**Алгоритмы разделения секрета.** Четыре приведенных ниже различных алгоритма представляют собой частные случаи общего теоретического подхода.

**Схема интерполяционных многочленов Лагранжа.**

Для создания пороговой схема А Шамир воспользовался уравнениями для многочленов в конечном поле . Выберем простое число p, которое больше количества возможных теней и больше самого большого из возможных секретов. Чтобы сделать секрет общим, сгенерируем произвольный многочлен степени m-1. Например, если нужно создать пороговую схему (3, n) (для восстановления M потребуется три тени), генерируется квадратичный многочлен $(ax^2 + bx + M) \bmod p$, где p - это случайное простое число, большее любого из коэффициентов . Коэффициенты a и b выбираются случайным образом, они хранятся в тайне и отбрасываются после того, как распределяются тени . M - это сообщение. Простое число должно быть опубликовано. :

Тени получаются с помощью вычисления многочлена в n различных точках k i =F($x_i$) Другими словами, первой тенью может быть значение многочлена при x = 1, второй тенью - значение многочлена при x = 2, и т.д. Так как в квадратичных многочленах три неизвестных коэффициента, a, b и M, для создания трех уравнений можно использовать любые три цели. Одной или двух теней не хватит, а четырех или пяти теней будет много .

Например, пусть M равно 11. Чтобы создать пороговую схему (3, 5), в которой любые трое из пяти человек могут восстановить M, сначала получим квадратичное уравнение (7 и 8 - случайно выбранные числа chosen randomly): $F(x) = (7x^2 + 5x + 11) \bmod 13$

Пятью тенями являются:

$k1 = F(1) = 7 + 8 + 11 \equiv 0 \pmod{13}$

$k\ 2 = F(2) = 28 + 16 + 11 \equiv 3 \pmod{13}$

$k3 = F(3) = 63 + 24 + 11 \equiv 7 \pmod{13}$

$k\ 4 = F(4) = 112 + 32 + 11 \equiv 12 \pmod{13}$

$k\ 5 = F(5) = 175 + 40 + 11 \equiv 5 \pmod{13}$

Чтобы восстановить M по трем теням, например, k 2, k 3 и k 5, решается система линейных уравнений:

$a*2^2 + b*2 + M = 3 \pmod{13}$

$a*3^2 + b*3 + M = 7 \pmod{13}$

$a*5^2 + b*5 + M = 5 \pmod{13}$

Решением будут a = 7, b = 8 и M = 11. Итак, M получено.

Эту схему разделения можно легко реализовать для больших чисел . Если вы хотите разбить сообщение на 30 равных частей так, чтобы восстановить сообщение можно было, объединив любые шесть из них , выдайте каждому из 30 человек значения многочлена пятой степени. $F(x) = ax^5 + bx^4 + cx^3 + dx^2 + ex + M \pmod{p}$

Шесть человек могут шесть неизвестных (включая M), но пятерым не удастся узнать ничего об M.

Наиболее впечатляющим моментом совместного использования секрета является то, что, если коэффициенты выбраны случайным образом, пять человек даже при помощи бесконечных вычислительных мощностей не смогут узнать ничего, кроме длины сообщения (которая и так им известна).

Это также безопасно, как одноразовый блокнот, попытка выполнить исчерпывающий поиск (то есть, перебор всех возможных шестых теней) покажет, что любое возможное сообщение останется секретным.

**Векторная схема.** Джордж Блэкли (George Blakley) изобрел схему, использующую понятие точек в пространстве. Сообщение определяется как точка в m-мерном пространстве. Каждая тень - это уравнение (m-1)-мерной гиперплоскости, содержащей эту точку. Например, если для восстановления сообщения нужны три тени, то оно является точкой в трехмерном пространстве. Каждая тень представляет собой иную плоскость. Зная одну тень, можно утверждать, что точка находится где-то на плоскости. Зная две тени - что она находится где-то на линии пересечения двух плоскостей . Зная три тени, можно точно определить, что точка находится на пересечении трех плоскостей .

Asmuth-Bloom В этой схеме используются простые числа.

Для (m, n)-пороговой схемы выбирается большое простое число p, большее M.

