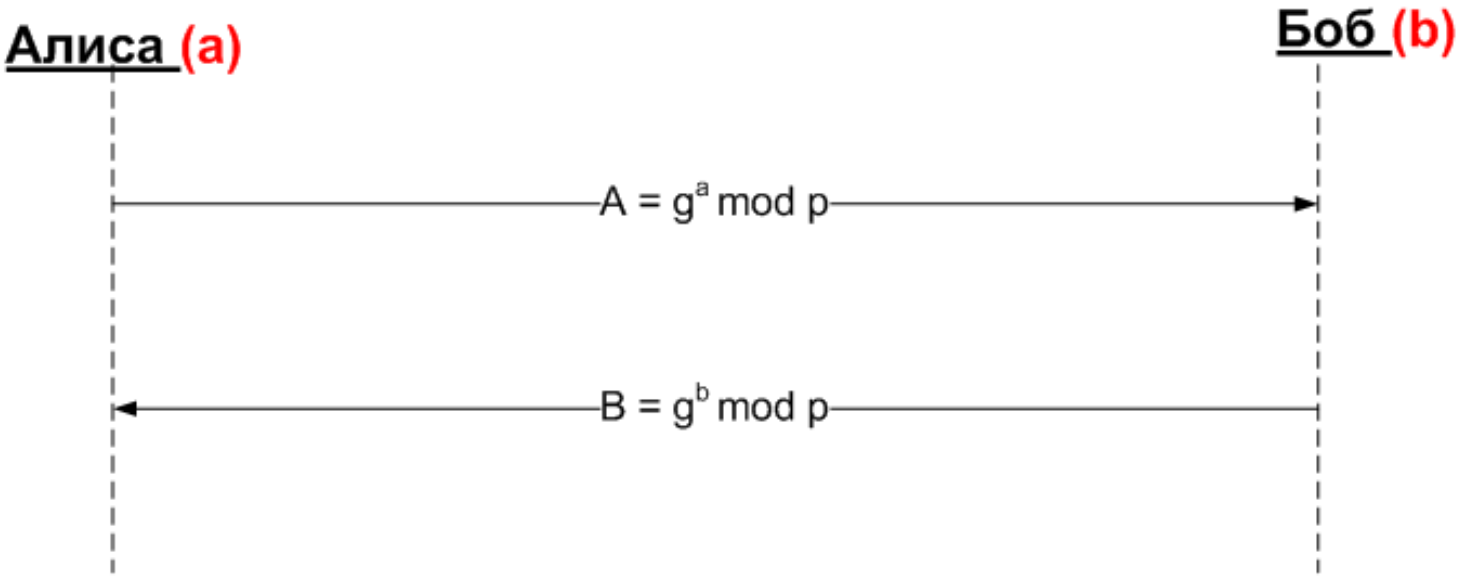
- Цель протокола: Обмен ключами между Алисой и Бобом
- Модель атакующего: пассивный



- Можно ли решить эту задачу увеличив сложность атаки?

Протокол Диффи-Хеллмана

- P большое простое число
- 0 < g < p любое целое число



$$B^a \bmod p = g^{ab} \bmod p = K_{ab} = g^{ab} = A^b \bmod p$$

Стойкость

- Пассивный атакующий видит
  - $A = g^a \pmod p$ и $B = g^b \pmod p$
- Хочет найти: $g^{ab} \pmod p$
- Какова сложность атаки?

Сложность решения проблемы ДХ

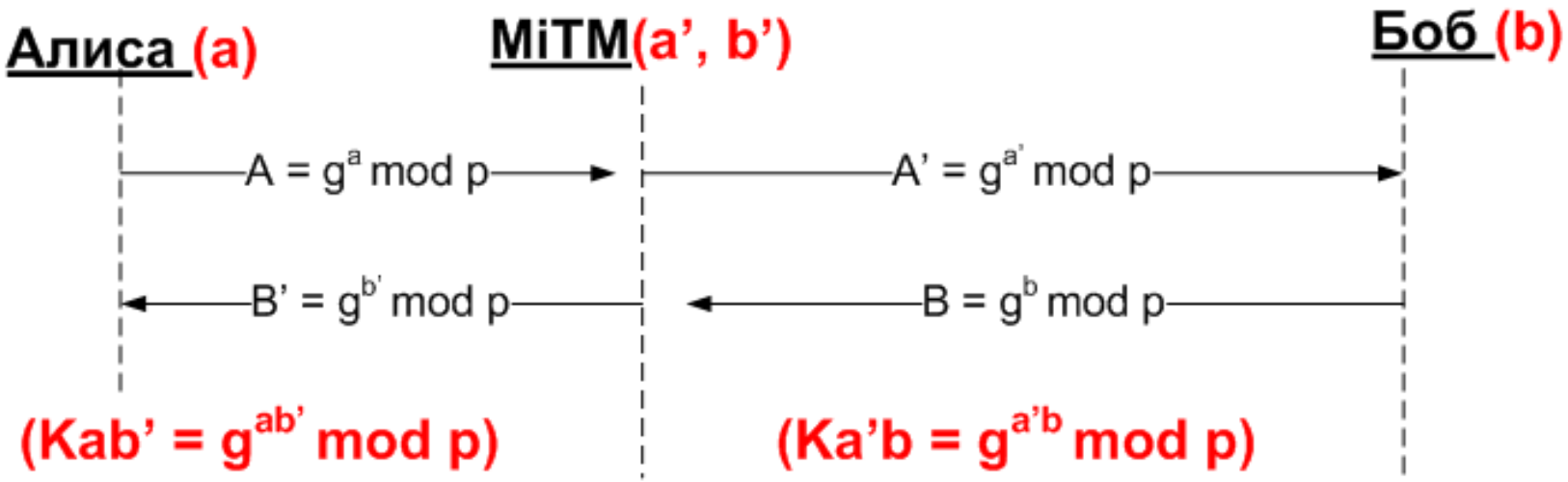
- Пусть p простое n-битное число
- Наилучший известный алгоритм: exp(O($\sqrt[3]{n}$))

cipher key size	modulus size	Elliptic Curve size
80 bits	1024 bits	160 bits
128 bits	3072 bits	256 bits
256 bits (AES)	**15360** bits	512 bits

