Криптографические средства защиты объектов информатизации

Лекция 8
Ассиметричные системы шифрования,
электронная цифровая подпись

Требования к асимметричным системам (У. Диффи, М. Хеллман)

- Вычисление пары ключей (открытого и закрытого) должно быть простым
- Отправитель, зная открытый ключ Ко легко вычисляет криптограмму $C=E_{ko}(M)$
- Получатель, используя закрытый ключ Кс и криптограмму С легко восстанавливает исходное сообщение $M=D_{кс}(C)$
- Противник, зная открытый ключ при попытке вычислить секретный наталкивается на непреодолимую проблему
- Противник, зная открытый ключ и криптограмму, при попытке восстановить исходное сообщение наталкивается на непреодолимую вычислительную проблему

Схема функционирования асимметричной криптосистемы Шифрование с открытым ключом

Однонаправленные функции

- Опр: Однонаправленной называется функция F: X□Y, обладающая свойствами
- 1) существует полиномиальный алгоритм вычисления F(x)
- 2) не существует полиномиального алгоритма инвертирования F (т.е. решения уравнения F(x)=y относительно x)

Примеры однонаправленных (односторонних функций)

- Разложение большого числа на простые множители N=P*Q,

где P и Q – простые.

- Задача дискретного логарифмирования

$$y = ind_g a$$

Асимметричные системы шифрования

- RSA (Ronald Linn Rivest, Adi Shamir, Leonard Adleman)
- El-Gamal(Шифросистема Эль-Гамаля)
- **DSA** (Digital Signature Algorithm)
- **Diffie-Hellman** (Обмен ключами Диффи — Хелмана)
- **ECC** (Elliptic Curve Cryptography, Системы шифрования основанные на эллиптических кривых)
- **ГОСТ Р 34.10-2001**
- **Rabin**
- **Luc**
- **McEliece**
- **Williams System** (Криптосистема Уильямса)

Криптосистема RSA (Rivest, Shamir, Adleman)

- Открытый текст шифруется блоками, длиной $2^k : 2^k<n<2^{k+1}$.
- Процедура шифрования состоит из 2-х этапов: определение ключей и шифрование/расшифрование

Алгоритм RSA. Определение ключей

Отправитель	Получатель
	1. Формирует 2 простых числа P и Q. 2. Вычисляет N=P*Q 3. Вычисляет функцию Эйлера φ(N)=(P-1)(Q-1) 4. Выбирает случайное Ко: $1<Ko<\varphi(N),\ (Ko, \varphi(N))=1$
	Ко, N – открытые ключи
Ко, N	
	5. Кс находят как решение линейного сравнения $Ko*Kc=1 \bmod \varphi(N)$

Алгоритм RSA. Шифрование/Расшифрование

Отправитель	**Получатель**
M- сообщение: 1<M<N **$C=M^{Ko} \bmod N$**	
шифротекст C	
	$M=C^{Kc} \bmod N$

Пример (Получатель)

- 1. P=5 Q=11
- N=5*11=55
- φ(N)=(P-1)(Q-1)=4*10=40
- Ko: 1<Ko<φ(N), (Ko, φ(N))=1. Ko=7
- Открытые ключи N=40, Ko=7 отправителю
- Секретный ключ: Ko*Kc=1 mod φ(N)
- 7Kc=1 mod 40
- φ(40)= φ(2^3*5)=(2^3-2^2)(5^1-5^0)=4*4=16
- Kc=7^{16-1}*1 mod 40 = 7^{15} mod 40 =
- 49*49*49*49*49*49*49*7 mod 40 =
- 9*9*9*9*9*9*9*7 mod 40 = 1*1*1*63 mod 40=23

Kc=23

Пример (Шифрование)

- Отправитель M=6
- $C=M^{Ko} \bmod N$
- $C=6^7 \bmod 55 = 6^3 6^3 6 \bmod 55=$

$216*216*6 \bmod 55 = 51*51*6 \bmod 55 =$

$51*306 \bmod 55= \ 51*31 \bmod 55 = 41$

Пример (Расшифрование)

Получатель C=41

$M=C^{Kc} \bmod N = 41^{23}$

$\bmod 55 = 41^{2}*41^{21}$

$\bmod 55 =$

$31*41^{21} \bmod 55 =$

$31^{11}*41 \bmod 55 =$

$26^{5}*31*41 \bmod 55 =$

$16^{2}*26*6 \bmod 55 =$

$36*46 \bmod 55 = 6$

Технологии, построенные на криптографии с открытым ключом

- Распределенная проверка подлинности (аутентификация)
- Коды аутентификации сообщений (Message authentication codes или MAC)
- Согласование общего секретного ключа сессии
- Шифрование больших объемов данных без предварительного обмена общим секретным ключом
- Электронная цифровая подпись

Протоколы идентификации и аутентификации

- При обмене информацией необходимо выполнять требования защиты:
- Получатель должен быть уверен в подлинности:

  - источника данных;
  - данных;
- Отправитель должен быть уверен:

  - доставке данных получателю;
  - подлинности доставленных данных.