Затем выбираются числа, меньшие p - $d_1$, $d_2$, . . . $d_n$, для которых: 1. Значения d i упорядочены по возрастанию, d i < $d_{i+1}$

2. Каждое d i взаимно просто с любым другим d i
3. $d_1 * d_2 * \ldots * d_m > p * d_{n-m+2} * d_{n-m+3} * \ldots * d_n$

Чтобы распределить тени, сначала выбирается случайное число r и вычисляется M' = M + r p. Тенями k i, являются k i = M' mod d i Объединив любые m теней, можно восстановить M, используя китайскую теорему об остатках, но это невозможно с помощью любых m-1 теней.

Karnin-Greene-Hellman В этой схеме используется матричное умножение.

**Более сложные пороговые схемы**

В предыдущих примерах показаны только простейшие пороговые схемы : секрет делится на n теней так, чтобы, объединив любые m из них, можно было раскрыть секрет. На базе этих алгоритмов можно создать намного более сложные схемы. Часто использу-ется алгоритм Шамира, хотя работают и все остальные.

Чтобы создать схему, в которой один из участников важнее других, ему выдается больше теней . Если для восстановления секрета нужно пять теней, и у кого-то есть три тени, а у всех остальных - по одной , этот человек вместе с любыми двумя другими может восстановить секрет. Без его участия для восстановления секрета потребуется пять человек. По несколько теней могут получить два человека и более. Каждому человеку может быть выдано отличное число теней. Независимо от того, сколько теней было роздано, для восстановления секрета потребуется любые m из них. Ни один человек, ни целая группа не смогут восстановить секрет, обладая только m-1 тенями. Для других схем представим сценарий с двумя враждебными делегациями . Можно распределить секрет так, чтобы для его восстановления потребовалось двое из 7 участников делегации A и трое из 12 участников делегации B. Создается многочлен степени 3, который является произведением линейного и квадратного выражений . Каждому участнику делегации A выдается тень, которая является значением линейного выражения, а участникам делегации B выдаются значения квадратичного выражения.

Для восстановления линейного выражения достаточны любые две тени участников делегации  A, но независимо от того, сколько других теней есть у делегации, ее участники не смогут ничего узнать о секрете.
Аналогично для делегации B: ее участники могут сложить три тени, восстанавливая квадратное выражение, но другую информацию, необходимую для восстановления секрета в целом, они получить не смогут .
Только перемножив свои выражения, участники двух делегаций смогут восстановить секрет . В общем случае, может быть реализована любая мыслимая схема разделения секрета .
Потребуется только написать систему уравнений, соответствующих конкретной системе.
Разделение секрета с мошенниками Этот алгоритм изменяет стандартную пороговую схему  (m, n) для обнаружения мошенников.

**Подсознательный канал** Ong-Schnorr-Shamir, разработанный Густавусом Симмонсом (Gustavus Simmons), использует схему идентификации Ong-Schnorr-Shamir: отправитель (A) выбирает общедоступный модуль n и закрытый ключ k так, чтобы n и k были взаимно простыми числами. В используется совместно А и Б, получателем в подсознательном канале. Открытый ключ вычисляется следующим образом: $h = -k^2 \mod n$

Если А нужно отправить подсознательное сообщение M в безобидном сообщении M', она сначала проверяет, что пары M' и n, а также M и n являются взаимно простыми числами.

А вычисляет $S\ 1 = 1/2*((M'/M + M)) \mod n$

$S\ 2 = 1/2*((M'/M - M)) \mod n$

Пара чисел S1 и S2 представляет собой подпись в традиционной схеме Ong-Schnortr-Shamir и одновременно является носителем подсознательного сообщения.

Другой предложенный Симмонсом подсознательный канал основан на схеме подписи ElGamal

DSA Подсознательный канал существует и в DSA. На самом деле их даже может быть несколько. Простейший подсознательный канал включает выбор k. Предполагается, что это будет 160-битовое число. Однако, если А выбирает конкретное k, то Б, зная закрытый ключ А, сможет раскрыть это k.

А посылает Б 160-битовое подсознательное сообщение в каждой подписи DSA, а все остальные будут только проверять подпись А. Дополнительное усложнение: Так как k должно быть случайным, А и Б должны использовать общий одноразовый блокнот и шифровать подсознательное сообщение с помощью этого блокнота, генерируя k.