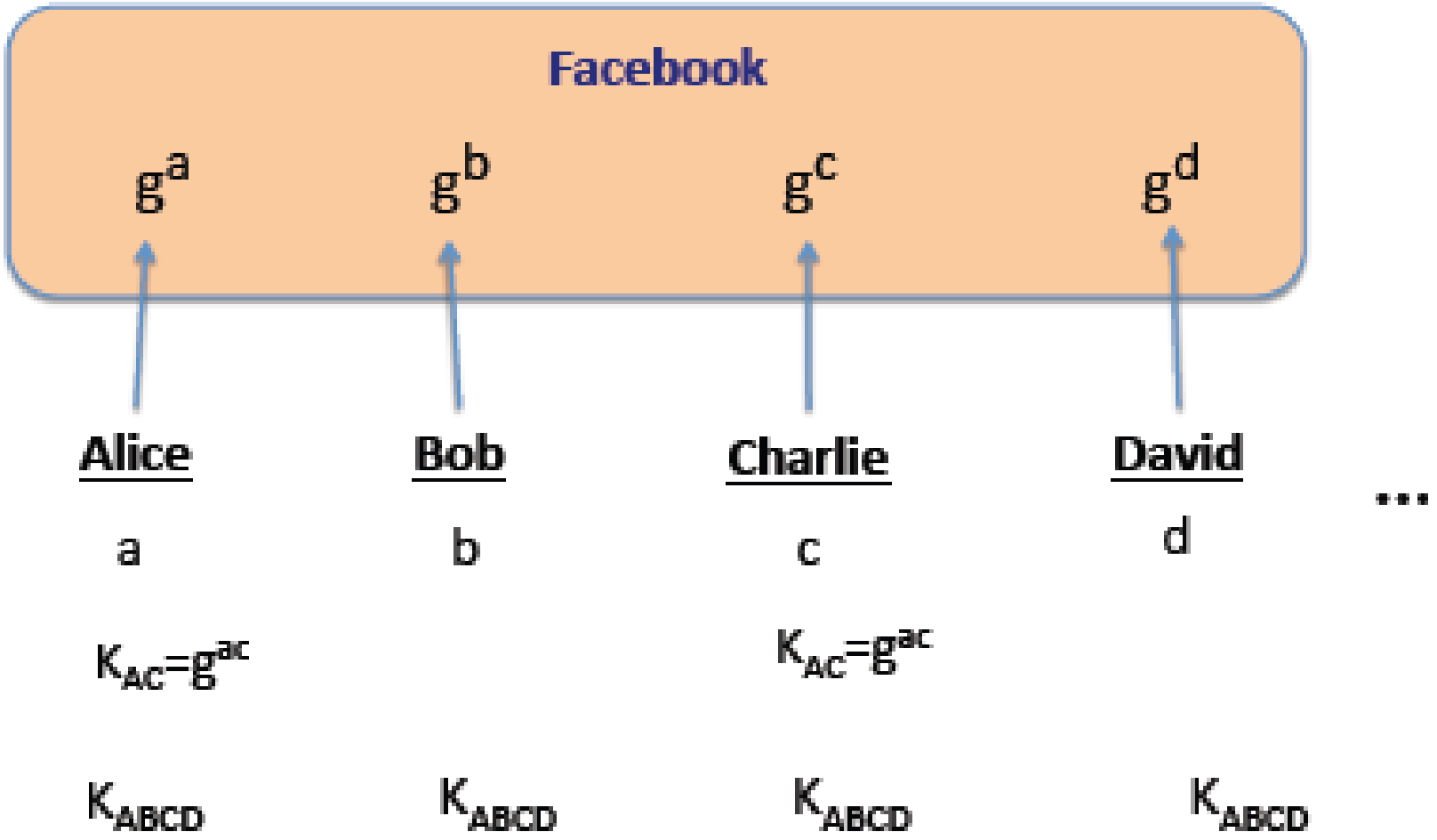
В результате: переход к эллиптическим кривым

Проблема протокола ДХ

- Нестойкость к атаке Человек-Посередине (MiTM)
  - Активный атакующий



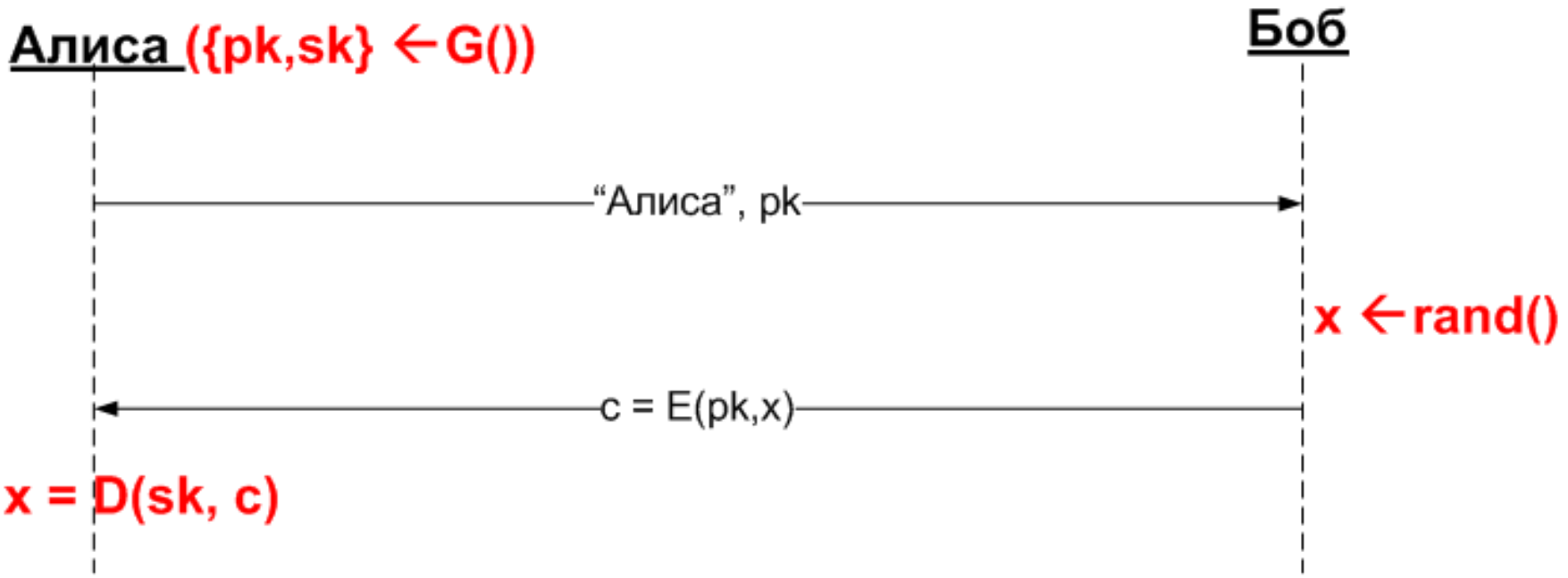
Многопользовательский протокол ДХ



Обмен ключами без посредников 3

- Цель: обмен секретом
- Модель атакующего: пассивный
- Подход: Использование публичной криптографии

Выработка общего секрета

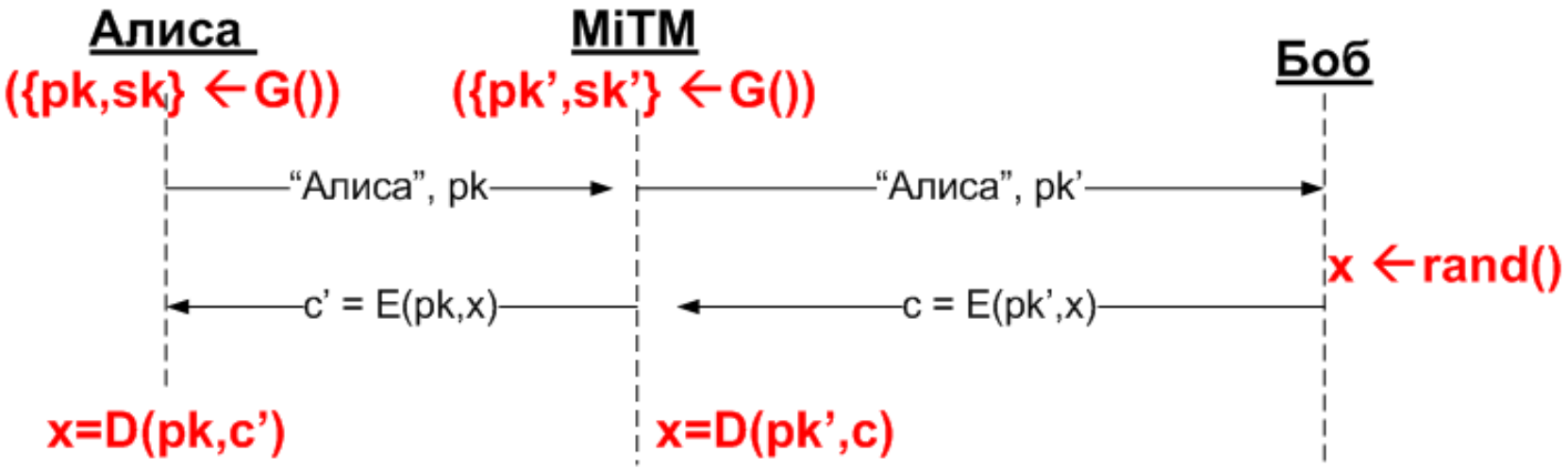


Стойкость

- Атакующий видит **pk, E(pk,x)**
- Хочет получить $x \in M$
- Семантическая стойкость →
- атакующий не может различить {pk, E(pk,x), x} и {pk, E(pk,x), rand $\in$ M} →
- → не может получить сессионный ключ
- Протокол по прежнему уязвим к Атаке Человек-посередине