Идентификация и аутентификация

- Идентификация – функция системы, которая выполняется когда объект пытается войти в систему
- А – доказывающий – проходит идентификацию
- В – проверяющий – проверяет личность доказывающего

Идентификация на основе пароля (схема 1)

- ID_i – идентифицирующая информация i-го пользователя (напр. ЛОГИН)
- K_i – аутентифицирующая информация i-го пользователя (напр. ПАРОЛЬ)
- F – однонаправленная функция

Номер пользователя	Информация для идентификации	Информация для аутентификации
1	ID_1	$E_1=F(ID_1,K_1)$
2	ID_2	$E_2=F(ID_2,K_2)$
3	ID_3	$E_3=F(ID_3,K_3)$
…	…	…

Идентификация на основе пароля (схема 2)

- ID_i – идентифицирующая информация i-го пользователя (ЛОГИН)
- K_i – аутентифицирующая информация i-го пользователя
- F – однонаправленная функция
- S_i – случайная последовательность – «соль»

Номер пользователя	Информация для идентификации	Информация для аутентификации
1	ID_1, S_1	$E_1=F(K_1,S_1)$
2	ID_2, S_2	$E_2=F(K_2,S_2)$
3	ID_3, S_3	$E_3=F(K_3,S_3)$
…	…	…

Правила составления паролей

- Ограничение на минимальную длину (не менее 8 символов)
- Наличие различных групп символов (верхний и нижний регистры, цифры, специальные символы)
- Не должен быть словом

Недостатки протоколов с паролем:

А передает секретную информацию (пароль) В

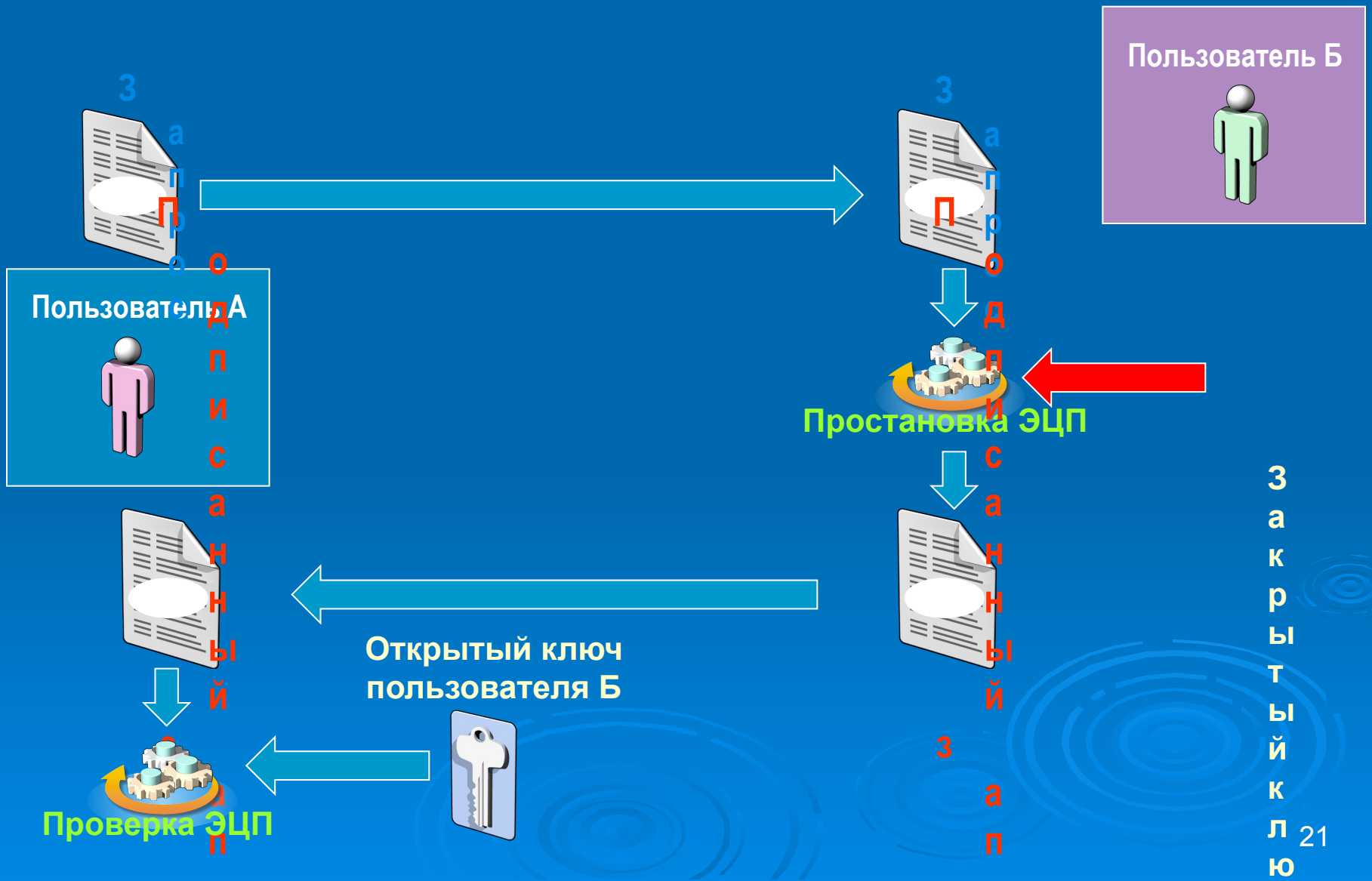
Протокол идентификации с нулевой передачей знаний

Область применения:
- электронные деньги
- системы электронного голосования
-электронные системы оплаты