**Неотрицаемые цифровые подписи** Автором этого алгоритма неотрицаемой подписи является Дэвид Чаум (David Chaum) Сначала опубликовываются большое простое число p и примитивный элемент g, которые будут совместно использоваться группой подписывающих. У А есть закрытый ключ x и открытый ключ $g^x \bmod p$. Чтобы подписать сообщение, А вычисляет $z = m^x \bmod p$. Это все, что А нужно сделать..

Проверка подписи немного сложнее.

1) Б выбирает два случайных числа, a и b, меньшие p, и отправляет A: $c = z^{a} (g^{x})^{b} \mod p$
2) A вычисляет $t=x^{-1} \mod (p-1)$, и отправляет Б: $d = c^{t} \mod p$
3) Б проверяет, что $d \equiv m^{a} g^{b} \pmod p$ Если это так, он считает подпись истинной.

Алгоритм для преобразуемых неотрицаемых подписей, которые можно проверять, отменять и преобразовывать в обычные неотрицаемые подписи основан на алгоритме цифровых подписей El- Gamal.

Схемы неотрицаемых подписей можно объединить со схемами разделения секрета, создав распределенные преобразуемые неотрицаемые подписи. Кто-нибудь может подписать сообщение, а затем распределить возможность подтверждения правильности подписи. Он может, например, потребовать, чтобы в протоколе убеждения Б в правильности подписи участвовали трое из пяти обладателей возможность подтверждения правильности.

Можно улучшить алгоритмы, позволяющие отказаться от необходимости доверенного лица - распределителя.

**Подписи, подтверждаемые доверенным лицом** Вот как А может подписать сообщение, а Б проверить его так , чтобы и К немного позже могла доказать Ду правильность подписи А.

Сначала опубликовываются большое простое число p и примитивный элемент g, которые будут совместно использоваться группой пользователей. Также опубликовывается n, произведение двух простых чисел. У К есть закрытый ключ z и открытый ключ $h = g^x \bmod p$. В этом протоколе А может подписать m так, чтобы Б мог проверить правильность ее подписи, но не мог убедить в этом третью сторону.

1) А выбирает случайное x и вычисляет $a = g^x \bmod p$

$b = h^x \bmod p$ А вычисляет хэш-значение m, H(m), и хэш-значение объединения a и b, H(a,b), а затем

$j = (H(m) \oplus H(a, b))^{1/3} \bmod n$ и посылает a, b и j -Б.

2) Б выбирает два случайных числа, s и t, меньших p, и посылает А

$$c = g^s h^t \bmod p$$

3) А выбирает случайное q, меньшее p, и посылает Б

$$d = g^q \bmod p \quad e = (cd)^x \bmod p$$

4) Б посылает А s и t.

5) А проверяет, что $g^s h^t \equiv c \pmod p$ затем она посылает Б q.

6) Б проверяет $d \equiv g^q \bmod p \quad e/a^q \equiv a^s b^t \pmod p$

$$(H(m) \oplus H(a,b)) = j^{1/3} \bmod n$$

Если все тождества выполняются, то Б считает подпись истинной . Б не может использовать запись этого доказательства для убеждения Д в истинности подписи , но Д может выполнить протокол с доверенным лицом А, К. Вот как К убеждает Д в том, что a и b образуют правильную подпись.

1) Д выбирает случайные u и v, меньшие p, и посылает К

$$k = g^u a^v \bmod p$$

2) К выбирает случайное w, , меньшее p, и посылает Д

$$l = g^w \bmod p \quad y = (kl)^z \bmod p$$

(3) Д посылает К u и v.

4) К проверяет, что $g^u a^v \equiv k \pmod p$

Затем она посылает Д w.

5) Д проверяет, что $g^w \equiv l \pmod p$

$y/h^w \equiv h^u b^v \pmod p$

Если все тождества выполняются, то Д считает подпись истинной.

В другом протоколе К может преобразовать протокол доверенного лица в обычную цифровую подпись .

