15) Кодирование с открытым ключом

Ключ, с помощью которого сообщение шифруется, называется открытым (public key).

Ключ для дешифровки называется секретным или приватным (private key).

Основатели RSA (1977): Роналд Ривест, Ади Шамир, Леонард Адлеман.

Первая идея схемы RSA – это рассмотрение блоков текста, состоящих из нескольких последовательных байтов, как элементов кольца вычетов по модулю n. Объединяя двоичные записи всех байтов блока, получают длинное двоичное слово, которое трактуется как двоичная запись целого числа. Это число, в свою очередь, рассматривается как элемент кольца вычетов по модулю n. Таким образом, блоки текста отождествляются с элементами кольца Z_n. Шифрование состоит в преобразовании элементов Z_n. **Число n в схеме RSA достаточно большое, не менее 200 десятичных знаков**. (Это очень важно, что блок как число < n.)

При использовании таких систем каждый участник переговоров имеет **открытый ключ** и **секретный ключ**. В системе RSA ключ состоит из двух целых чисел. Участников переговоров может быть несколько, но для примера мы будем говорить о переговорах Алисы (A) и Боба (B). Их открытые ключи мы будем обозначать P_A и P_B , а секретные - S_A и S_B.

Каждый участник создает два своих ключа. Секретный ключ он хранит в тайне, а открытый сообщает остальным участникам (и вообще всем желающим, например, через газету или Internet; открытые ключи можно публиковать в специальных справочниках и т. п.).

Обозначим через D множество всех возможных сообщений (например, это может быть множество всех битовых строк). Потребуем, чтобы каждый ключ задавал перестановку множества D, и через $P_A()$ и $S_A()$ будем обозначать перестановки, соответствующие ключам Алисы. Мы считаем, что каждая из перестановок $P_A()$ и $S_A()$ может быть быстро вычислена, если только известен соответствующий ключ.

Мы хотим, чтобы ключи одного участника задавали взаимно обратные перестановки, т. е. чтобы

$$M = S_A(P_A(M)) \qquad (33.37)$$

и

$$M = P_A(S_A(M)) \qquad (33.38)$$

было выполнено для любого сообщения $\mathrm{M} \in D$.

Самое главное – чтобы никто, кроме Алисы, не мог вычислять функцию за разумное время; именно на этом основаны все полезные свойства криптосистемы, перечисленные выше. Поэтому – то Алиса и держит значение $S_A()$ в секрете: если кто-либо узнает ее секретный ключ, он сможет расшифровать адресованные ей сообщения, подделывать ее подпись или подменять сообщения, которые она отправляет от своего имени. Главная трудность при разработке криптосистемы состоит в том, чтобы придумывать функцию $S_A()$, для которой трудно было бы найти быстрый способ вычисления, даже зная такой способ для обратной функции $P_A()$.

Опишем процесс пересылки шифрованного сообщения. Допустим, Боб желает послать Алисе секретное сообщение. Это происходит так:

- Боб узнаёт $P_A()$ - открытый ключ Алисы (по справочнику или прямо от Алисы);
- Боб зашифровывает свое сообщение M и посылает Алисе **результат шифрования** $C = P_A(M)$;
- Алиса получает C и восстанавливает исходное сообщение $M = S_A(C)$.