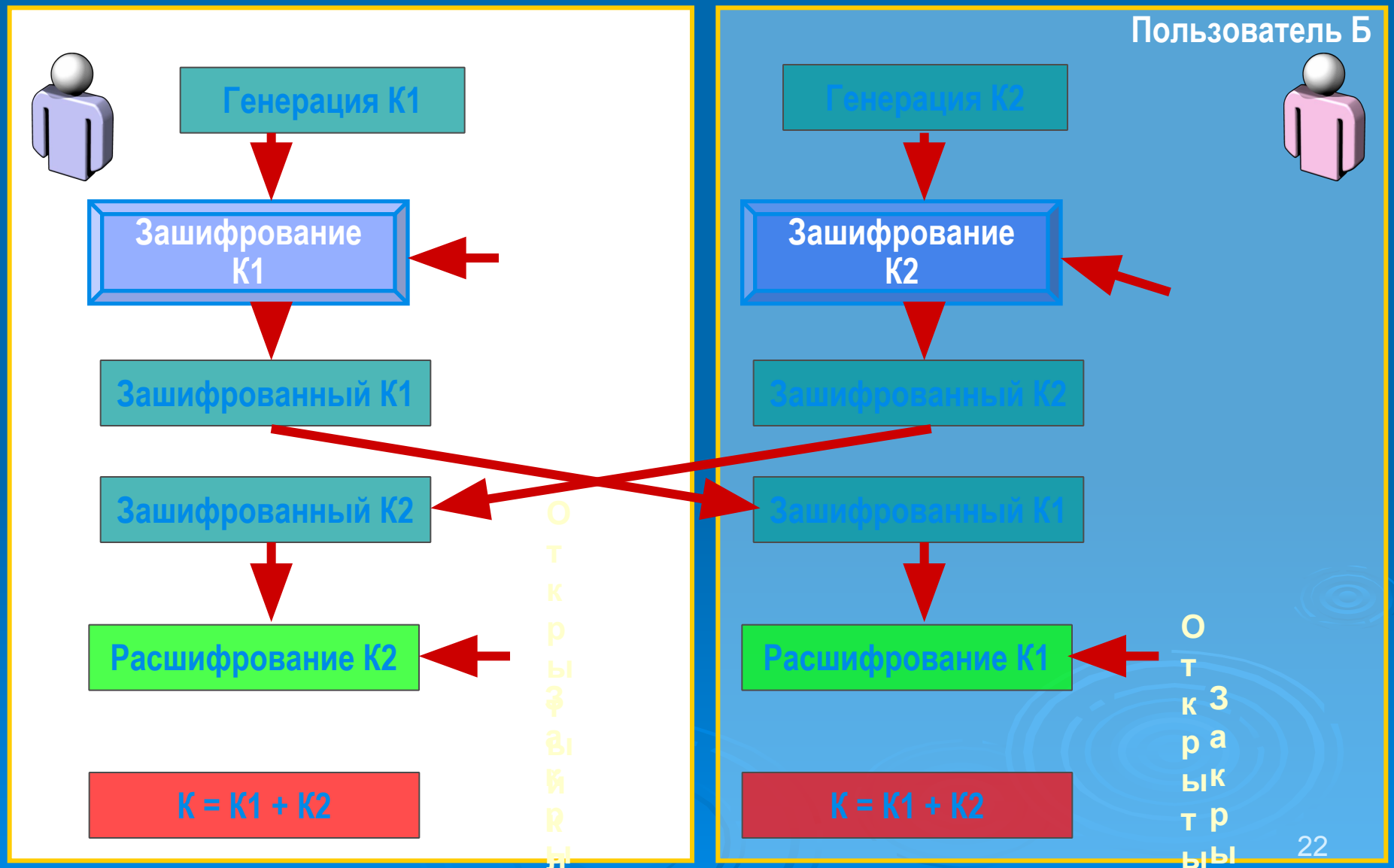
Протокол Фиата-Шимира

SMART - карта	Устройство чтения
Знает s, $ID=s^2$, $N=P*Q$	Знает n, ID
	ID
Генерирует случайное число r	
Вычисляет $t = r^2 \bmod N$	
	t
	Выбирает случайное e = 0, или e=1
	e
Вычисляет $u=(r*s^e) \bmod n$	
	u
	Проверяет $u^2 =(t*ID^e) \bmod N$

Аутентификация (доказательство владения закрытым ключом)



Согласование общего секретного ключа сессии



Протокол передачи ключей. Прямой обмен ключами Диффи-Хеллмана

A	B
P, g < P - открытые ключи	
Генерирует случайное число a	Генерирует случайное число b
Вычисляет $S1=g^a \bmod P$	Вычисляет $S2=g^b \bmod P$
S1 □ □ S2	
Вычисляет $K=S2^a \bmod P = g^{ab} \bmod P$	Вычисляет $K=S1^b \bmod P = g^{ab} \bmod P$

