СЛАЙД 1 ==========

1. Введение

Алгоритм Double Ratchet используется двумя сторонами для обмена зашифрованными сообщениями на основе общего секретного ключа. Обычно стороны используют какой-либо протокол согласования ключей (например, X3DH (Extended Triple Diffie-Hellman)) для согласования общего секретного ключа. После этого стороны используют алгоритм Double Ratchet для отправки и получения зашифрованных сообщений.

Стороны получают новые ключи для каждого сообщения Double Ratchet, так что более ранние ключи не могут быть вычислены из более поздних. Стороны также отправляют открытые значения Диффи-Хеллмана, прикрепленные к их сообщениям. Результаты вычислений Диффи-Хеллмана подмешиваются к полученным ключам, так что более поздние ключи не могут быть вычислены из более ранних. Эти свойства обеспечивают определенную защиту более ранних или более поздних зашифрованных сообщений в случае компрометации ключей одной из сторон.

Ниже представлены "Двойной храповик" и его вариант шифрования заголовков, а также обсуждаются их защитные свойства.

СЛАЙД 2 ==========

2. Обзор

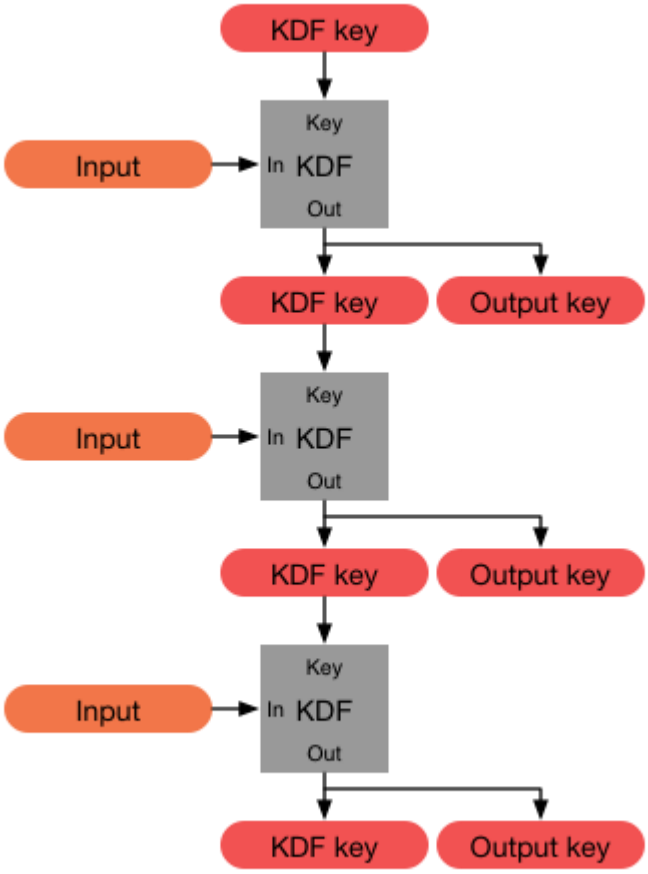
2.1. Цепочки KDF

Цепочка KDF (key derivation function - функция формирования ключа) является основным понятием алгоритма Double Ratchet.

Мы определяем **KDF** как криптографическую функцию, которая принимает секретный случайный **ключ KDF** и некоторые входные данные и возвращает выходные данные. Выходные данные неотличимы от случайных при условии, что ключ неизвестен (то есть KDF удовлетворяет требованиям криптографического "PRF" (pseudorandom function family)). Если ключ не является секретным и случайным, KDF все равно должен обеспечивать безопасный криптографический хэш своего ключа и входных данных.

Конструкции HMAC и HKDF при использовании безопасного хэш-алгоритма удовлетворяют определению KDF.

Мы используем термин **"цепочка KDF"**, когда часть выходных данных KDF используется в качестве **выходного ключа**, а часть - для замены ключа KDF, который затем может быть использован с другим входом. На следующей диаграмме представлена цепочка KDF, обрабатывающая три входа и производящая три выходных ключа:



Цепочка KDF обладает следующими свойствами:

Устойчивость: Выходные ключи кажутся случайными для противника, не знающего ключи KDF. Это верно, даже если противник может контролировать входы KDF.

Защита наперед: Противнику, узнавшему ключ KDF в определенный момент времени, выходные ключи из прошлого кажутся случайными.

Восстановление после взлома: Будущие выходные ключи кажутся случайными для противника, который узнает ключ KDF в какой-то момент времени, при условии, что будущие входные данные добавили достаточную энтропию.

В **сессии Double Ratchet** между Алисой и Бобом каждая сторона хранит ключ KDF для трех цепочек: **корневой**, **отправляющей** и **принимающей** (отправляющая цепочка Алисы совпадает с принимающей цепочкой Боба, и наоборот).

Когда Алиса и Боб обмениваются сообщениями, они также обмениваются новыми открытыми ключами Диффи-Хеллмана, а выходные секреты Диффи-Хеллмана становятся входными для корневой цепочки. Выходные ключи корневой цепочки становятся новыми ключами KDF для цепочек отправителей и получателей. Это называется **храповиком Диффи-Хеллмана**.

Цепочки отправителей и получателей продвигаются вперед по мере отправки и получения каждого сообщения. Их выходные ключи используются для шифрования и дешифрования сообщений. Это называется **храповиком с симметричным ключом.**

В следующих разделах более подробно объясняются храповики симметричных ключей и Диффи-Хеллмана, а затем показано, как они объединяются в двойной храповик.