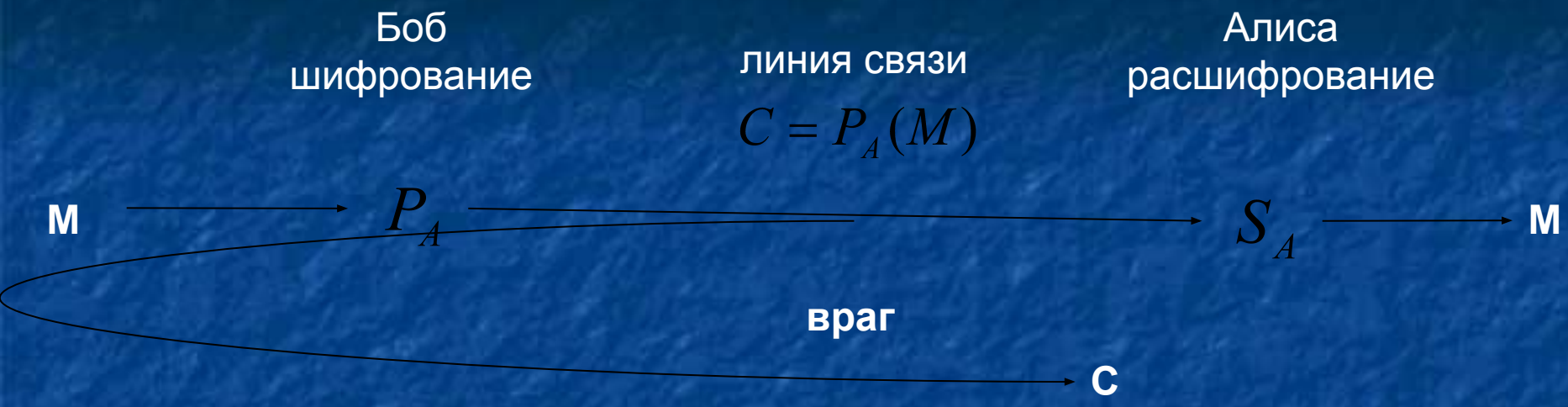


Рис. Шифрование с открытым ключом. Боб шифрует сообщение M с помощью функции P_A и получает результат шифрования $C = P_A(M)$. Функции $S_A()$ и $P_A()$ взаимно обратны, поэтому Алиса может восстановить исходное сообщение: $M = S_A(C)$. Никто, кроме Алисы, не знает способа вычисления $S_A()$, поэтому сообщение M останется секретным, даже если враг перехватит C и знает $P_A()$.

Теперь объясним, как снабдить сообщение цифровой подписью. Пусть Алиса хочет послать Бобу ответ , подписанный цифровой подписью Тогда:

- Алиса вычисляет **цифровую подпись** (digital signature) $\sigma = S_A(M')$;
- Алиса посылает Бобу пару (M' , σ), состоящую из сообщения и подписи;
- Боб получает пару (M', σ) и убеждается в подлинности подписи, проверив равенство $M' = P_A(\sigma)$.