Атака человек посередине

- Протокол не стоек к активным атакам



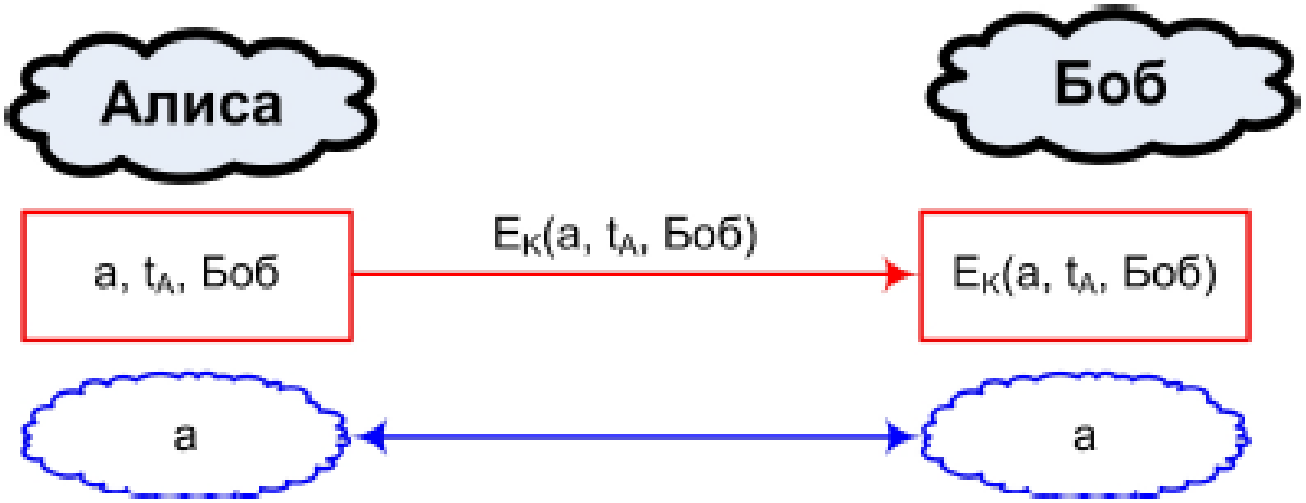
Желаемые свойства ключа

- Аутентификация личности (entity authentication): Алиса уверена, что с ней разговаривает Боб.
- Аутентификация источника данных (data origin authentication): Алиса уверена, что именно Боб написал это сообщение.
- Аутентификация ключа (key authentication): Алиса уверена, что только Боб (и еще, может быть, Центр) сможет узнать секретный ключ.

- Подтверждение ключа (key conrmation): Алиса уверена, что Боб ключ получил и готов его использовать.
- Идеальная прямая безопасность (perfect forward secrecy): если Чарли узнает настоящий перманентный секретный ключ, это не поможет ему взломать частные ключи от предыдущих, уже состоявшихся обменов между Алисой и Бобом.
- Свежесть ключа (key freshness): участники уверены, что ключ свежий и раньше не использовался.

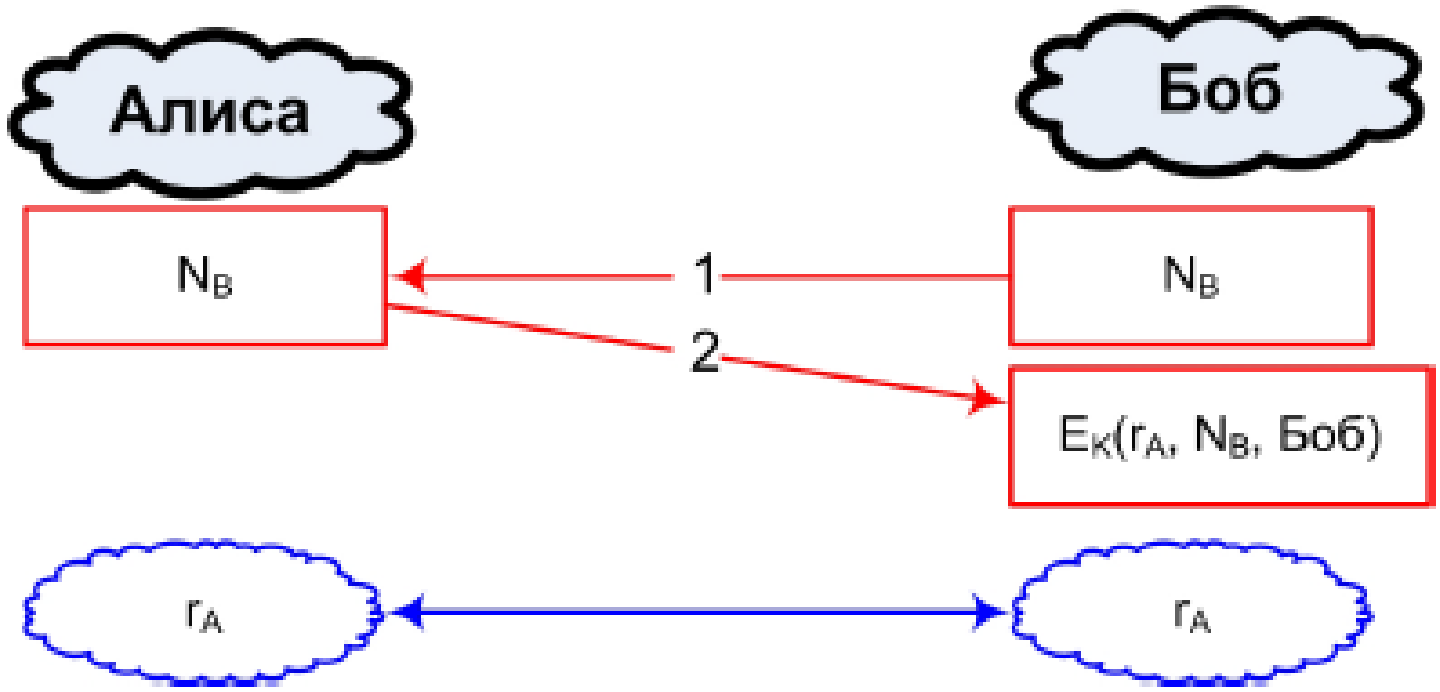
Симметричное шифрование

- Пусть у Алисы и Боба уже есть общий секрет
- Цель: Передача временного ключа
- Однопроходовый key transport