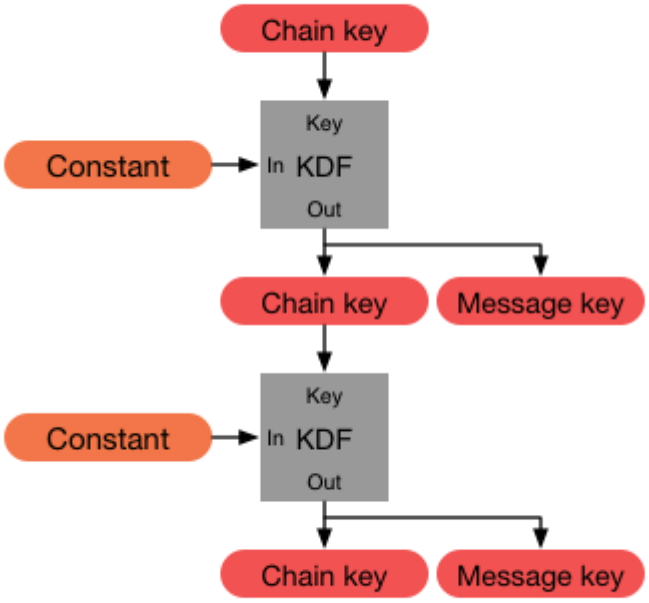
СЛАЙД 3 ==========

2.2. Храповик с симметричным ключом

Каждое отправленное или полученное сообщение шифруется уникальным **ключом сообщения**. Ключи сообщений являются выходными ключами цепочек KDF отправителя и получателя. Ключи KDF для этих цепочек будут называться **ключами цепочки**.

Входы KDF для цепочек отправки и получения постоянны, поэтому эти цепочки не обеспечивают восстановления после взлома. Цепочки отправки и получения просто обеспечивают шифрование каждого сообщения

уникальным ключом, который может быть удален после шифрования или дешифрования. Вычисление следующего ключа цепочки и ключа сообщения из заданного ключа цепочки - это один **шаг храповика** в **храповике симметричных ключей**. На приведенной ниже схеме показаны два шага:



Поскольку ключи сообщений не используются для получения других ключей, ключи сообщений могут храниться без ущерба для безопасности других ключей сообщений. Это полезно для обработки потерянных или неупорядоченных сообщений.

СЛАЙД 4 ==========

2.3. Храповик Диффи-Хеллмана

Если злоумышленник украдет цепные ключи одной из сторон, он сможет вычислить все будущие ключи сообщений и расшифровать все будущие сообщения. Чтобы предотвратить это, двойной храповик объединяет храповик с симметричным ключом с **храповиком DH**, который обновляет ключи цепочки на основе результатов Диффи-Хеллмана.

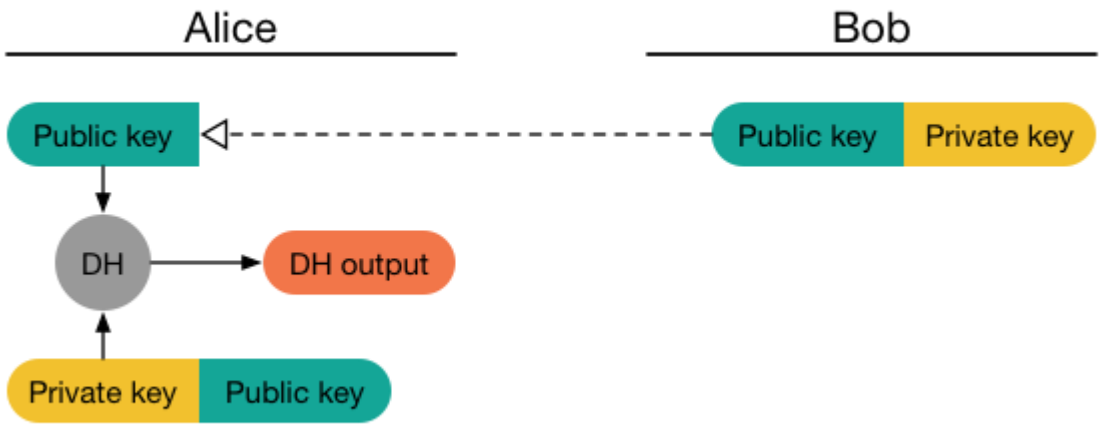
Для реализации храповика DH каждая сторона генерирует пару ключей DH (открытый ключ и закрытый ключ Диффи-Хеллмана), которая становится их текущей **парой ключей храповика**. Каждое сообщение от любой стороны

начинается с заголовка, содержащего текущий открытый ключ храповика отправителя. Когда от удаленной стороны поступает новый открытый ключ храповика, выполняется **шаг храповика DH**, который заменяет текущую пару ключей храповика локальной стороны новой парой ключей.

Это приводит к "пинг-понгу", когда стороны по очереди заменяют пары ключей-храповиков. Подслушивающее лицо, ненадолго скомпрометировавшее одну из сторон, может узнать значение текущего закрытого ключа храповика, но этот закрытый ключ в конце концов будет заменен на некомпрометированный. В этот момент вычисления Диффи-Хеллмана между парами ключей-храповиков будут определять выход DH, неизвестный злоумышленнику.

На следующих диаграммах показано, как храповик DH извлекает общую последовательность выходов DH.