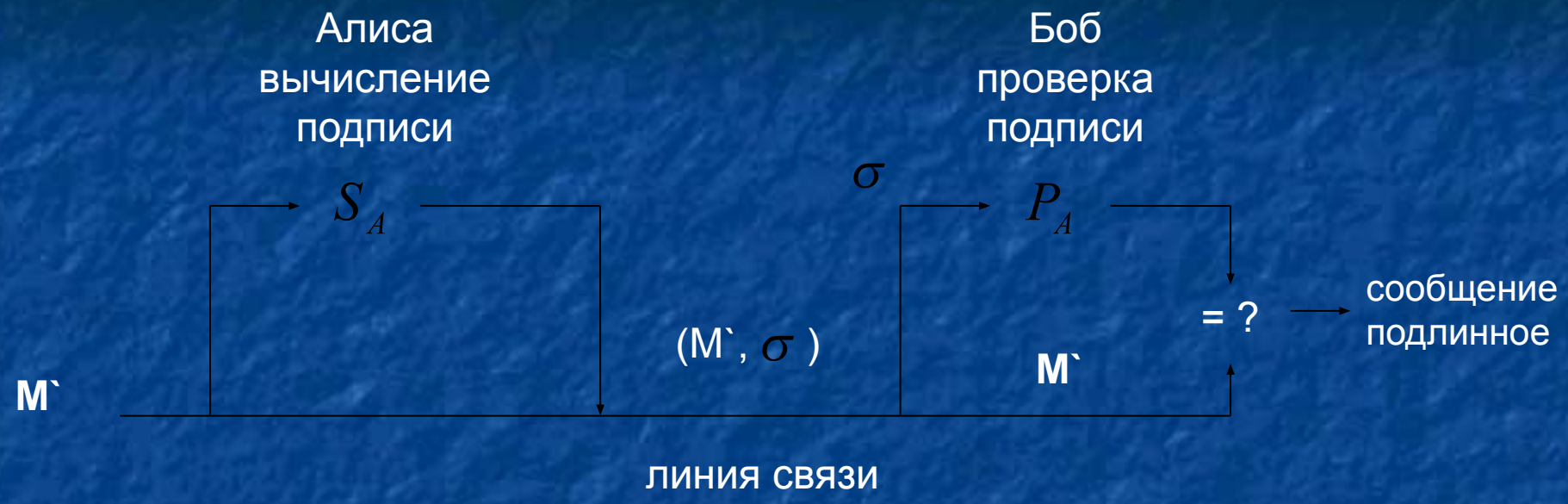


Рис. Цифровая подпись в системе с открытым ключом. Алиса подписывает свое сообщение M`, прикладывая к нему цифровую подпись $\sigma = S_A(M`)$. Боб , получая от Алисы пару (M`, $\sigma$), проверяет соотношение $M` = P_A(\sigma)$. Если оно выполняется, подпись и само сообщение подлинны.

16) Криптосистема RSA

Чтобы построить пару ключей для криптосистемы RSA надо сделать следующее.

1. Взять два больших простых числа *p* и *q* (скажем, около 100 десятичных цифр в каждом).
2. Вычислить *n=pq*.
3. Взять небольшое нечетное число *e*, взаимно простое с $\varphi(n)$ (Из определения функции Эйлера следует, что $\varphi(n) = (p-1)(q-1)$).)
4. Вычислить $d = e^{-1} \bmod \varphi(n)$. (Если НОД ($e, \varphi(n)$))=1, то $ed \equiv 1 (\bmod \varphi(n))$.)
5. Составить пару *P=(e, n)* –**открытый ключ** (RSA public key).
6. Составить пару *S=(d, n)* – **секретный ключ** (RSA secret key).

Открытому ключу *P=(e, n)* соответствует преобразование

$$P(M) = M^e \bmod n$$

а секретному ключу *S=(d, n)* – преобразование

$$S(C) = C^d \bmod n$$

Как уже говорилось, эти преобразования можно использовать и для шифрования, и для электронных подписей.

Для возведения в степень в предыдущих формулах разумно пользоваться процедурой быстрого возведения в степень. Если считать, что числа d и n имеют порядка β битов, а число e имеет *O(1)* битов, то преобразование *P* потребует *O(1)* умножений по модулю n ($O(\beta^2)$ битовых операций), а преобразования *S* потребует $O(\beta)$ умножений ($O(\beta^3)$ битовых операций (разумеется при известном ключе).

Теорема 14 (корректность системы RSA).

Формулы

$$P(M) = M^e \bmod n \quad \text{и} \quad S(C) = C^d \bmod n$$

задают взаимно обратные перестановки множества.

Доказательство.

$$P(S(M)) = S(P(M)) = M^{ed} \bmod n$$

$$P(S(M)) = P(M^d) \qquad S(P(M)) = S(M^e)$$

$$P(M^d) = (M^d)^e \qquad S(M^e) = (M^e)^d$$

$$(M^d)^e = (M^e)^d$$

$$ed \equiv 1 (\bmod \varphi(n)) \Rightarrow ed = k\varphi(n) + 1$$

$\Rightarrow$ по усиленной теореме Эйлера

$$M^{ed} = M^{k\varphi(n)+1} \equiv M (\bmod n)$$

ч.т.д.

17) Криптостойкость схемы RSA.