key authentication, key freshness, conrmation

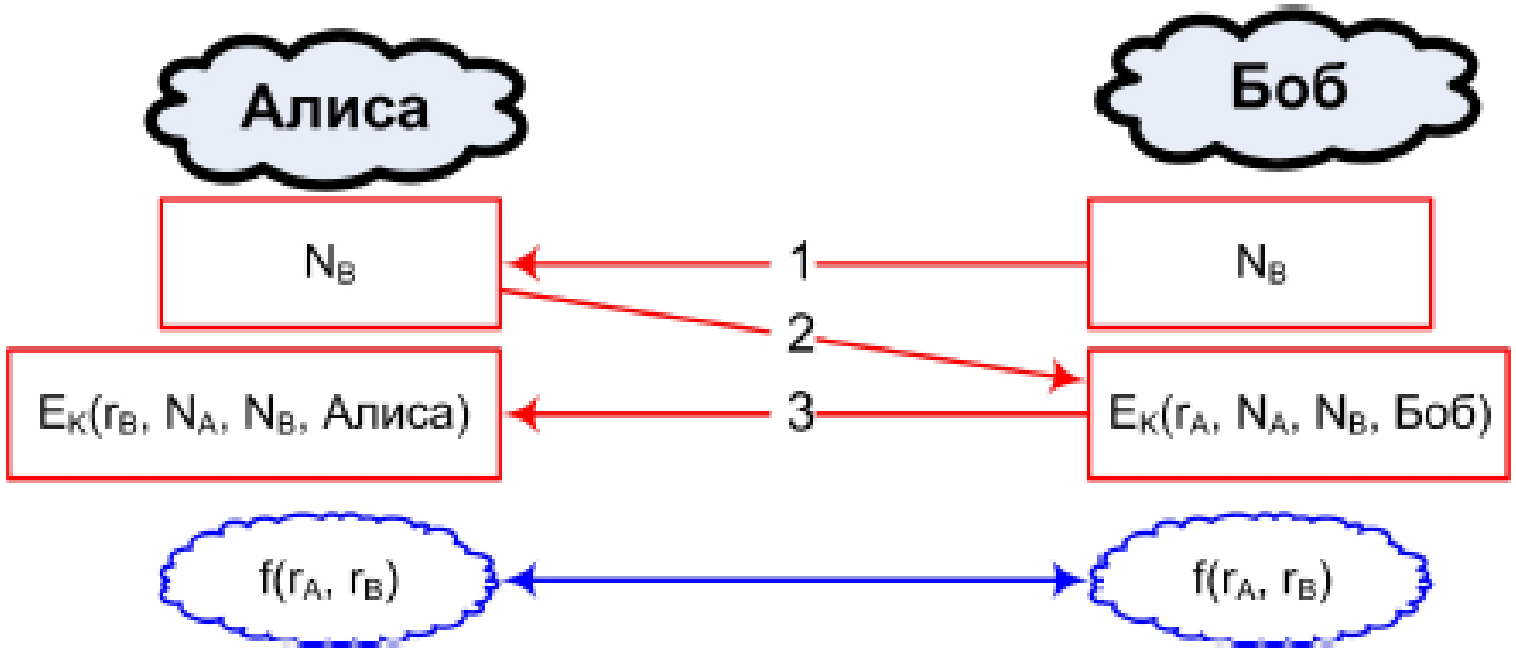
Протокол challenge-response



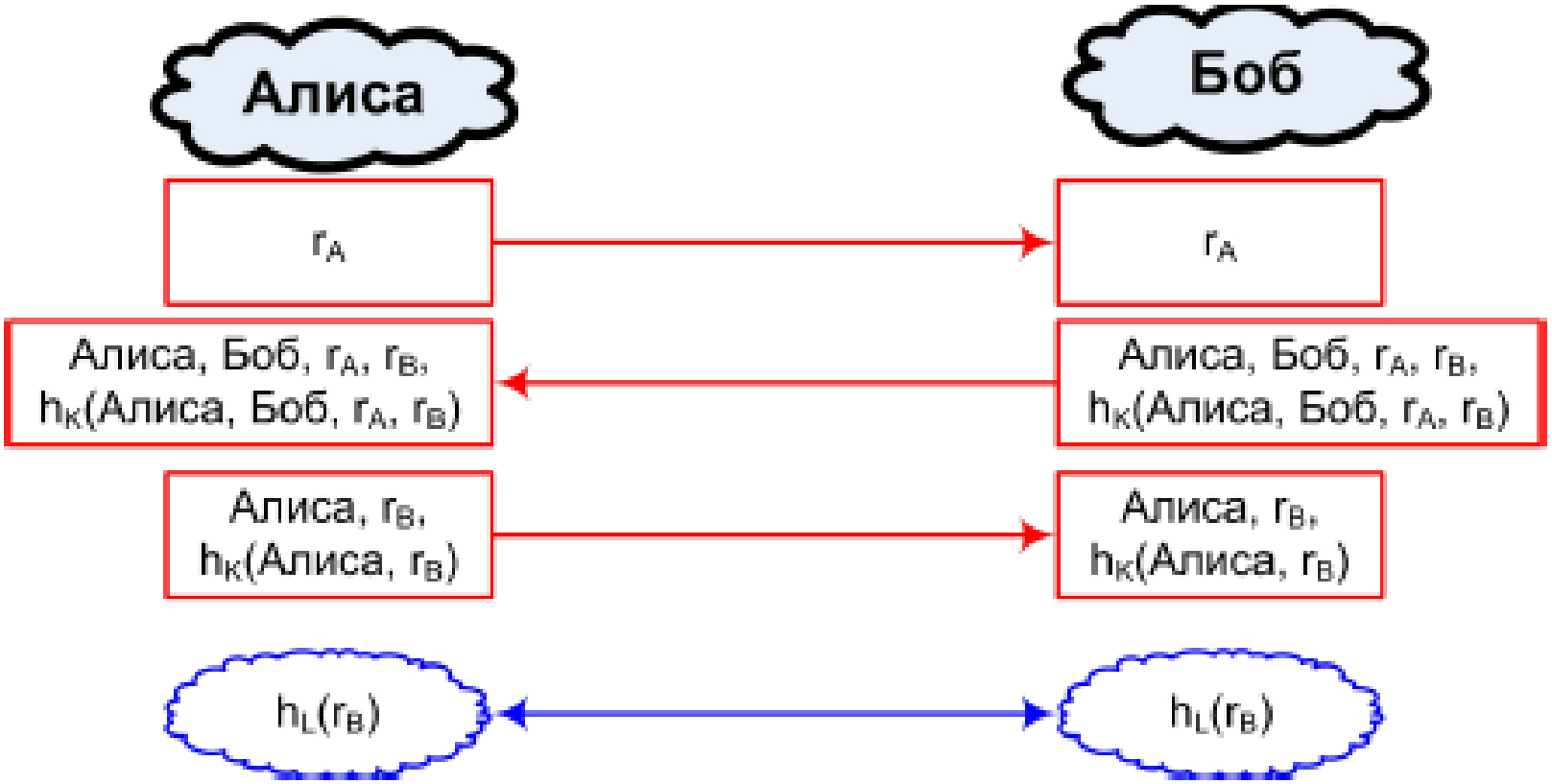
- Не нуждается в синхронизации времени

Согласование ключа

- Алиса не контролирует ключ



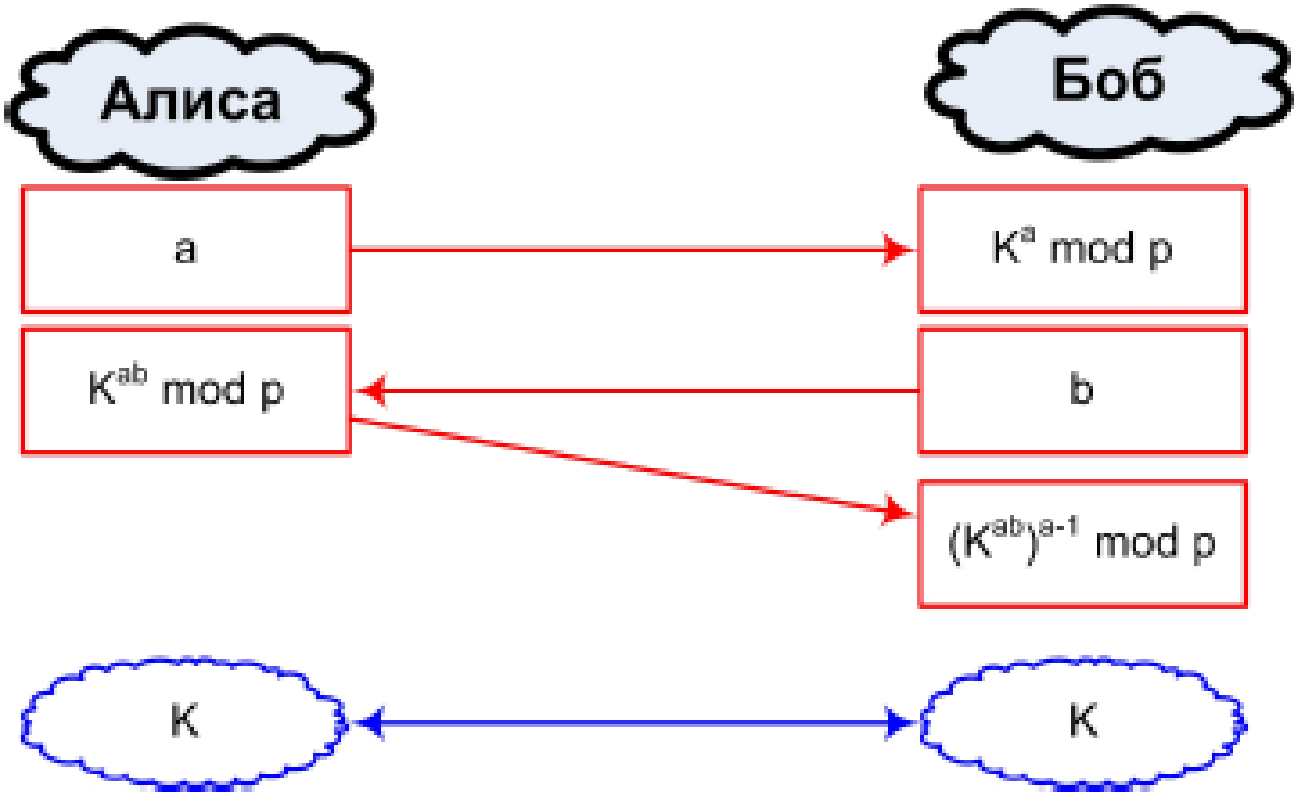
Authenticated key exchange



Взаимное key authentication и key confirmation

Протокол Шамира передачи ключа