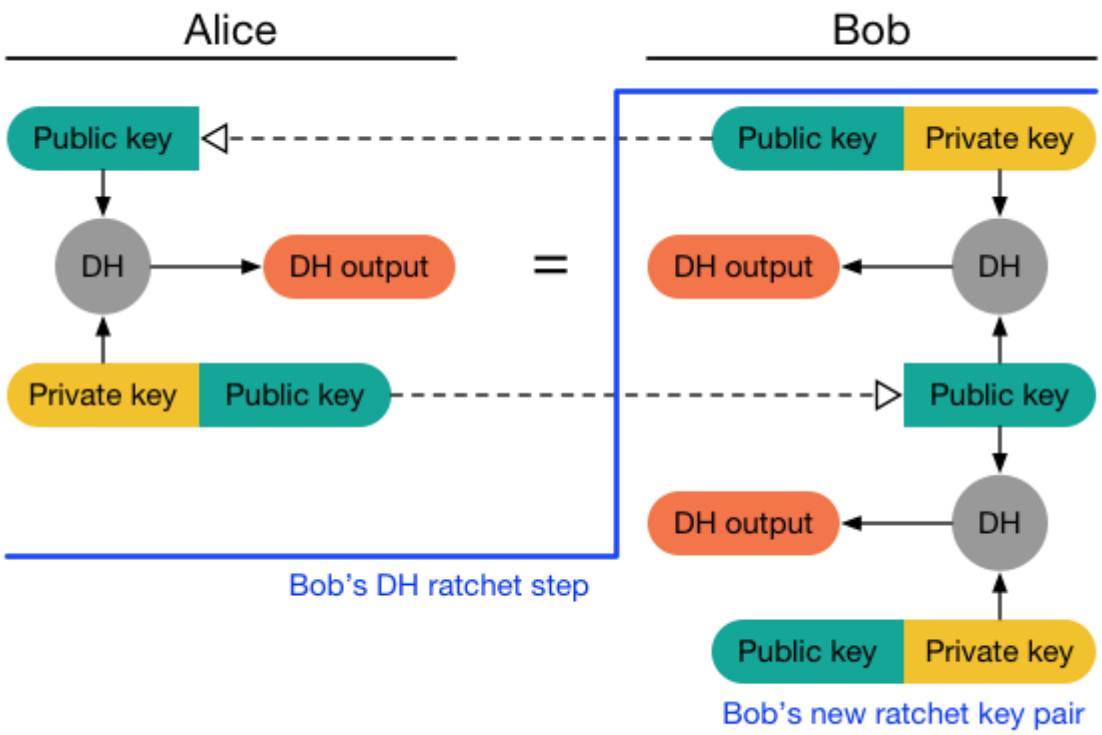
Алиса инициализируется открытым ключом храповика Боба. Открытый ключ храповика Алисы еще не известен Бобу. В процессе инициализации Алиса выполняет вычисление DH между своим закрытым ключом храповика и открытым ключом храповика Боба:



СЛАЙД 5 ==========

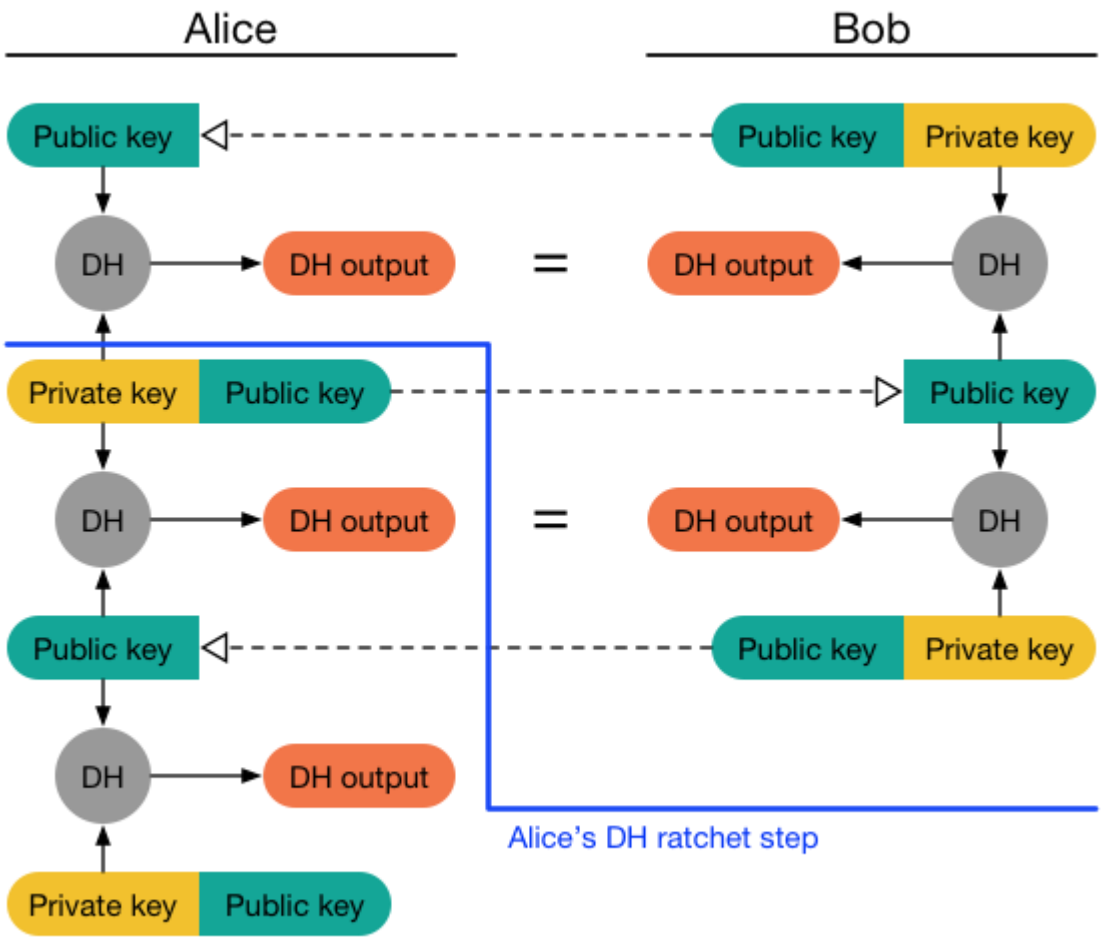
Начальные сообщения Алисы афишируют ее открытый ключ с храповиком. Как только Боб получает одно из этих сообщений, Боб выполняет