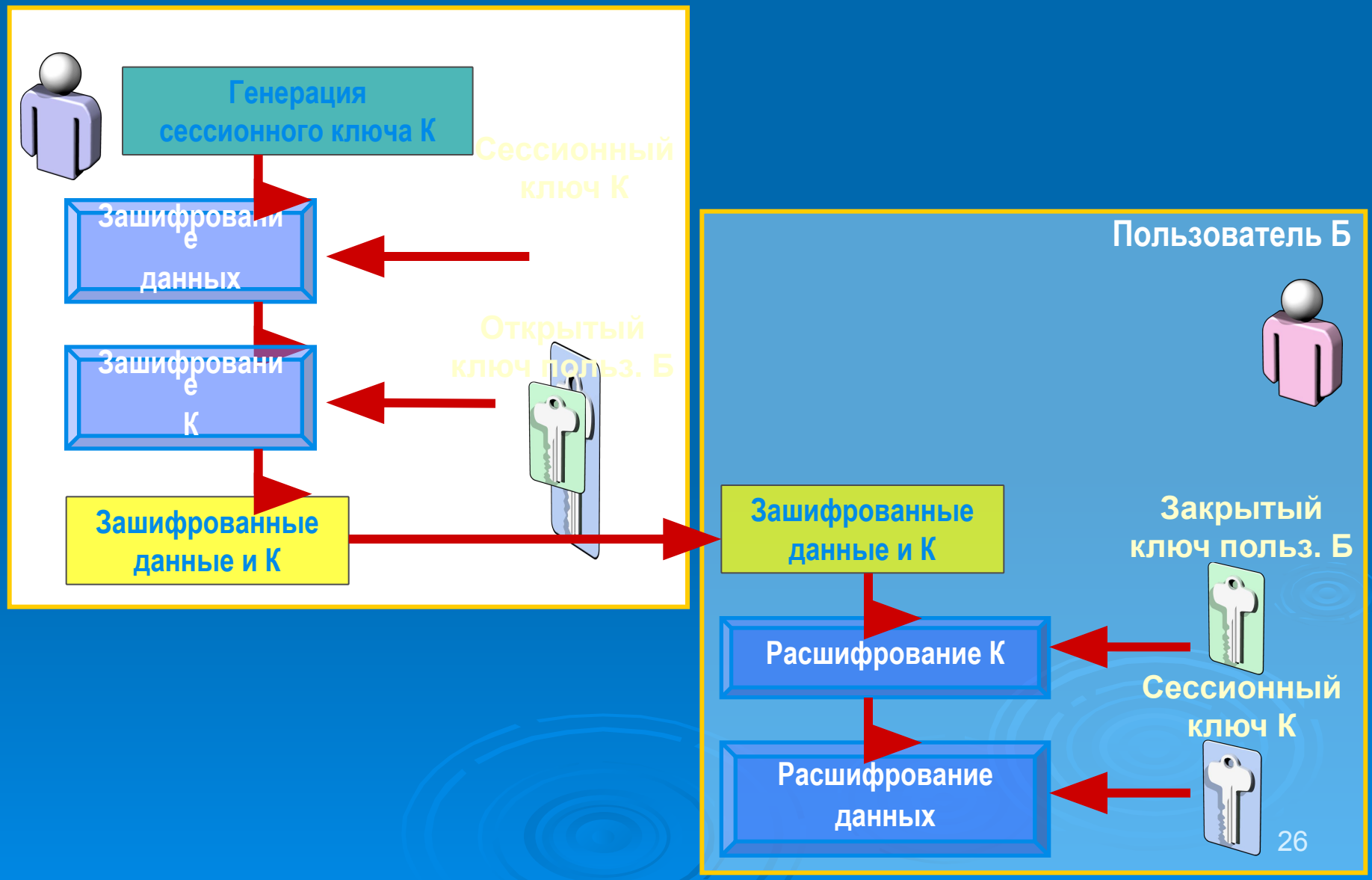
Пример

A	B
P=17 g=5	
Генерирует случайное число a = 3	Генерирует случайное число b =7
Вычисляет $S1=5^3 \bmod 17=6$	Вычисляет $S2=5^7 \bmod 17=10$
S1=6 ⇨ ⇦ S2=10	
Вычисляет $K=10^3 \bmod 17$ =14	Вычисляет $K=6^7 \bmod 17$ =14