**Вычисления с зашифрованными данными**

Проблема дискретного логарифма Существует большое простое число p и генератор g. А хочет для конкретного x найти такое e, для которого $g^e \equiv x \pmod p$. Это трудная проблема, и А не хватает вычислительных мощностей для вычисления результата . У Б есть такие возможности - он представляет правительство, или мощный вычислительный центр, или еще какую-нибудь влиятельную организацию. Вот как А может получить помощь Б, не раскрыв ему x :

1) А выбирает случайное число r, меньшее p.
2) А вычисляет $x' = xg^r \bmod p$
3) А просит Б решить $g^{e'} \equiv x' \pmod p$
4) Б вычисляет e' и посылает его А.
5) А восстанавливает e, вычисляя $e = (e' - r) \bmod (p - 1)$

Аналогичные протоколы используются для проблем квадратичных остатков и примитивных корней.

**Бросание "честной" монеты: с помощью квадратных корней, с помощью возведения в степень по модулю p, с помощью целых чисел Блюма.**

Существует простая функция **однонаправленного сумматоры** $A(x_i, y) = x_{i-1}^{\ y} \bmod n$. Числа n (являющееся произведением двух простых чисел) и x0 должны быть заранее согласованы. Тогда суммированием $y_1$, $y_2$ и $y_3$ будет $((x_0^{y0} \bmod n)^{y2} \bmod n)^{y3} \bmod n$

Это вычисление не зависит от порядка $y_1$, $y_2$ и $y_3$.

**Раскрытие секретов "все или ничего"**

Фиксированным битовым индексом (fixed bit index, FBI) x и y называется последовательность номеров совпадающих битов этих строк.

Например: x = 110101001011

y = 101010000110

FBI(x, y) = {1, 4, 5, 11}

Честные и отказоустойчивые криптосистемы

Честная схема Diffie-Hellman

Отказоустойчивая схема Diffie-Hellman

**Доказательство с нулевым знанием для дискретного логарифма**
**Доказательство с нулевым знанием для возможности вскрыть RSA**
**Слепые подписи** использует алгоритм RSA.
**Передача с забыванием**
**Безопасные вычисления с несколькими участниками**
Понятие **вероятностного шифрования** было изобретено Шафи Голдвассером (Shafi Goldwasser) и Сильвией Микали. Идеей вероятностного шифрования является устранение утечки информации в криптографии  с открытыми ключами. Так как криптоаналитик всегда может расшифровать случайные сообщения открытым ключом, он может получить некоторую информацию. При условии, что у него есть шифротекст $C = E_K(M)$, и он пытается получить открытый текст M, он может выбрать случайное сообщение  M' и зашифровать его: $C' = E_K(M')$. Если C' = C, то он угадал правильный открытый текст. В противном случае он делает следующую попытку .

Кроме того, вероятностное шифрование позволяет избежать даже частичной утечки информации об оригинальном сообщении. При использовании криптографии с открытыми ключами криптоаналитик иногда может узнать кое-что о битах: XOR 5-го, 17-го и 39-го битов равно 1, и т.п. Цель этого метода состоит в том, чтобы ни вычисления, проводимые над шифротекстом, ни проверка любых других открытых текстов не смогли дать криптоаналитику никакой информации о соответствующем открытом тексте.

При вероятностном шифровании алгоритм шифрования является вероятностным, а не детерминированным . Другими словами, многие шифротексты при расшифровке дают данный открытый текст , и конкретный шифротекст, используемый в любом конкретном шифровании, выбирается случайным образом.

$C_1 = E_K(M), C_2 = E_K(M),...,C_i = E_K(M)$

$M = D_K(C_1) = D_K(C_2) = D_K(C_3) = . . . = D_K(C_i)$

При вероятностном шифровании криптоаналитику больше не удастся шифровать произвольные открытые тексты в поисках правильного шифротекста. Для иллюстрации пусть у криптоаналитика есть шифротекст $C\,i = E_K(M)$.

Даже если он правильно угадает M, полученный при шифровании $E_K(M)$ результат будет совершенно другим шифротекстом C: C j. Сравнивая C i и C j, он не может по их совпадению определить правильность своей догадки. Даже если у криптоаналитика есть открытый ключ шифрования, открытый текст и шифротекст, он  не может без закрытого ключа дешифрирования доказать, что шифротекст является результатом шифрования конкретного открытого текста. Даже выполнив исчерпывающий поиск, он может доказать только, что каждый возможный открытый текст является возможным открытым текстом . В этой схеме шифротекст всегда будет **больше** открытого текста. Этого невозможно избежать, это является результатом того, что многие шифротексты расшифровываются в один и тот же открытый текст . В первой схеме вероятностного шифрования шифротекст получался настолько больше открытого текста, что он был бесполезным. Однако Мануэль Блюм (Manual Blum) и Голдвассер (Goldwasser) получили эффективную реализацию вероятностного шифрования с помощью генератора псевдослучайных битов  Blum Blum Shub (BBS)

Генератор BBS основан на теории квадратичных остатков . Существуют два простых числа, p и q, конгруэнтных 3 по модулю 4. Это закрытый ключ. Их произведение, pq = n, является открытым ключом. (Запомните свои p и q, безопасность схемы опирается на сложность разложения n на множители.) Для шифрования сообщения M сначала выбирается случайное x, взаимно простое с n. Затем вычисляется $x_0 = x^2 \mod n$

$x_0$ служит стартовой последовательностью для генератора псевдослучайных битов BBS, а выход генератора используется в качестве потокового шифра. Побитно выполняется XOR M с выходом генератора. Генератор выдает биты b i (младший значащий бит xi, где $x_i = x_{i-1}^2 \mod n$), поэтому

$M=M_1, M_2, M_3, \ldots M_t$

$c = M_1 \oplus b_1, M_2 \oplus b_2, M_3 \oplus b_3, \ldots M_t \oplus b_t$

где t - это длина открытого текста. Добавьте последнее вычисленное значение, xt, к концу сообщения, и дело сделано. Расшифровать это сообщение можно только одним способом - получить x0 и с этой стартовой последовательностью запустить генератор BBS, выполняя XOR выхода с шифротекстом.

Так как генератор BBS безопасен влево, значение $x_t$ бесполезно для криптоаналитика. Только тот, кому известны p и q, может расшифровать сообщение. При наличии $x_0$ дешифрирование несложно. Просто задайте стартовую последовательность генератора BBS и выполните XOR результата с шифротекстом.

Эту схему можно сделать еще быстрее, используя все известные безопасные биты xi, а не только младший значащий бит. С таким улучшением вероятностное шифрование Blum-Goldwasser оказывается быстрее RSA и не допускает утечки информации об открытом тексте. Кроме того, можно доказать, что сложность вскрытия этой схемы равна сложности разложения n на множители. С другой стороны, эта схема совершенно небезопасна по отношению к вскрытию с выбранным шифротекстом. По младшим значащим битам правильных квадратичных остатков можно вычислить квадратный корень любого квадратичного остатка. Если это удастся, то удастся и разложение на множители.

**Квантовая криптография** вводит естественную неопределенность квантового мира. С ее помощью можно создавать линии связи, которые невозможно послушать, не внося помех в передачу . Законы физики надежно защищают такой квантовый канал, даже если подслушивающий может предпринимать любые действия, даже если он имеет доступ к неограниченной вычислительной мощности, даже если P = NP. Шарль Бенне (Charles Bennett), Жиль Брассар (Gilles Brassard), Клод Крепо (Claude Crepeau) и другие расширили эту идею, описав квантовое распределение ключей, квантовое бросание монеты, квантовое вручение бита , квантовую передачу с забыванием и квантовые вычисления с несколькими участниками . В соответствии с законами квантовой механики частицы на самом деле не находятся в одном месте, а с определенной вероятностью существуют сразу во многих местах . Однако это так только до тех пор, пока не приходит ученый и не обмеряет частицу, "оказавшуюся" в данном конкретном месте . Но измерить все параметры частицы (например, координаты и скорость) одновременно невозможно.

Если измерить одну из этих двух величин, сам акт измерения уничтожает всякую возможность измерить другую величину. Неопределенность является фундаментальным свойством квантового мира, и никуда от этого не денешься. Эту неопределенность можно использовать для генерации секретного ключа . Путешествуя, фотоны колеблются в определенном направлении, вверх-вниз, влево-вправо, или, что более вероятно, под каким-то углом . Обычный солнечный свет неполяризован, фотоны колеблются во всех возможных направлениях . Когда направление колебаний многих фотонов совпадает, они являются поляризованными. Поляризационные фильтры пропускают только те фотоны, которые поляризованы в определенном направлении, а остальные блокируются . Например, горизонтальный поляризационный фильтр пропускает только фотоны с горизонтальной поляризацией. Повернем этот фильтр на 90 градусов, и теперь сквозь него будут проходить только вертикально поляризованные фотоны.

Пусть у вас есть импульс горизонтально поляризованных фотонов . Если они попробуют пройти через горизонтальный фильтр, то у них у всех прекрасно получится. Если медленно поворачивать фильтр на 90 градусов, количество пропускаемых фотонов будет становиться все меньше и меньше, и наконец ни один фотон не пройдет через фильтр. Это противоречит здравому  смыслу. Кажется,  что даже незначительный поворот  фильтра должен остановить все фотоны, так как они горизонтально поляризованы . Но в квантовой механике каждая частица с определенной вероятностью может изменить свою поляризацию и проскочить через фильтр . Если угол отклонения фильтра невелик, эта вероятность высока, а если он равен 90 градусам, то вероятность равна нулю . А если угол поворота фильтра равен 45 градусам, вероятность фотона пройти фильтр равна 50 процентам. Поляризацию можно измерить в любой системе координат: двух направлениях, расходящихся под прямым углом. Примерами систем координат являются прямоугольная - горизонтальное и вертикальное направления – и диагональная - левая и правая диагонали.

Если импульс фотонов поляризован в заданной системе координат, то при измерении в той же системе координат вы узнаете поляризацию. При измерении в неправильной системе координат, вы получите случайный результат. Мы собираемся использовать это свойство для генерации секретного ключа:

1) А посылает Б последовательность фотонных импульсов. Каждый из импульсов случайным образом поляризован в одном из четырех направлений: горизонтальном, вертикальном, лево- и праводиагональном. Например, А посылает Б:

| | / — \ — | — /

2) У Б есть детектор поляризации. Он может настроить свой детектор на измерение прямоугольной или диагональной поляризации. Одновременно мерить и ту, и другую у него не получится, ему не позволит квантовая механика. Измерение одной поляризации не даст измерить другую. Итак, он устанавливает свои детекторы произвольным образом:

X + + X X X + X + +

Теперь, если Б правильно настроит свой детектор, он зарегистрирует правильную поляризацию. Если он настроит детектор на измерение прямоугольной поляризации, и импульс будет поляризован прямоугольно, он узнает, какую поляризацию фотонов выбрала А. Если он настроит детектор на измерение диагональной поляризации, а импульс будет поляризован прямоугольно, то результат измерения будет случайным. Б не сможет определить разницу. В приведенном примере он может получить следующий результат:

/ | — \ / \ — / — |

3) Б сообщает А по незащищенному каналу, какие настройки он использовал .

4) А сообщает Б, какие настройки были правильными. В нашем примере детектор был правильно установлен для импульсов 2, 6, 7 и 9.

5) А и Б оставляют только правильно измеренные поляризации. В нашем примере они оставляют: * | * * * \ — * — * С помощью заранее приготовленного кода А и Б преобразуют в биты эти результаты измерений поляризации

Например, горизонтальная и леводиагональная могут означать единицу, а вертикальная и праводиагональная - ноль. В нашем примере они оба получат: 0 0 1 1

Итак, А и Б получили четыре бита. С помощью этой системы они могут генерировать столько битов, сколько им нужно. В среднем Б правильно угадывает в 50 процентах случаев, поэтому для генерации n битов А придется послать 2n фотонных импульсов. Они могут использовать эти биты как секретный ключ симметричного алгоритма или обеспечить абсолютную безопасность, получив достаточно битов для использования в качестве одноразового блокнота.

Замечательным является то, что Е не сможет подслушать. Как и Б, ей нужно угадать тип измеряемой поляризации, и, как и у Б, половина ее догадок будет неправильной. Так как неправильные измерения изменяют поляризацию фотонов, то при подслушивании она неминуемо вносит ошибки в передачу .

Если это так, А и Б получат различные битовые последовательности.

Итак, А и Б заканчивают протокол подобными действиями:
(6) А и Б сравнивают несколько битов своих строк. По наличию расхождений они узнают о подслушивании.
Если строки не отличаются, то они отбрасывают использованные для сравнения биты и используют оставшиеся. Улучшения этого протокола позволяют А и Б использовать свои биты даже в присутствии Е. Они могут сравнивать только четность битовых подмножеств. Тогда, если не обнаружено расхождений, им придется отбросить только один бит подмножества. Это обнаруживает подслушивание с вероятностью 50 процентов, но если они сверят таким образом n различных битовых подмножеств, вероятность Е подслушать и остаться незамеченной будет равна
$1/2^{n}$ . В квантовом мире не бывает пассивного подслушивания. Если Е попытается раскрыть все биты, она обязательно разрушит канал связи. Бенне и Брассар построили работающую модель квантового распределения ключей и обменялись безопасными битами на оптической скамье.

Протокол управления секретными ключами компании IBM В конце 70-х годов IBM, используя только симметричную криптографию, разработала законченную систему управления ключами для передачи данных и безопасности файлов в компьютерных сетях  Не так важны реальные механизмы протокола, как его общая философия : за счет автоматизации генерации, распределения, установки, хранения, изменения и разрушения ключей этот протокол далеко продвинулся, обеспечивая безопасность лежащих в его основе криптографических алгоритмов . Этот протокол обеспечивает три вещи: безопасную связь между сервером и различными терминалами, безопасное хранение файлов на сервере и безопасную связь между серверами. Протокол не обеспечивает настоящего прямого соединения терминал-терминал, хотя его модификация может реализовать такую возможность . Каждый сервер сети подключен к криптографической аппаратуре, которая выполняет все шифрование и дешифрирование. У каждого сервера есть **Главный ключ** (Master Key), $KM_0$, и два варианта, $KM_1$ и $KM_2$, которые являются упрощенными вариантами $KM_0$.

Эти ключи используются для шифрования других ключей и для генерации новых ключей. У каждого терминала есть Главный ключ терминала (Master Terminal Key), KMT, который используется для обмена ключами с другими терминалами. KMT хранятся на серверах, зашифрованные ключом $KM_1$. Все остальные ключи, например, используемые для шифрования файлов ключей (они называются KNF), хранятся в зашифрованной форме, закрытые ключом $KM_2$. Главный ключ $KM_0$ хранится в энергонезависимом модуле безопасности. Сегодня это может быть либо ключ в ПЗУ, либо магнитная карточка, или ключ может вводиться пользователем с клавиатуры (возможно как текстовая строка, преобразуемая в ключ ). $KM_1$ и $KM_2$ не хранятся где-нибудь в системе, а, когда понадобится, вычисляются по $KM_0$. Сеансовые ключи для связи между серверами генерируются на сервере с помощью псевдослучайного процесса. Аналогичным образом генерируются ключи для шифрования хранимых файлов (KNF). Сердцем протокола служит устойчивый к вскрытию модуль, называемый криптографической аппаратурой (cryptographic facility).

И на сервере, и на терминале все шифрование и дешифрирование происходи именно в этом модуле. В этом модуле хранятся самые важные ключи, используемые для генерации действительных ключей шифрования. После того, как эти ключи записаны, считать их становится невозможным . Кроме того, они помечены для конкретного использования: ключ, предназначенный для решения одной задачи, не может случайно быть использован для решения другой. Эта концепция векторов управления ключами возможно является самым значительным достижением этой системы . Модификация этой схемы построена на базе сетевых узлов с аппаратурой проверки подлинности ключей, которая обслуживает локальные терминалы . Эта модификация была разработана, чтобы:
- обезопасить дуплексный канал между двумя пользовательскими терминалами .
— Обезопасить связь с помощью шифрованной почты .
— Обеспечить защиту личных файлов.
— Обеспечить возможность цифровой подписи.

Для связи и передачи файлов между пользователями в этой схеме используются ключи, генерированные в аппаратуре проверки подлинности ключей, отправляемые пользователям после шифрования с помощью главного ключа. Информация о личности пользователя встраивается в ключ, предоставляя доказательство того, что сеансовый ключ используется конкретной парой пользователей . Возможность проверки подлинности ключей является главной в этой системе. Хотя в системе не используется криптография с открытыми ключами, она поддерживает возможность, похожую на цифровую подпись : ключ может быть прислан только из конкретного источника и прочитан только в конкретном месте назначения .