- Нет общего секрета → нет аутентификации
- Идеальная прямая безопасность



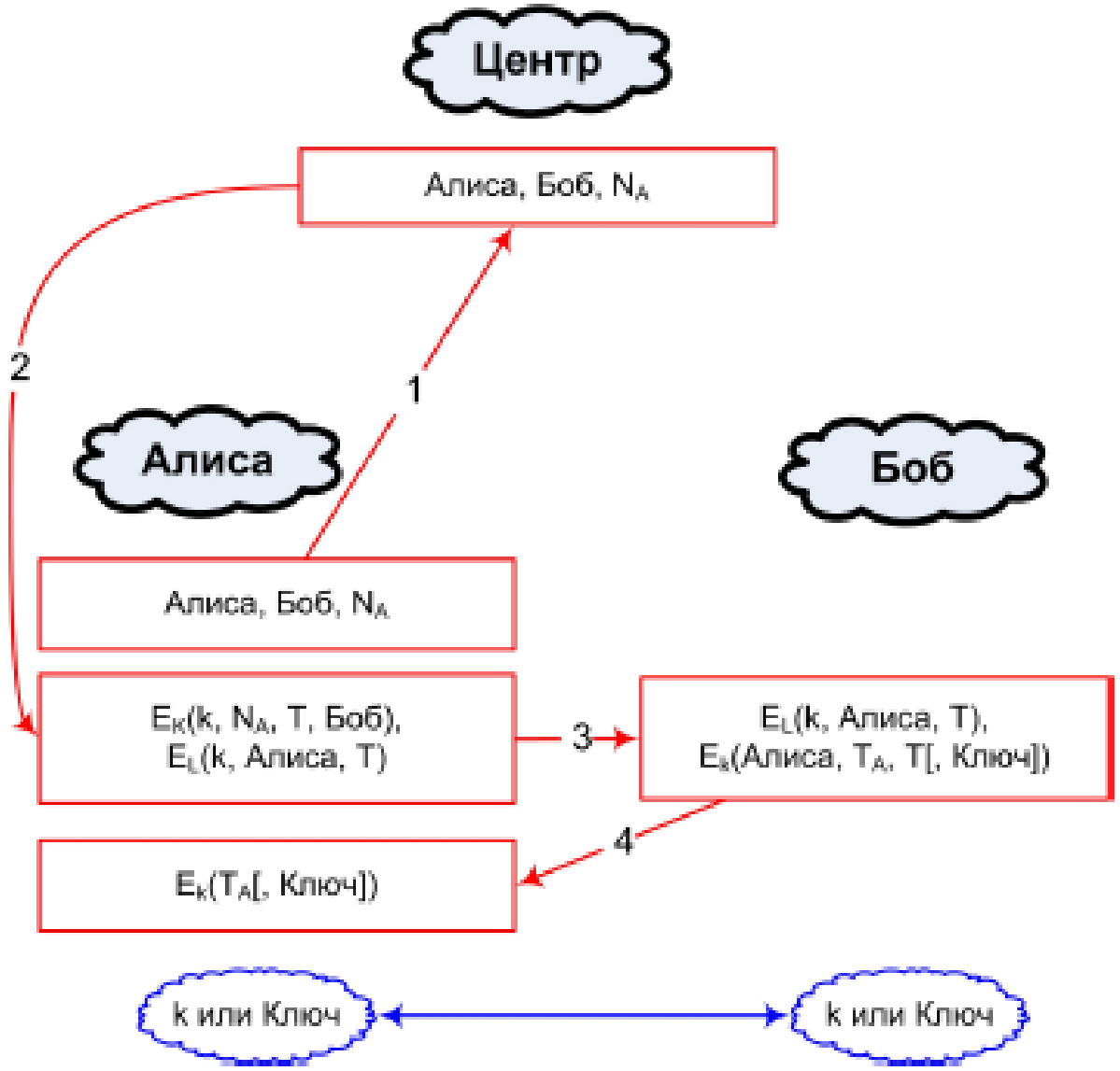
Протоколы с сервером

- Теперь давайте предположим, что у нас есть сервер, которому верят и Алиса, и Боб.
- У сервера и Алисы есть заранее секретный ключ K, а у сервера и Боба секретный ключ L.
- Алиса и Боб хотят сделать временный секретный ключ для общения друг с другом.