шаг храповика DH: он вычисляет результат DH между открытым ключом храповика Алисы и своим закрытым ключом храповика, который равен начальному результату DH Алисы. Затем Боб заменяет свою пару ключей храповика и вычисляет новый результат DH:



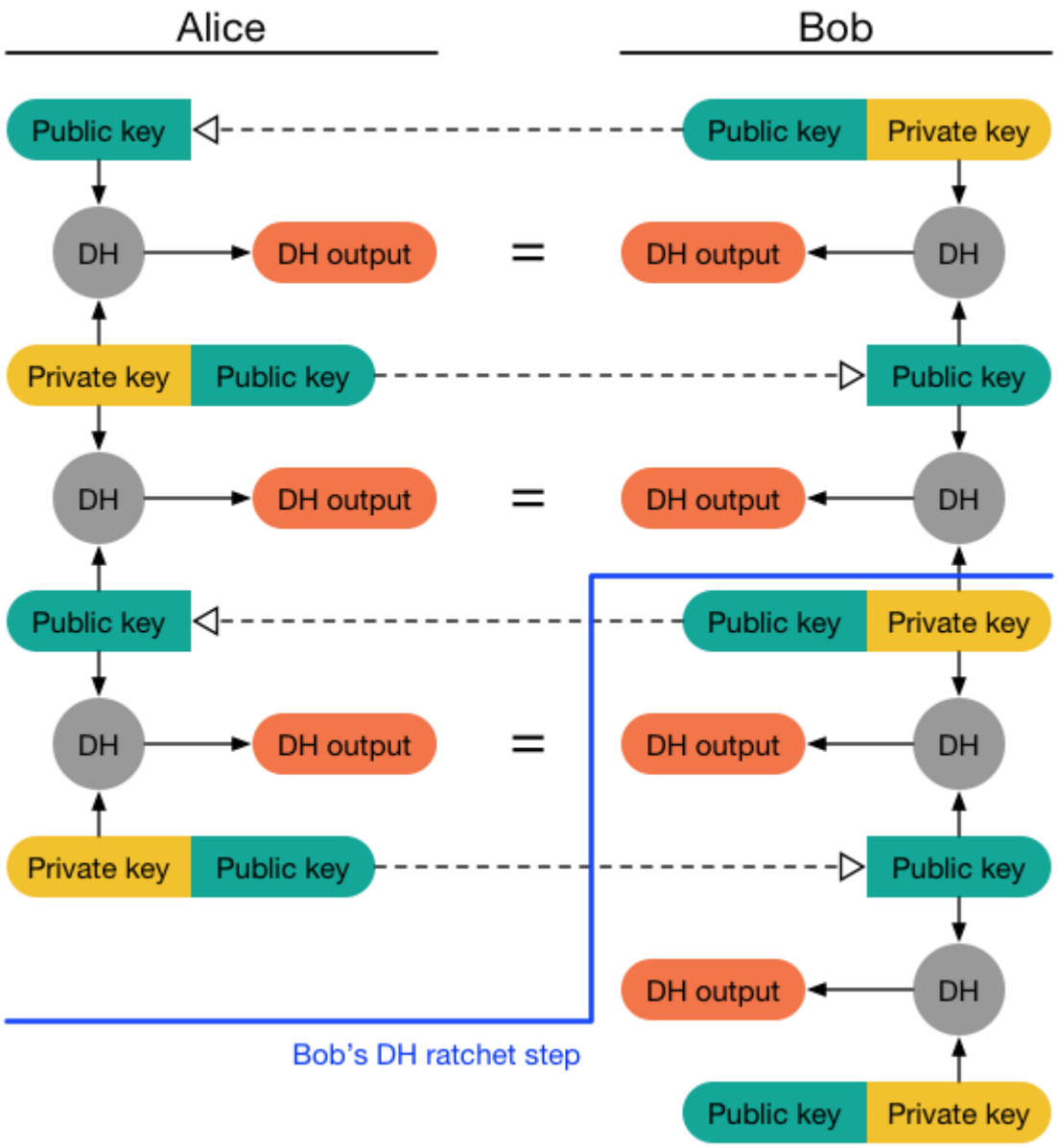
СЛАЙД 6 ==========

Сообщения, отправленные Бобом, афишируют его новый открытый ключ. В конце концов Алиса получит одно из сообщений Боба и выполнит шаг храповика DH, заменив свою пару ключей храповика и получив два выхода DH: один, соответствующий последнему ключу Боба, и новый:



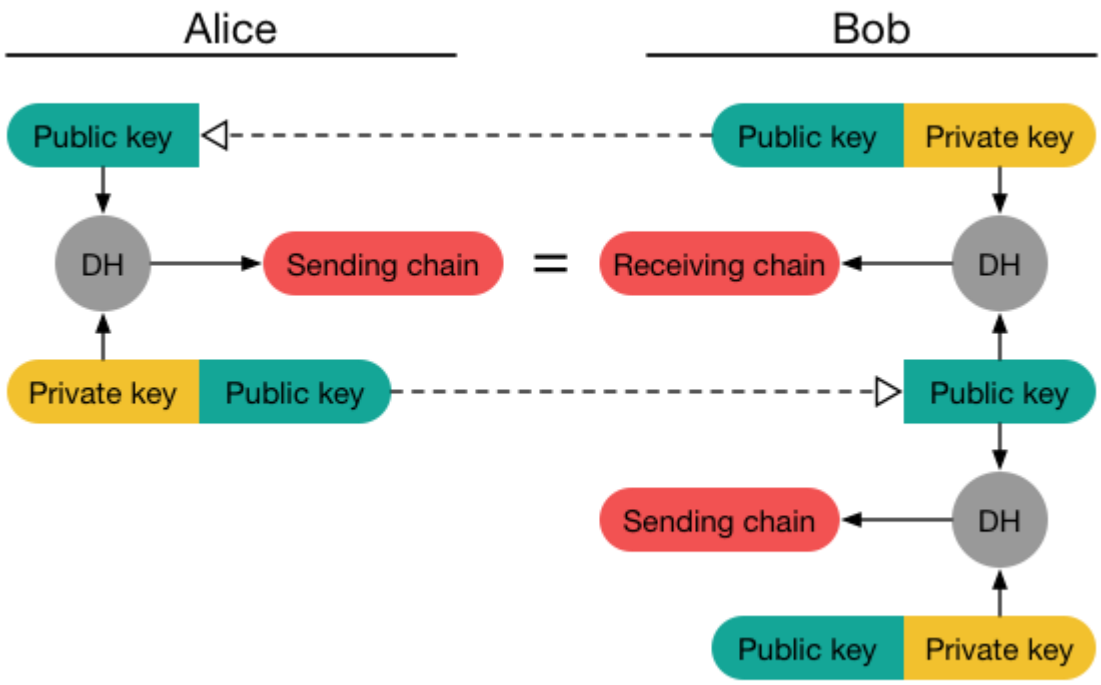
СЛАЙД 7 ==========

Сообщения, отправленные Алисой, афишируют ее новый открытый ключ. В конце концов Боб получит одно из этих сообщений и выполнит второй шаг храповика DH, и так далее:



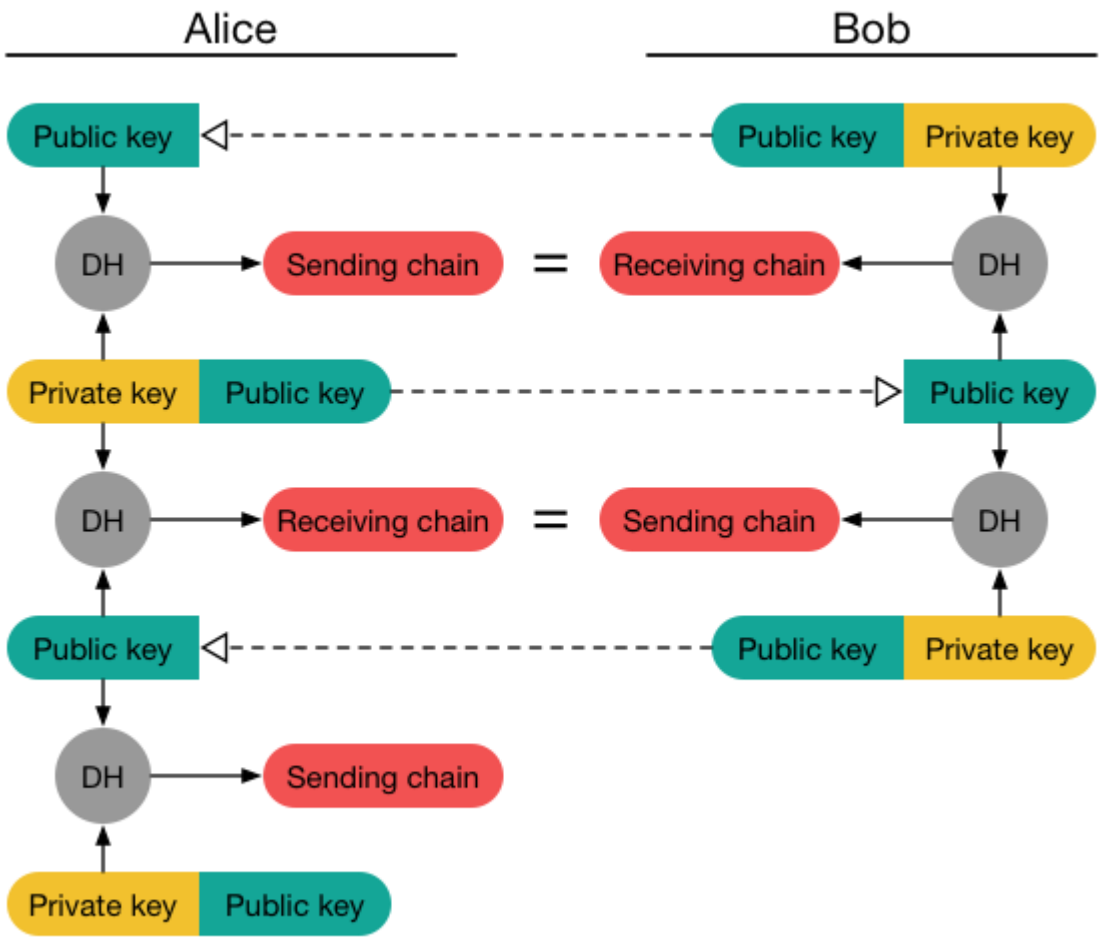
СЛАЙД 8 ==========

Результаты DH, сгенерированные на каждом шаге храповика DH, используются для получения новых цепных ключей отправки и получения. На приведенной ниже диаграмме рассматривается первый шаг храповика Боба. Боб использует свой первый вывод DH для получения цепочки получения, которая совпадает с цепочкой отправки Алисы. Боб использует второй результат DH для получения новой цепочки отправки:



СЛАЙД 9 ==========

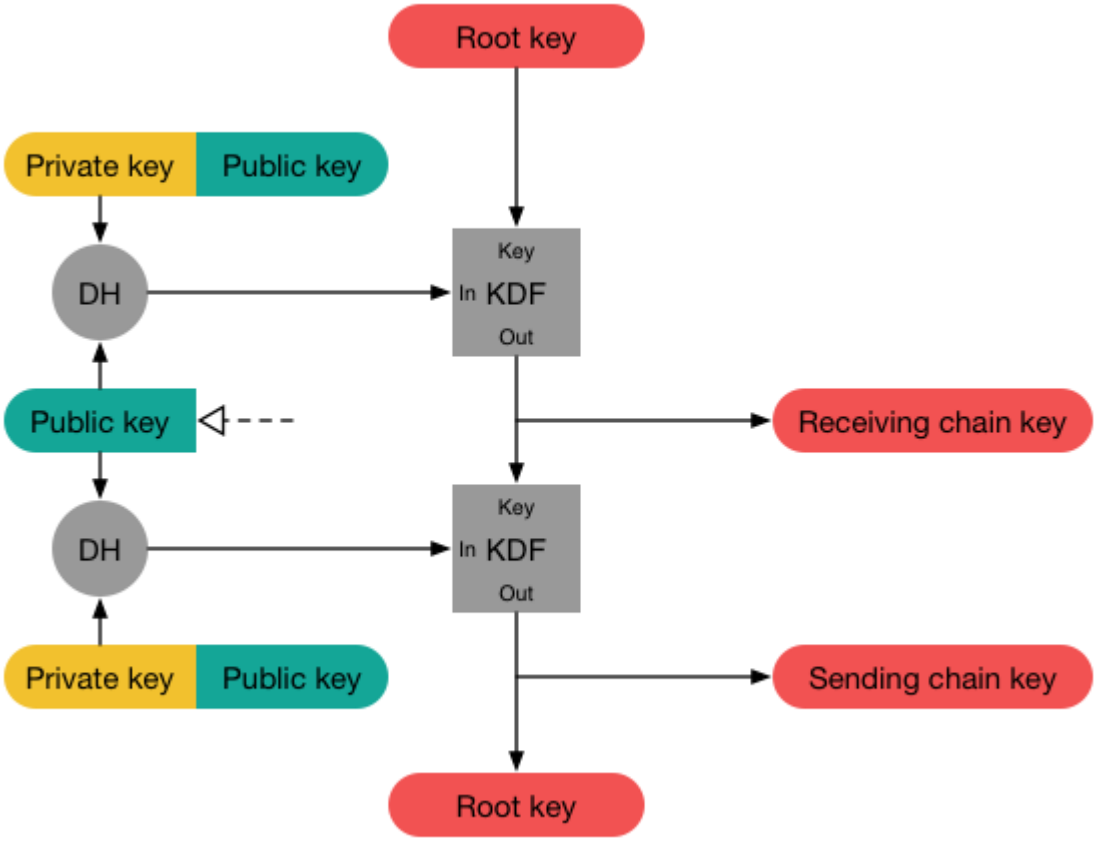
По мере того, как стороны поочередно выполняют шаги храповика DH, они поочередно вводят новые цепочки отправки:



СЛАЙД 10 ==========

Однако приведенная выше картина является упрощением. Вместо того чтобы получать ключи цепочки непосредственно из выходов DH, выходы DH используются в качестве входов KDF в корневую цепочку, а выходы KDF из корневой цепочки используются в качестве ключей цепочки для отправки и получения. Использование цепочки KDF повышает устойчивость к взлому и восстанавливает работоспособность.

Таким образом, полный шаг храповика DH состоит из двойного обновления корневой цепочки KDF и использования выходных ключей KDF в качестве новых ключей цепочки приема и отправки:



СЛАЙД 11 ==========

2.4. Двойной храповик

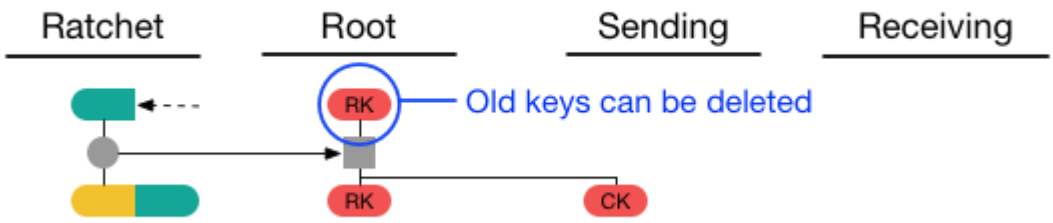
Сочетание симметрично-ключевого и DH-храповика дает Двойной храповик:

- Когда сообщение отправляется или принимается, к цепочке отправителей или получателей применяется шаг храповика симметричного ключа для получения ключа сообщения.
- Когда получен новый открытый ключ храповика, перед храповиком симметричных ключей выполняется шаг храповика DH для замены цепных ключей.

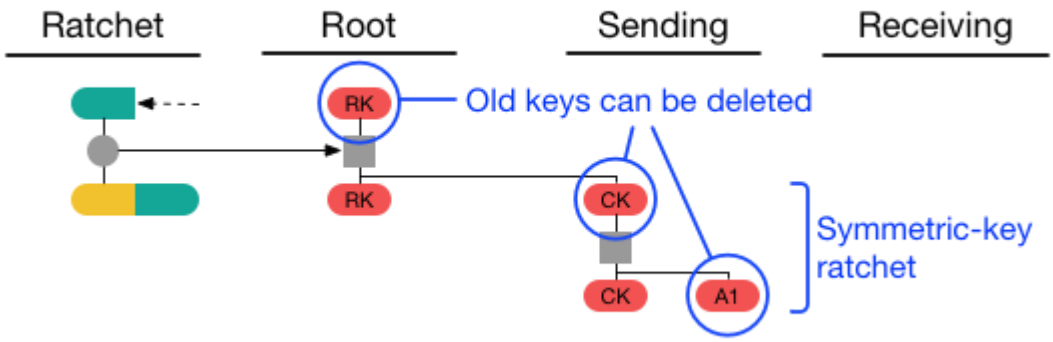
ДОПОЛНИТЕЛЬНОЕ ==========

На приведенной ниже схеме Алиса была инициализирована с открытым ключом храповика Боба и общим секретом, который является начальным корневым ключом. В процессе инициализации Алиса генерирует новую пару