Атака третьей стороны

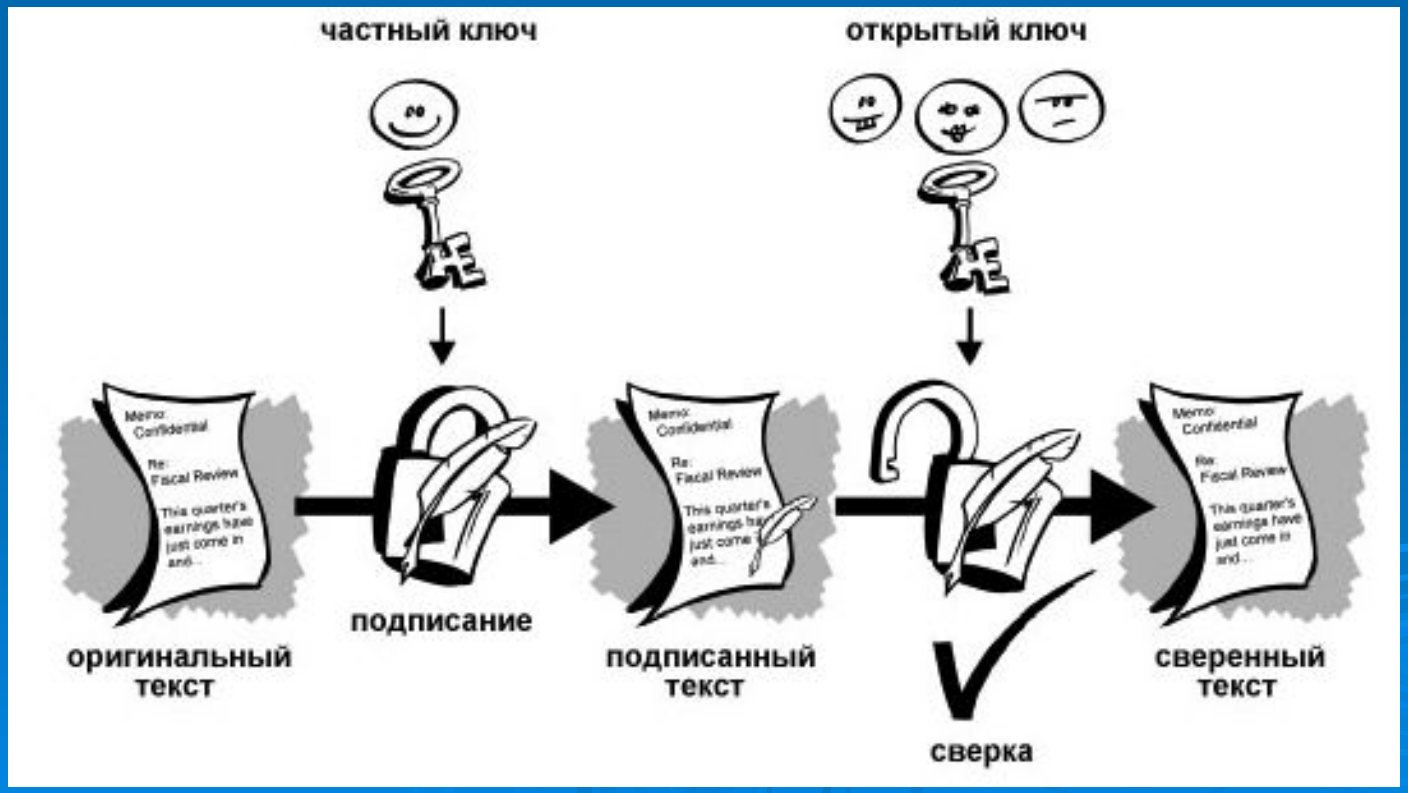
A	?	B
P, g		
Генерирует случайное число a	Генерирует: a1 (для b) b1 (для a)	Генерирует случайное число b
Вычисляет $S1=g^{a} \bmod P$	$C1=g^{a1} \bmod P$ $C2=g^{b1} \bmod P$	Вычисляет $S2=g^{b} \bmod P$
S1 □	C1 □ □ C2	□ S2
Вычисляет $K1=C2^{a} \bmod P$ $=g^{a*b1} \bmod P$	Знает K1, K2 – может незаметно для A и B расшифровывать/подменять информацию	Вычисляет $K2=C1^{b} \bmod P$ $=g^{a1*b} \bmod P$

Шифрование без предварительного обмена симметричным секретным ключом



Электронная цифровая подпись

Время, когда мы могли писать только пером и на бумаге давно кануло в лету... С появлением персональных компьютеров неизбежно должно было появится **нечто**, что бы дало возможность персонализировать документы, однозначно подтверждая авторство.



Электронная цифровая подпись

ЭЦП представляет собой конечную цифровую последовательность, зависящую от самого сообщения или документа и от некоторого секретного ключа, известного только подписывающему субъекту.

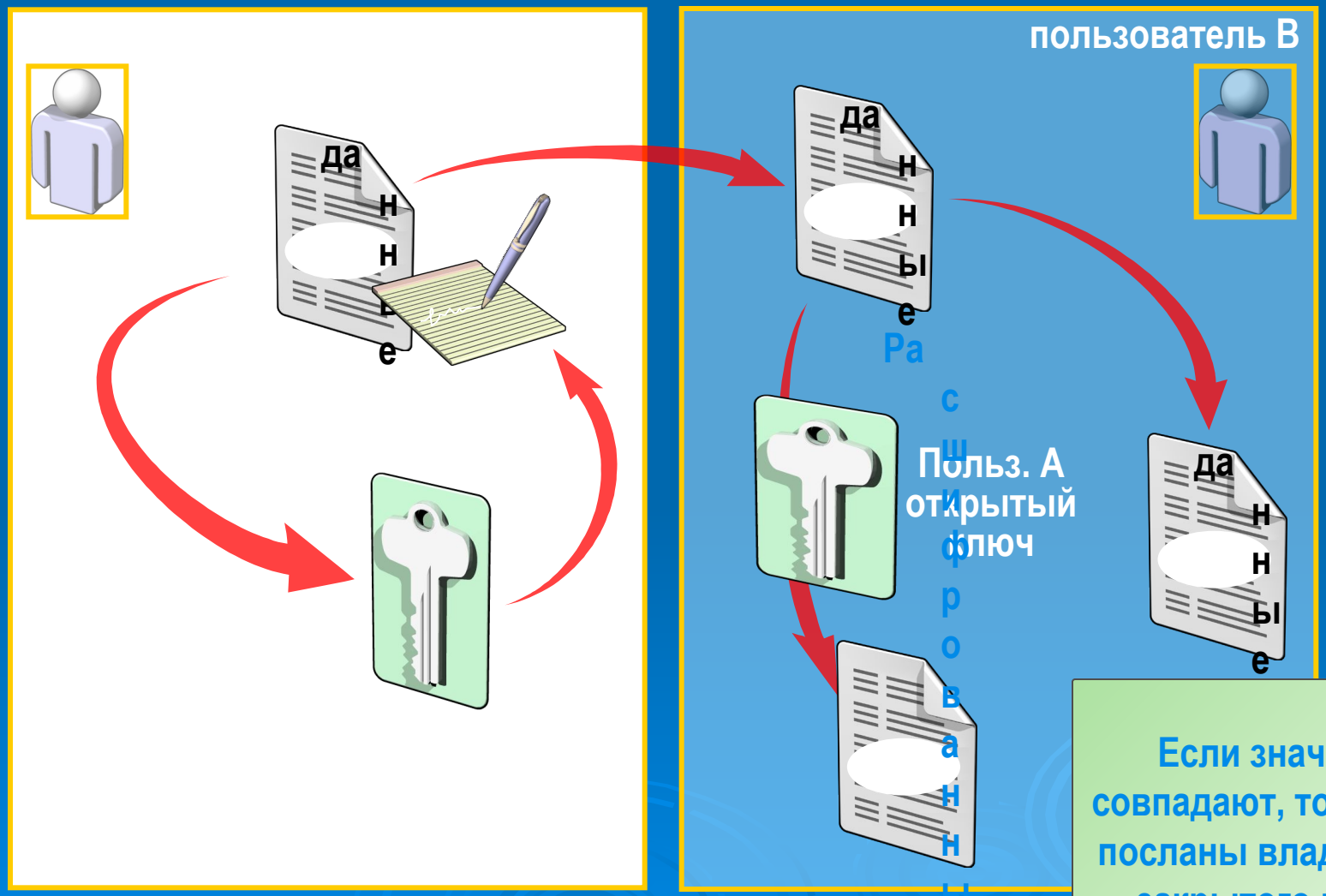
ЭЦП позволяет решить следующие три задачи:

- осуществить *аутентификацию источника данных*
- установить *целостность* сообщения или электронного документа
- обеспечить невозможность отказа от факта подписи

Алгоритмы ЭЦП

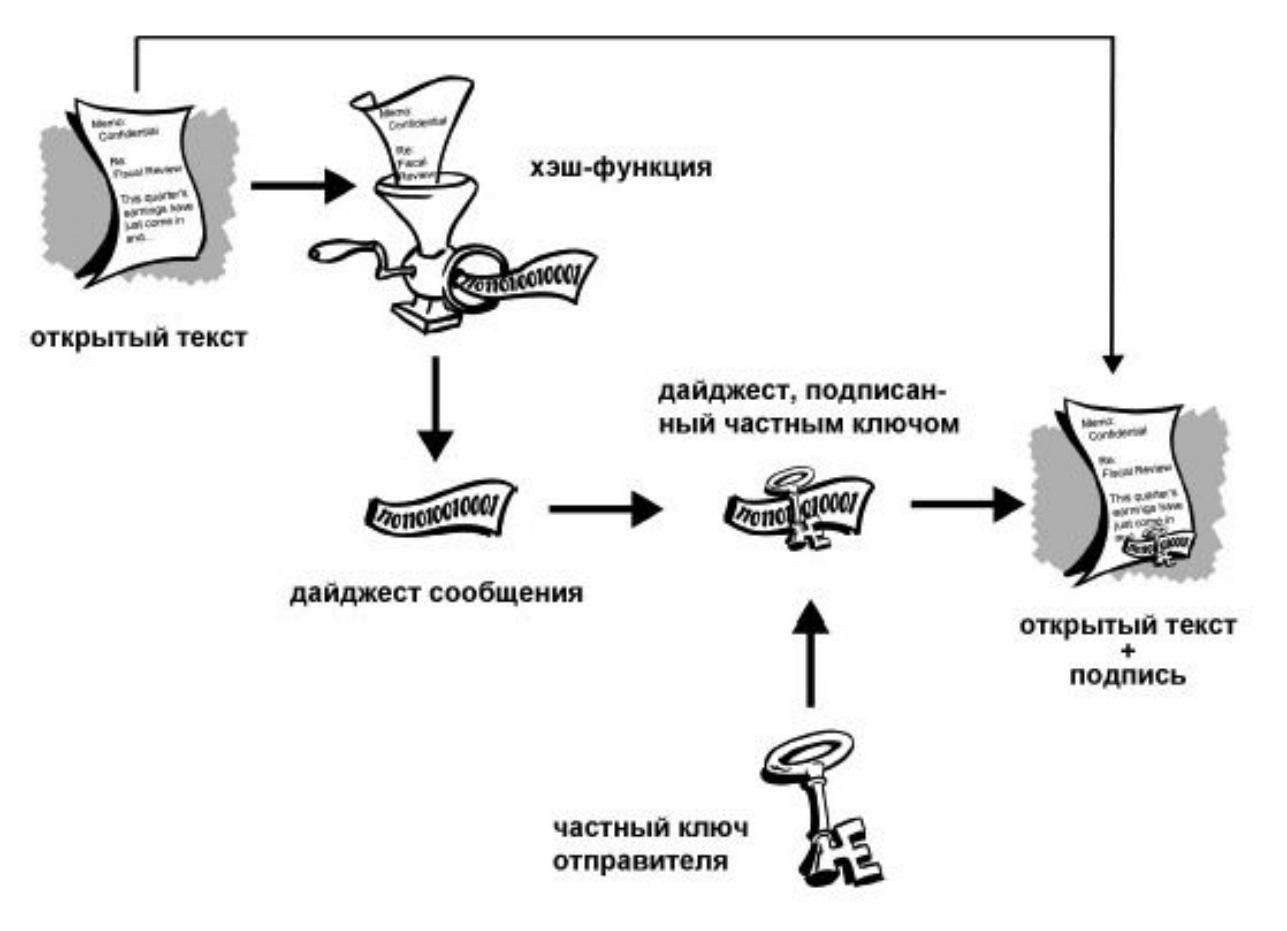
- RSA (Ronald Linn Rivest, Adi Shamir, Leonard Adleman)
- DSA (Digital Signature Algorithm)
- EC – DSA (вариант DSA основанный на эллиптических кривых)
- Алгоритм подписи Шнорра
- Алгоритм подписи Ниберга-Руппеля
- и др.