Kerberos

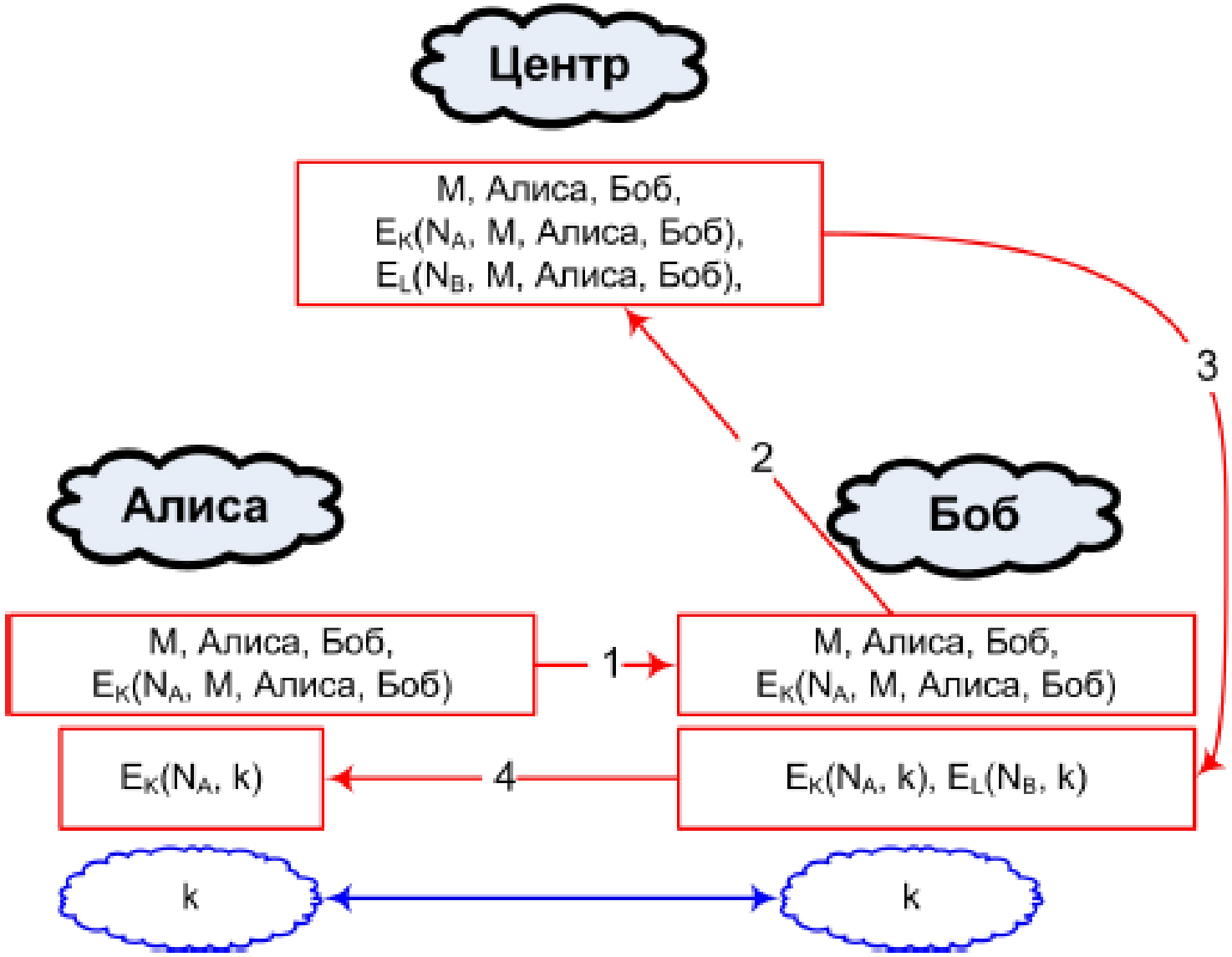
- Главная цель -- аутентификация (как раз entity), но как побочный эффект и ключ согласовывается.

- Обозначения:
  - Алиса и Центр знают секретный ключ K;
  - Боб и Центр знают секретный ключ L;
  - Алиса выбирает NA и вводит timestamp TA по своим часам;
  - Центр выбирает временный ключ k для Алисы и Боба;
  - T период валидности (lifetime), выбираемый Центром.
- Достоинства:
  - взаимные entity authentication и key confirmation.


Центр
Алиса, Боб, N_A
2
1
Алиса
Боб
Алиса, Боб, N_A
$E_K(k, N_A, T, Боб)$,
$E_L(k, Алиса, T)$
3
$E_L(k, Алиса, T)$,
$E_b(Алиса, T_A, T[, Ключ])$
4
$E_k(T_A[, Ключ])$
k или Ключ
k или Ключ

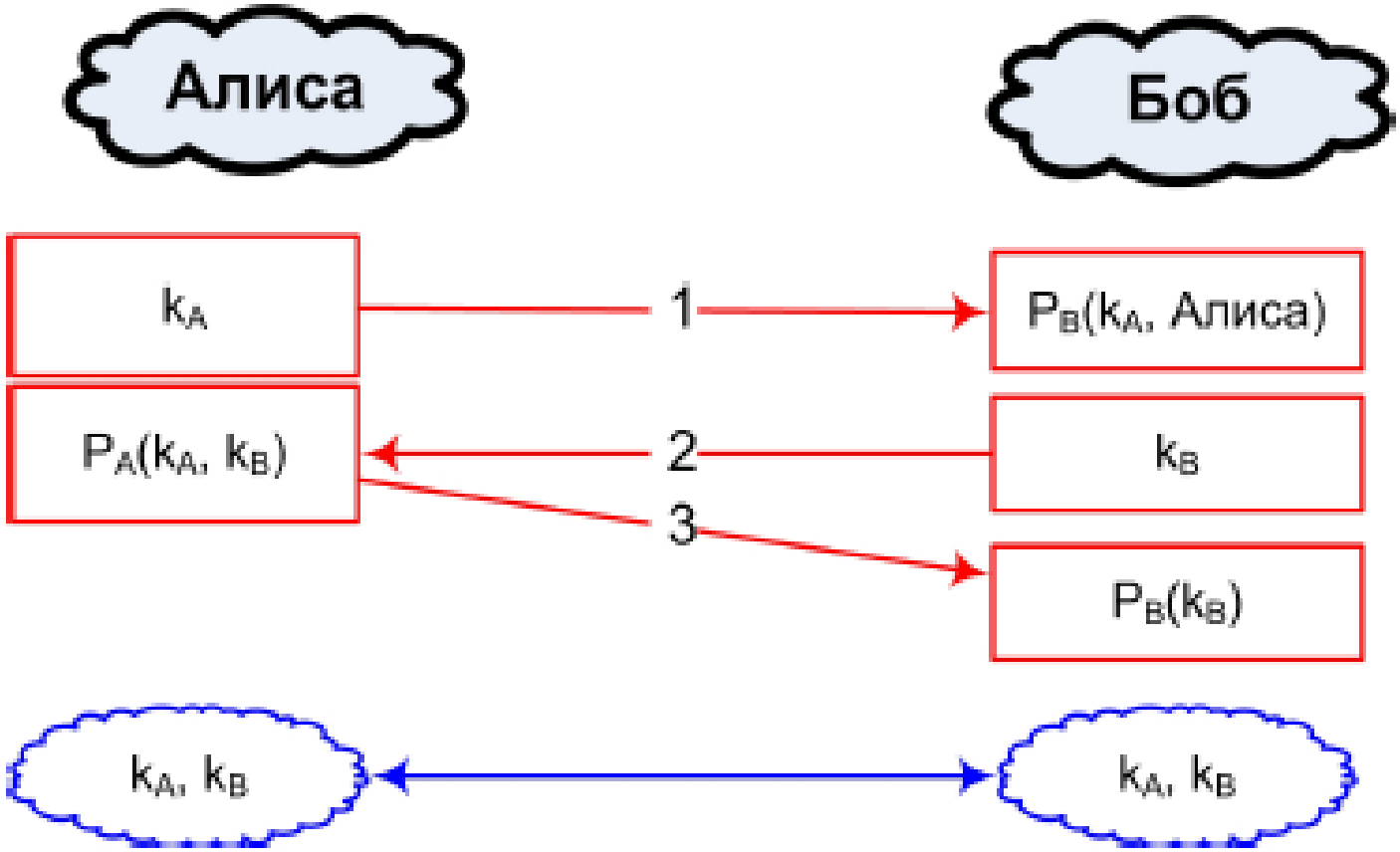
Протокол Отвэя-Рииса

- Протокол Отвэя-Рииса (Otway-Rees) похож на Керберос, но не нужны timestamp‘ы.
- Достигается key authentication и key freshness
- Возможно модифицировать и получить confirmation.


Центр
M, Алиса, Боб,
$E_K(N_A, M, Алиса, Боб)$,
$E_L(N_B, M, Алиса, Боб)$,
3
2
Алиса
Боб
M, Алиса, Боб,
$E_K(N_A, M, Алиса, Боб)$
1
M, Алиса, Боб,
$E_K(N_A, M, Алиса, Боб)$
$E_K(N_A, k)$
4
$E_K(N_A, k)$, $E_L(N_B, k)$
k
k

Протоколы на публичных ключах

- Needham-Schroeder протокол (требует PKI)

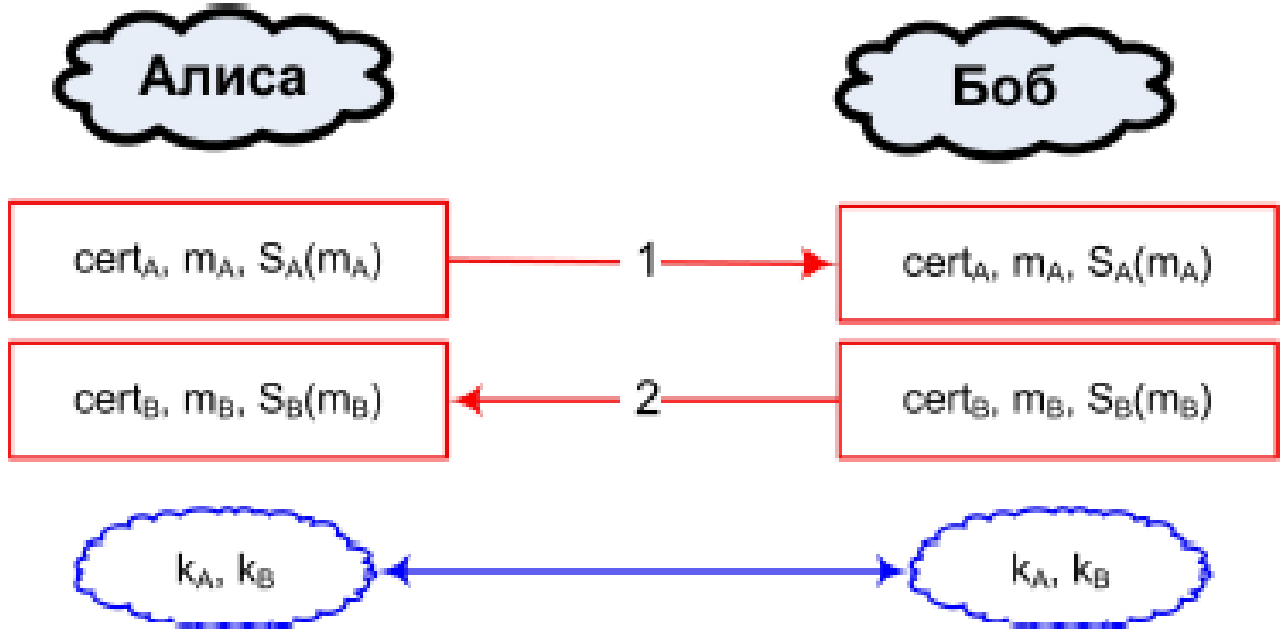


X.509

- Алгоритм X.509 -- стандарт аутентификации с открытым ключом.
- Он предполагает систему *сертификатов*, которые выпускают специальные доверенные стороны.
- Сертификат Алисы, выпущенный Центром, содержит публичные ключи Алисы для подписи и кодирования, а также подписан Центром, т.е. его никто не может подделать.

X.509

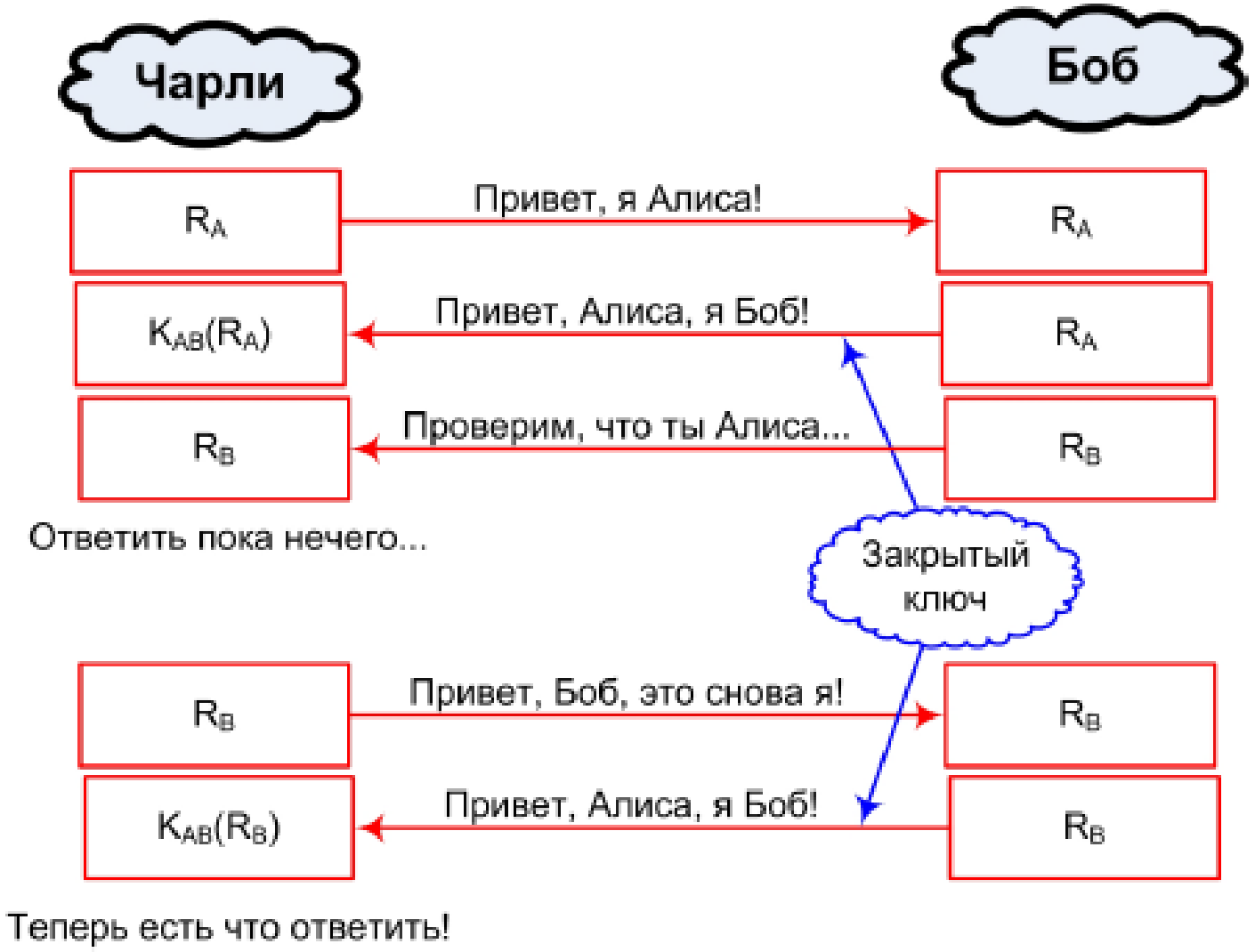
- Здесь сообщение Алисы $m_A = (t_A, r_A, \text{Боб}, PB(k_A))$, а
- сообщение Боба $m_B = (t_B, r_B, \text{Алиса}, r_A, PA(k_B))$
- В результате происходит аутентификация и обмен секретными ключами k_A и k_B.