Открытый ключ представляет собой пару чисел (e, n), секретный – пару (d, n), где d – обратный элемент в кольце вычетов по модулю . Следовательно для вычисления нужно знать функцию Эйлера $\phi(n)$. Задача вычисления функции Эйлера эквивалентна задаче о разложении числа n на множители. Задача разложения числа на множители очень сложна. Она привлекла внимание многих математиков, начиная с Эйлера, который разложил на множители пятое число Ферма

$$F5 = 2^{32} + 1 = 4294967297$$

до этого ошибочно считавшееся простым. Эйлер использовал изящную идею – найти два квадрата, совпадающих по модулю n, которая и по сей день лежит в основе многих современных алгоритмов разложения.

18) Электронные подписи.

Наиболее популярными функциями хэширования являются MD5 (Message Digest 5 – профиль сообщения 5), создающий 16-байтовый результат, и алгоритм SHA (Secure Hash Algorithm – надежный алгоритм хэширования), формирующий 20-байтовый результат.

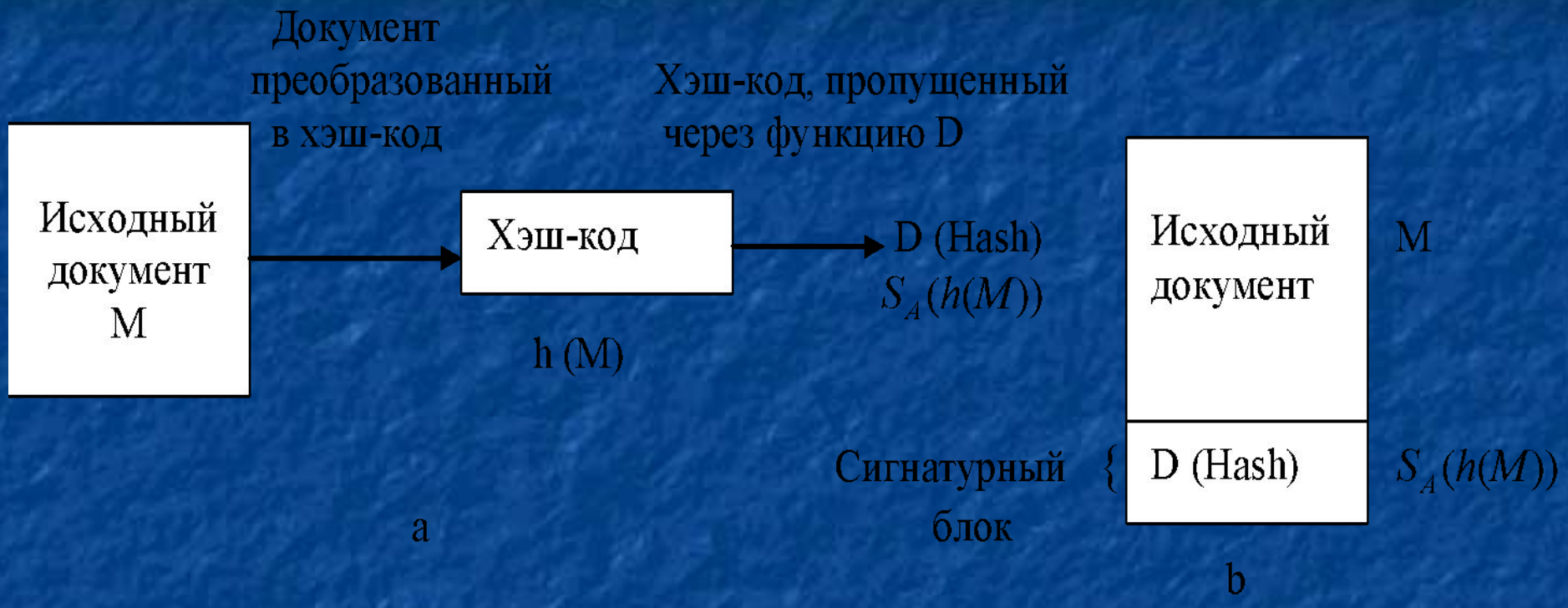


Рис. Вычисление сигнатурного блока (а); что получает получатель (б)

Когда **документ и хэш-код прибывают**, получатель сначала с помощью алгоритма MD5 или SHA (о выборе алгоритма отправитель и получатель договариваются заранее) **вычисляет хэш-код документа h(M).** Затем получатель применяет с сигнатурному блоку алгоритм шифрования с открытым ключом, получая $P_A(S_A(h(M))) = h(M)$. В результате он снова **зашифровывает "расшифрованный" хэш-код,** снова получая **оригинальное значение хэш-кода**. Если вычисленный заново хэш-код не совпадает с расшифрованным сигнатурным блоком, это значит, что либо сообщение, либо сигнатурный блок были **повреждены** – или случайно, или преднамеренно. Смысл этой схемы в том, что **медленное шифрование с открытым ключом применяется только для небольшого по размерам хэш-кода.**

19) Атаки на RSA

Известно: открытый ключ *(e, n)* ; зашифрованный текст Найти: *M* – исходное (незашифрованное) сообщение. Решение:

1. Возводим в степень (…(P^e) ^{e} ….) ^{e}= $P^{(e^j)}$ j раз, пока не получим P, запоминая предыдущий результат возведения в степень. целостность данных;

- $P^{e^j} = P$
  $(P^{e\ldots e})^e = M^e$
  $A = P^{e^{j-1}}$
  $A^e = M^e$
  $A^{ed} = M^{ed}$

  Используя усиленную теорему Эйлера, получим:

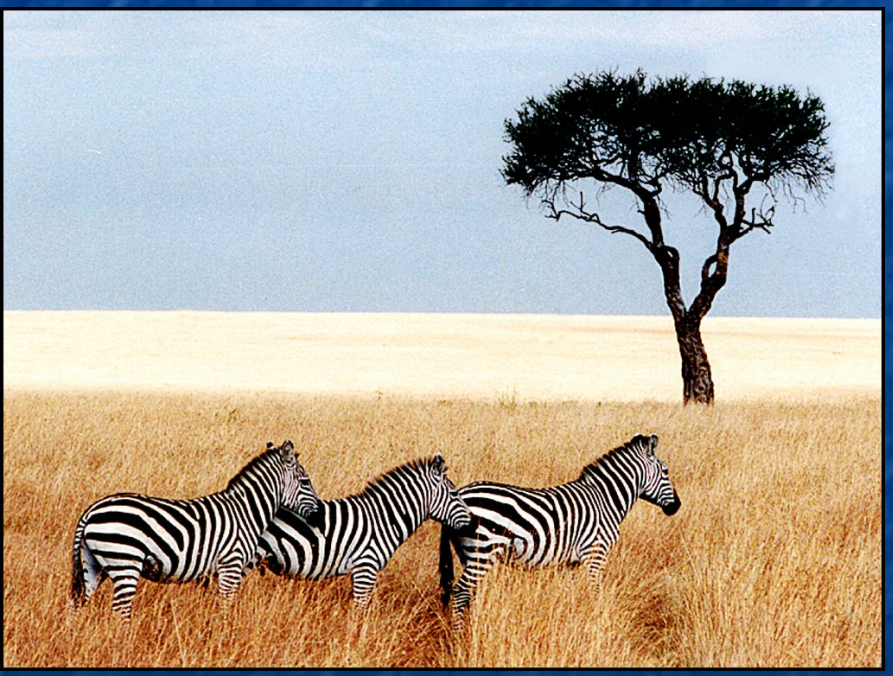
  $$A^{ed} = A,\ M^{ed} = M \implies A = M.$$

Таким образом, (j-1) раз возведенное в степень зашифрованное сообщение P и есть исходное незашифрованное сообщение M. Но оно было запомнено на предыдущем шаге.

3. Атака не эффективна, так как число операций может оказаться сравнимым или даже большим, чем при разложении чисел на простые сомножители.

20) СТЕГАНОГРАФИЯ

а)



б)