Принцип работы ЭЦП для коротких сообщений



Использование Хэш-функций

Хэш-функция – функция, которая осуществляет сжатие строки произвольного размера в строку чисел фиксированного размера



Хеш-функция

Хеширование – математическое однонаправленное преобразование текста в число фиксированной размерности

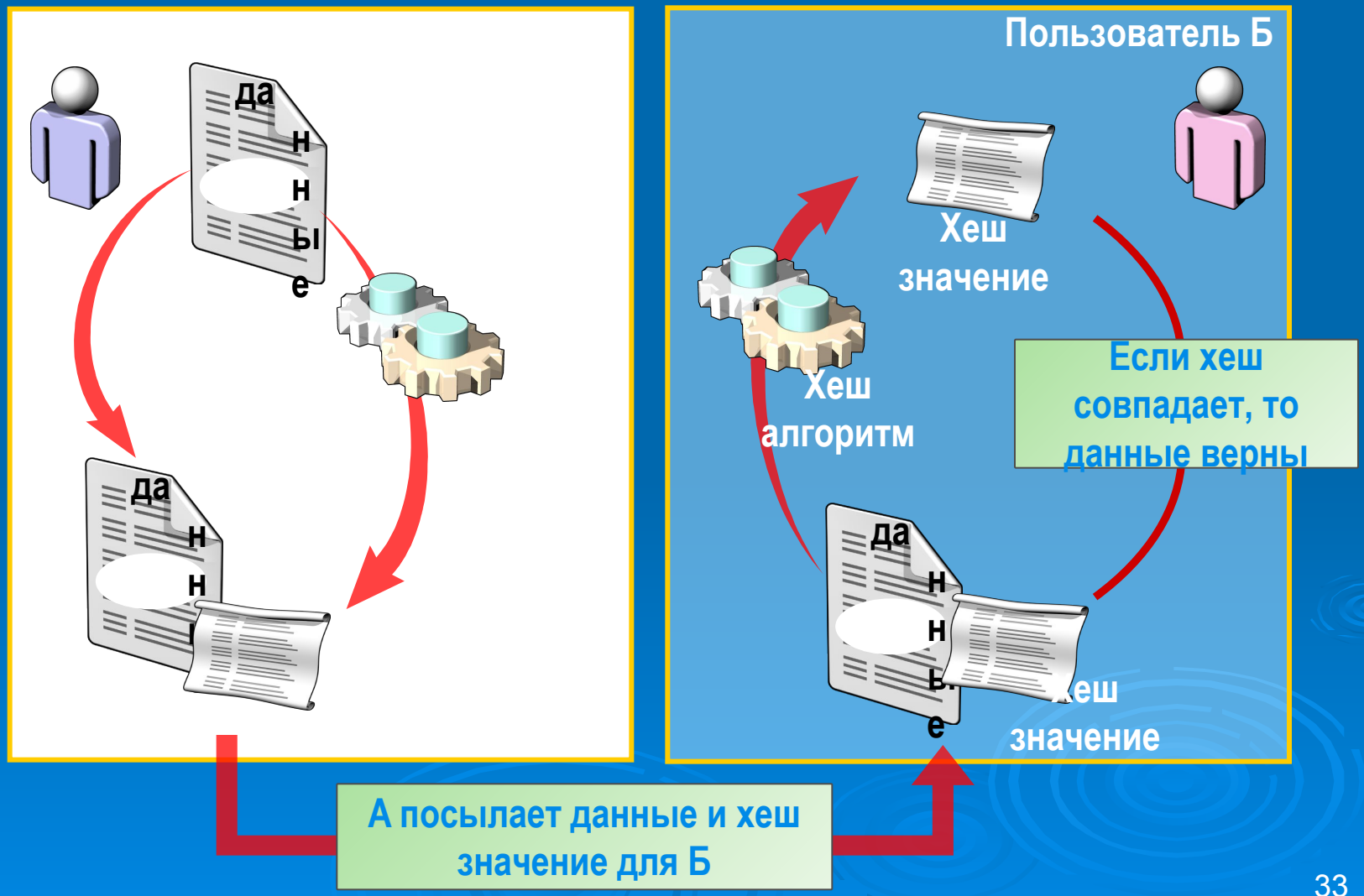
Свойства хеш-функции

- Свойство лавинности
- Свойство стойкости к коллизиям
- Свойство необратимости

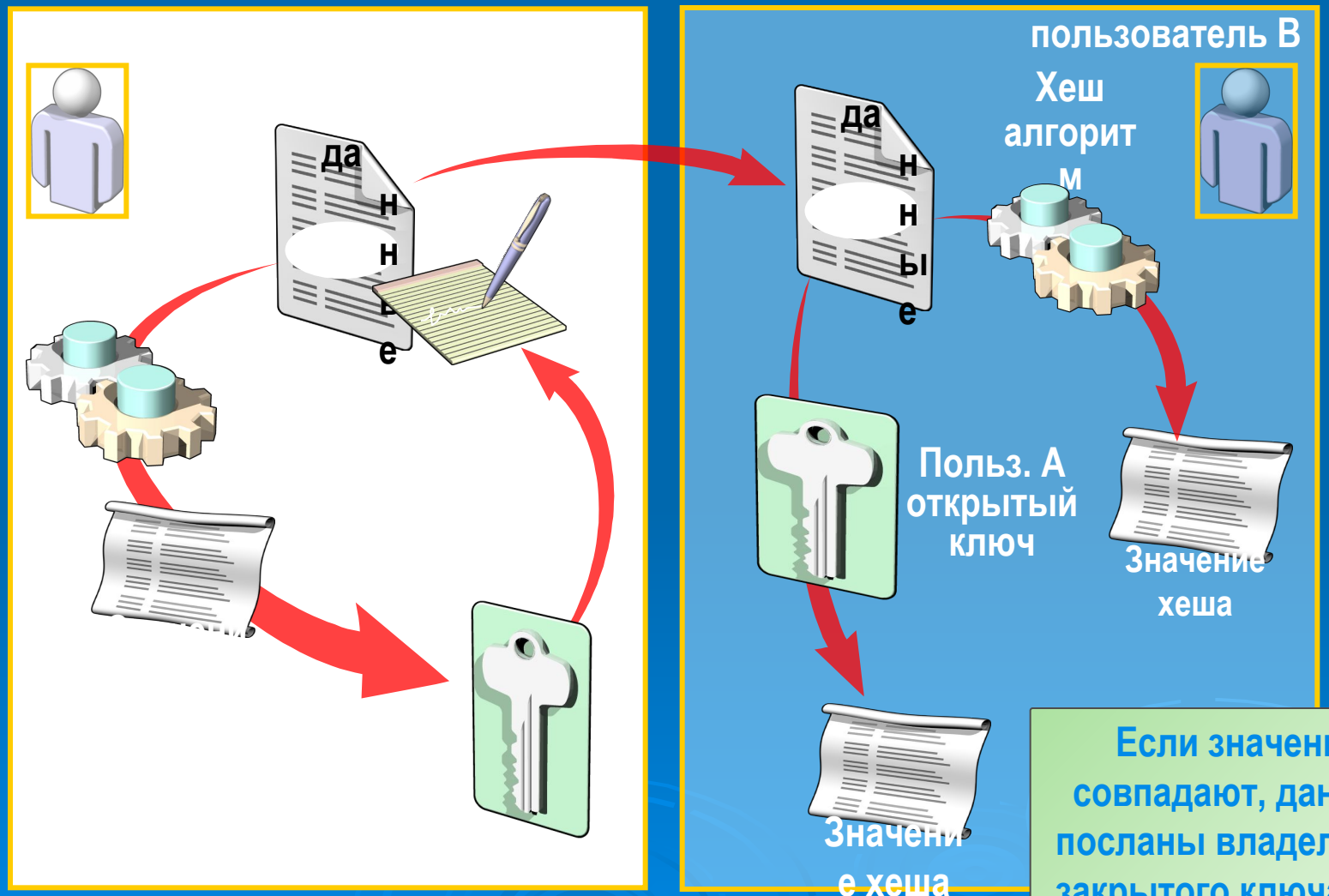
Алгоритмы, реализующие хеш-функции

- MD4, MD5 (128 бит)
- SHA и его варианты SHA1 (160 бит), SHA-256, SHA-512, SHA-384
- RIPEMD-160
- российский стандарт ГОСТ Р 34.11-94

Проверка целостности данных с использованием хеша



Принцип работы ЭЦП с хешированием сообщений



Электронная цифровая подпись (на основе RSA)

А сформировал свою пару секретного (Кс) и открытого(Ко) ключей и на их основе построил функции Dkc(m) и Eko(m): Eko(Dkc (m))=m

- Функция Dkc (m) - функция подписи сообщения m
- Функция Eko (m) - функция проверки подписи для сообщения m.

Схема взаимодействия отправителя (А) и получателя (В)

Отправитель	**Получатель**
M- исходный текст	
m=H(M) **вычисляет хэш-функцию**	
1<m<N **$C=m^{Kc} \bmod N$ - подпись**	
M,C	
	$m=C^{Ko} \bmod N$ **Проверяет m = H(M)**

Атаки на цифровую подпись

- Атаки на алгоритмы (возможные ошибки в алгоритмах):
  - Повторение одних и тех же значений алгоритмами генерации случайных чисел
  - Возникновение “коллизий” для хеш-функции
  - Хранение алгоритма в секрете
- Атаки на криптосистему
- Атаки на реализацию:
  - Секретный ключ ЭЦП хранится на жестком диске
  - После завершения работы системы ЭЦП, ключ, хранящийся в оперативной памяти, не затирается
  - Обеспечивается безопасность сеансовых ключей и недостаточное внимание уделяется защите главных ключей
  - Отсутствует контроль целостности программы генерации или проверки ЭЦП, что позволяет злоумышленнику подделать подпись или результаты ее проверки
- Атаки на пользователя

Нет в мире совершенства...

- К сожалению, использование ЭЦП связано со своими, весьма серьезными проблемами. Наиболее острая аналогична "основному вопросу" асимметричного шифрования: как убедиться, что открытый ключ для проверки ЭЦП действительно принадлежит лицу, поставившему подпись, а не подменен злоумышленником по дороге? Ведь успешная подмена открытого ключа ложным позволит злоумышленнику легко подделать вожделенную подпись. Конечно, есть и "противоядия" - методы борьбы с подменой открытых ключей, например, их сертификация...