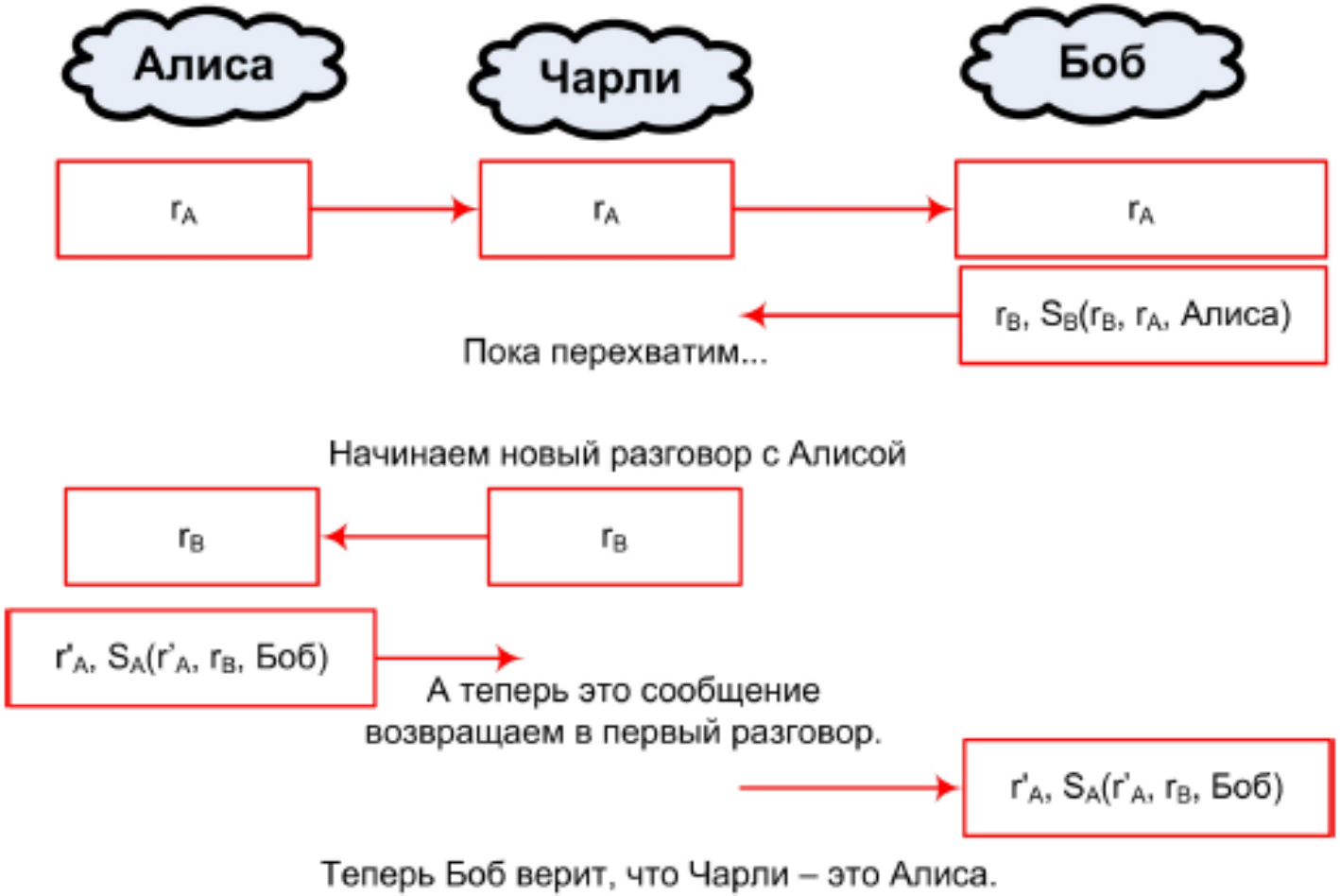
Активные атаки

- MiTM (человек посередине)
- Reflection attack
- Interleaving attack
- Misplaced trust

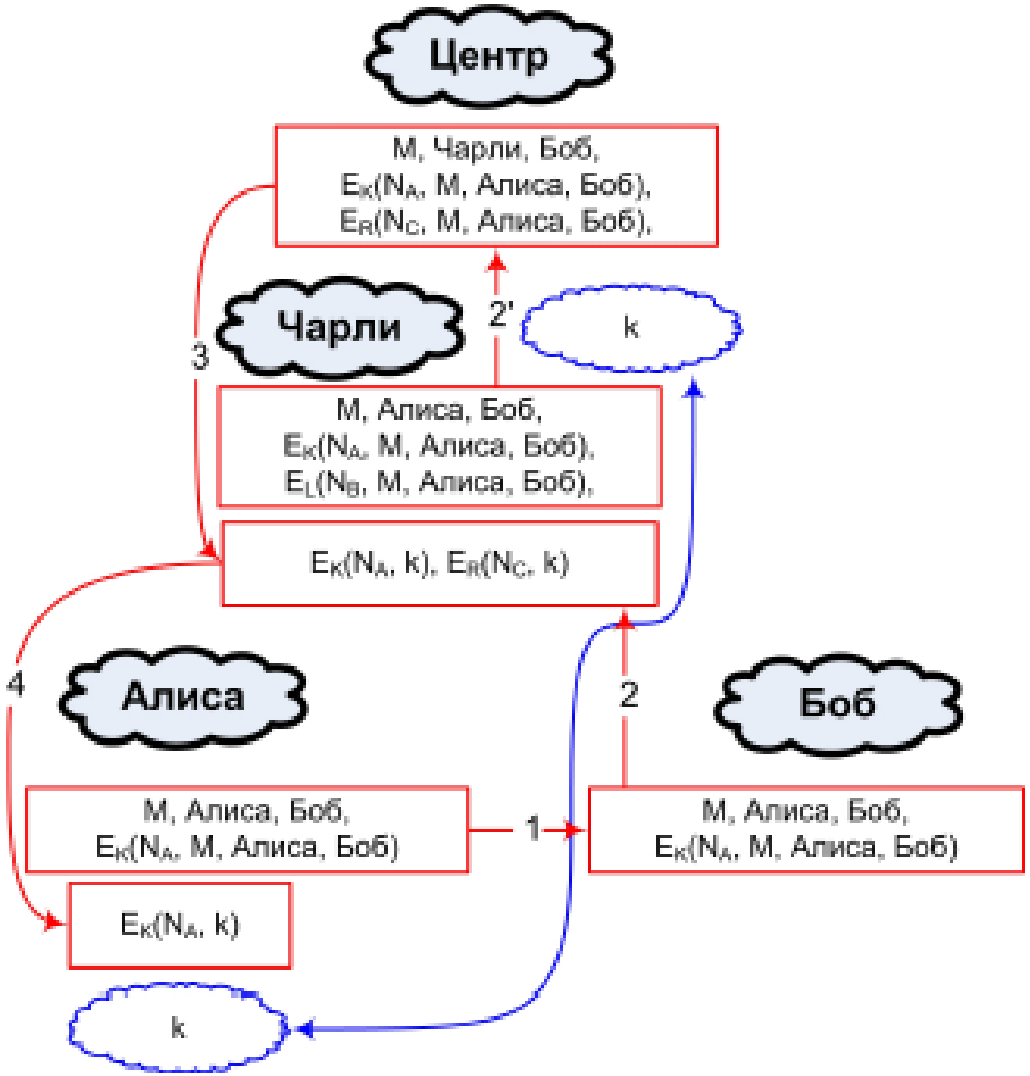
Reflection attack



Interleaving attack



Misplaced trust



Формальный анализ протоколов

- Идея: Всякий реальный протокол должен быть устойчив против известных атак.
- Моделирование и проверка работы
- Создание экспертных систем
- Использование специфических логик в терминах «знание», «доверие»
- Запись свойств криптографических систем в алгебраическом виде