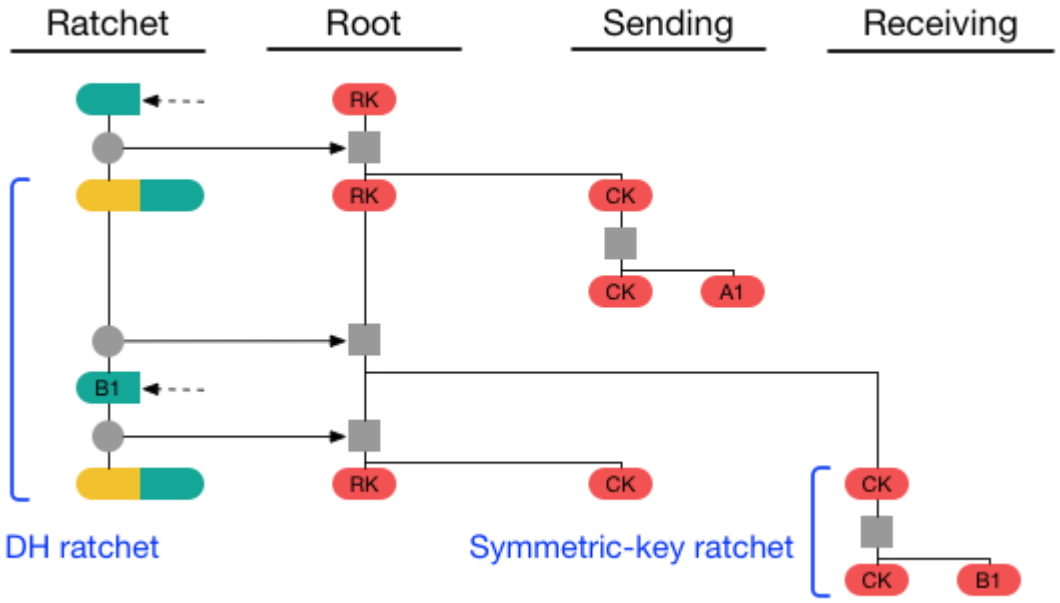
храповых ключей и передает результат DH в корневой KDF для вычисления нового корневого ключа (*RK*) и ключа цепочки отправки (*CK*):



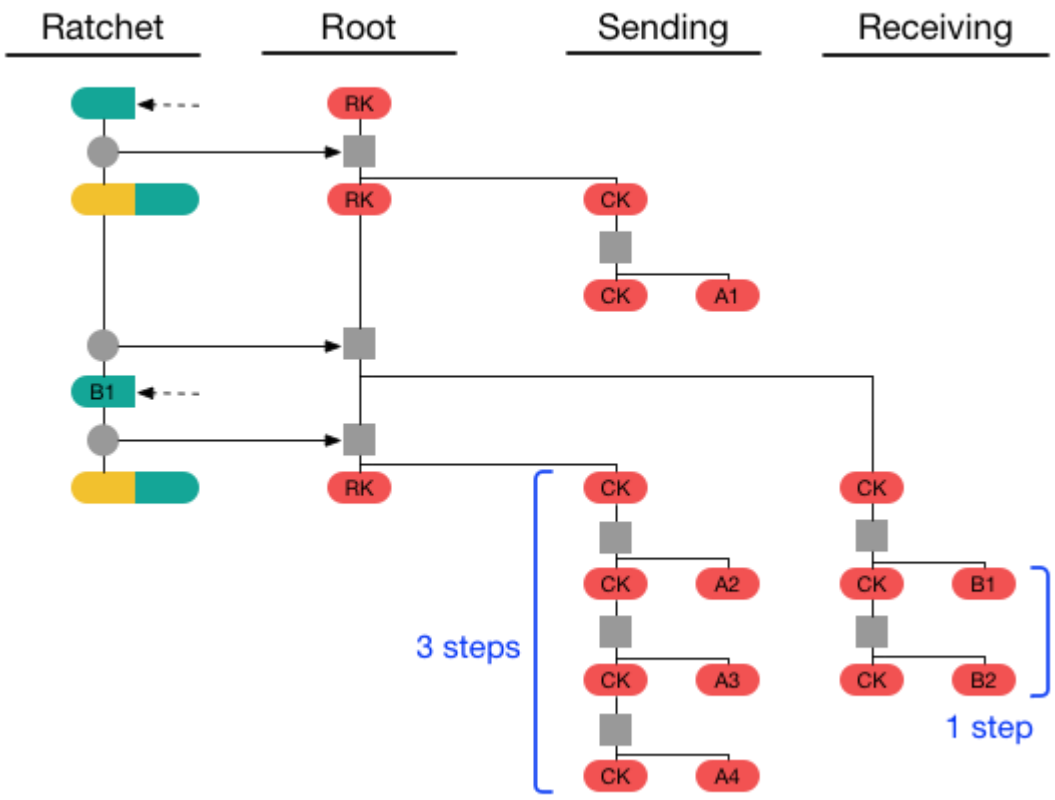
Когда Алиса отправляет свое первое сообщение *A1*, она применяет шаг храповика симметричного ключа к своему цепному ключу отправки, в результате чего получается новый ключ сообщения (ключи сообщений будут помечены сообщением, которое они шифруют или расшифровывают). Новый цепной ключ сохраняется, а ключ сообщения и старый цепной ключ могут быть удалены:



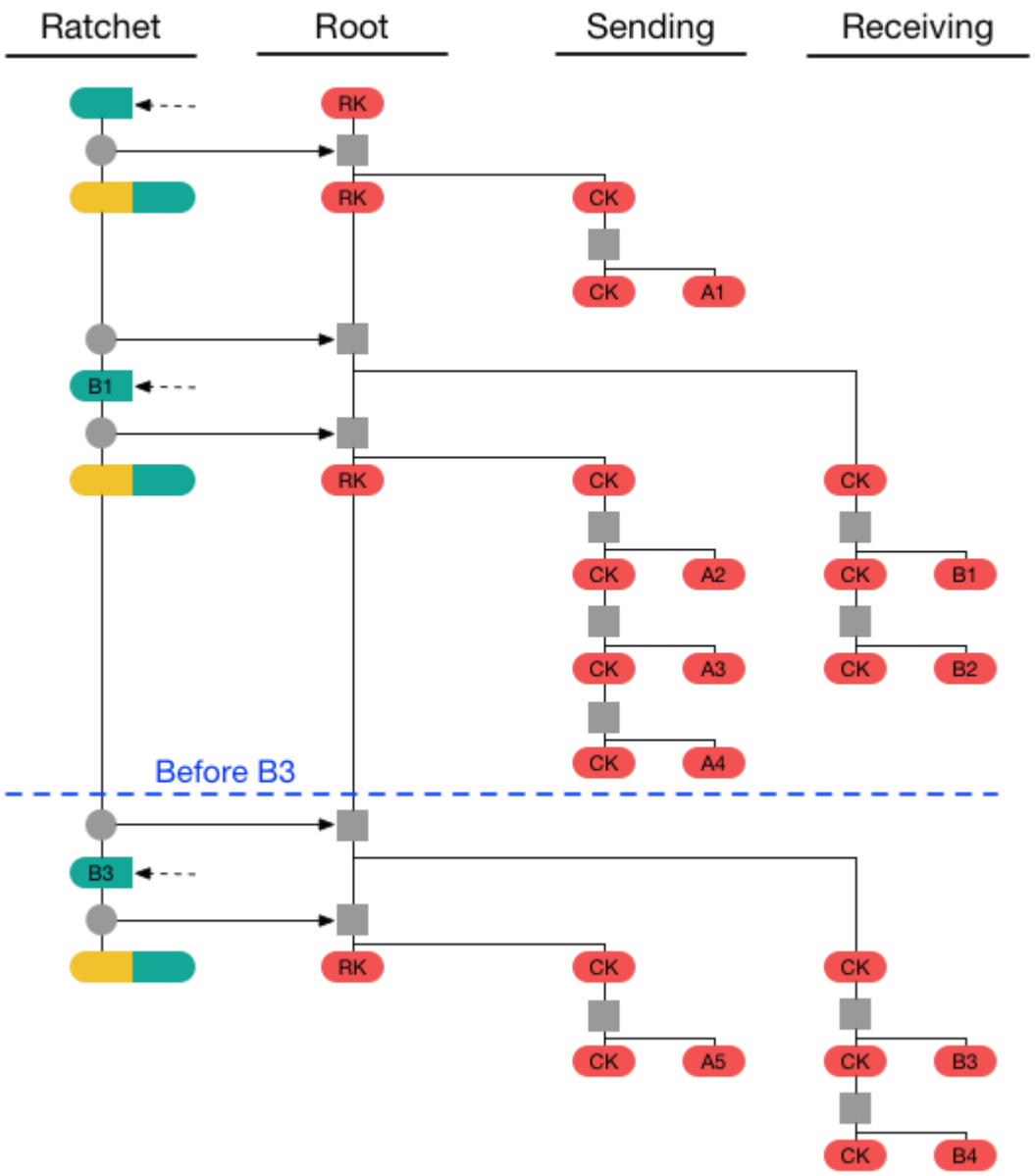
Если Алиса в следующий раз получит ответ *B1* от Боба, он будет содержать новый открытый ключ с храповиком (открытые ключи Боба помечены сообщением, когда они были впервые получены). Алиса применяет шаг храповика DH для получения новых цепных ключей получения и отправки. Затем она применяет шаг храповика с симметричным ключом к цепочке получения, чтобы получить ключ сообщения для полученного сообщения:



Предположим, что Алиса отправляет сообщение *A2*, получает сообщение *B2* со старым открытым ключом Боба с храповиком, затем отправляет сообщения *A3* и *A4*. Цепочка отправки Алисы будет храниться три шага, а цепочка получения - один:



Предположим, что Алиса получает сообщения *B3* и *B4* со следующим ключом храповика Боба, а затем отправляет сообщение *A5*. Окончательное состояние Алисы будет следующим:



2.6. Сообщения не по порядку

Double Ratchet обрабатывает потерянные или не по порядку сообщения, включая в заголовок каждого сообщения номер сообщения в цепочке отправки (N=0,1,2,...) и длину (количество ключей сообщений) в предыдущей цепочке отправки (PN). Это позволяет получателю перейти к соответствующему ключу сообщения, сохраняя пропущенные ключи на случай, если пропущенные сообщения придут позже.

При получении сообщения, если срабатывает шаг храповика DH, то полученное *значение* PN минус длина текущей цепочки приема - это

количество пропущенных сообщений в этой цепочке приема. Полученное *N* - это количество пропущенных сообщений в новой цепочке приема (т. е. в цепочке после храповика DH).

Если шаг храповика DH не срабатывает, то полученное значение *N* минус длина принимающей цепочки - это количество пропущенных сообщений в этой цепочке.

Например, рассмотрим последовательность сообщений из предыдущего раздела, когда сообщения *B2* и *B3* пропущены. Сообщение *B4* запустит храповой шаг DH Алисы (вместо *B3*). Сообщение *B4* будет иметь *PN=2* и *N=1*. При получении *B4* у Алисы будет цепочка приема длины 1 (*B1*), поэтому Алиса будет хранить ключи сообщений *B2* и *B3*, чтобы их можно было расшифровать, если они придут позже: