Информационная безопасность

Лебедева Т.Ф.

Управление ключами

Кроме выбора подходящей для конкретной ИС криптографической системы, важная проблема - управление ключами. Как бы ни была сложна и надежна сама криптосистема, она основана на использовании ключей. Если для обеспечения конфиденциального обмена информацией между двумя пользователями процесс обмена ключами тривиален, то в ИС, где количество пользователей составляет десятки и сотни, управление ключами - серьезная проблема.

Под *ключевой информацией* понимается совокупность всех действующих в ИС ключей. Если не обеспечено достаточно надежное управление ключевой информацией, то завладев ею, злоумышленник получает неограниченный доступ ко всей информации.

Управление ключами - информационный процесс, включающий в себя три элемента:

- генерацию ключей;
- накопление ключей;
- распределение ключей.

- **Генерация ключей**
- **Известно, что не стоит использовать неслучайные ключи. В серьезных ИС используют специальные программные и аппаратные методы генерации случайных ключей.**
- **Как правило, используется датчики случайных и псевдослучайных чисел. Степень случайности должна быть высокой. Идеальными генераторами являются устройства на основе натуральных случайных процессов. Физический датчик случайных чисел встроен в ядро процессора Pentium 3.**
- **Математически случайные числа можно получить, используя десятичные знаки трансцендентных чисел, например, π или e, которые вычисляются с помощью стандартных математических методов.**
- **ИС со средними требованиями защищенности использует программное получение случайных чисел.**

Управление ключами

- Накопление ключей
- Под *накоплением ключей* понимается организация их хранения, учета и удаления.
- Поскольку ключ является самым привлекательным для злоумышленника объектом, открывающим ему путь к конфиденциальной информации, то вопросам накопления ключей следует уделять особое внимание.
- *Секретные ключи никогда не должны записываться в явном виде на носителе, который может быть считан или скопирован.*
- В достаточно сложной ИС один пользователь может работать с большим объемом ключевой информации, и иногда даже возникает необходимость организации мини-баз данных по ключевой информации. Такие базы данных отвечают за принятие, хранение, учет и удаление используемых ключей.

Управление ключами

- Итак, каждая информация об используемых ключах должна храниться в зашифрованном виде. Ключи, зашифровывающие ключевую информацию называются *мастер-ключами*. Желательно, чтобы мастер-ключи каждый пользователь знал наизусть, и не хранил их вообще на каких-либо материальных носителях.
- Очень важным условием безопасности информации является периодическое обновление ключевой информации в ИС. При этом переназначаться должны как обычные ключи, так и мастер-ключи. В особо ответственных ИС обновление ключевой информации желательно делать ежедневно.
- Вопрос обновления ключевой информации связан и с третьим элементом управления ключами - распределением ключей.

Распределение ключей

Распределение ключей - самый ответственный процесс в управлении ключами. К нему предъявляются два требования:

- Оперативность и точность распределения
- Скрытность распределяемых ключей.

В последнее время заметен сдвиг в сторону использования криптосистем с открытым ключом, в которых проблема распределения ключей имеет другое значение. Тем не менее распределение ключевой информации в ИС требует новых эффективных решений.

Распределение ключей между пользователями реализуются двумя разными подходами:

1) *Путем создания одного ли нескольких центров распределения ключей.* Недостаток такого подхода состоит в том, что в центре распределения известно, кому и какие ключи назначены и это позволяет читать все сообщения, циркулирующие в ИС.
2) *Прямой обмен ключами* между пользователями информационной системы. В этом случае проблема состоит в том, чтобы надежно удостоверить подлинность субъектов.

В обоих случаях должна быть гарантирована *подлинность сеанса связи*. Это можно обеспечить двумя способами:

- ***Механизм запроса-ответа*, который состоит в следующем. Если пользователь *A* желает быть уверенным, что сообщения, которые он получает от *B*, не являются ложными, он включает в посылаемое для *B* сообщение непредсказуемый элемент (запрос). При ответе пользователь *B* должен выполнить некоторую операцию над этим элементом (например, добавить 1). Это невозможно осуществить заранее, так как не известно, какое случайное число придет в запросе. После получения ответа с результатами действий пользователь *A* может быть уверен, что сеанс является подлинным. Недостатком этого метода является возможность установления хотя и сложной закономерности между запросом и ответом.**
- ***Механизм отметки времени ("временной штемпель").* Он подразумевает фиксацию времени для каждого сообщения. В этом случае каждый пользователь ИС может знать, насколько "старым" является пришедшее сообщение.**

Управление ключами

Распределение ключей в симметричных криптосистемах **может проходить следующими способами:**

1. **можно через курьера доставить ключ, но так как ключи должны обновляться, то доставлять дорого и неэффективно**
2. **получение двумя пользователя общего ключа от центрального органа (Центр Распределении Ключей – ЦРК). Передаваемый ключ шифруется ключом ЦРК. Недостаток: в ЦРК может появиться злоумышленник. Можно в виде дерева организовать хранение ключей в ЦРК.**
3. **Третий способ предложили ученые Диффи и Хеллман – протокол обмена ключами по открытому каналу. Протокол – это последовательность шагов, которые принимают 2 или большее число сторон для совместного решения задачи. Все шаги следуют в порядке очередности.**

Управление ключами

Протокол обмена ключами по открытому каналу Диффи-Хеллмана.

Цель: двум пользователям A и B получить общий секретный ключ.

1) *Пользователь A генерирует случайное число X.*
2) *Пользователь B генерирует случайное число Y.*
3) *A вычисляет: La=(a^x) mod m*
4) *B вычисляет: Lb=(a^y)mod m*
5) *A посылает пользователю B La.*
6) *B посылает пользователю A Lb.*
7) *A вычисляет Kab =((Lb)^x)mod m=((a^y mod m)^x) mod m=(a^xy) mod m.*
8) *B вычисляет Kab =((La)^y)mod m=((a^x mod m)^y) mod m=(a^xy) mod m.*

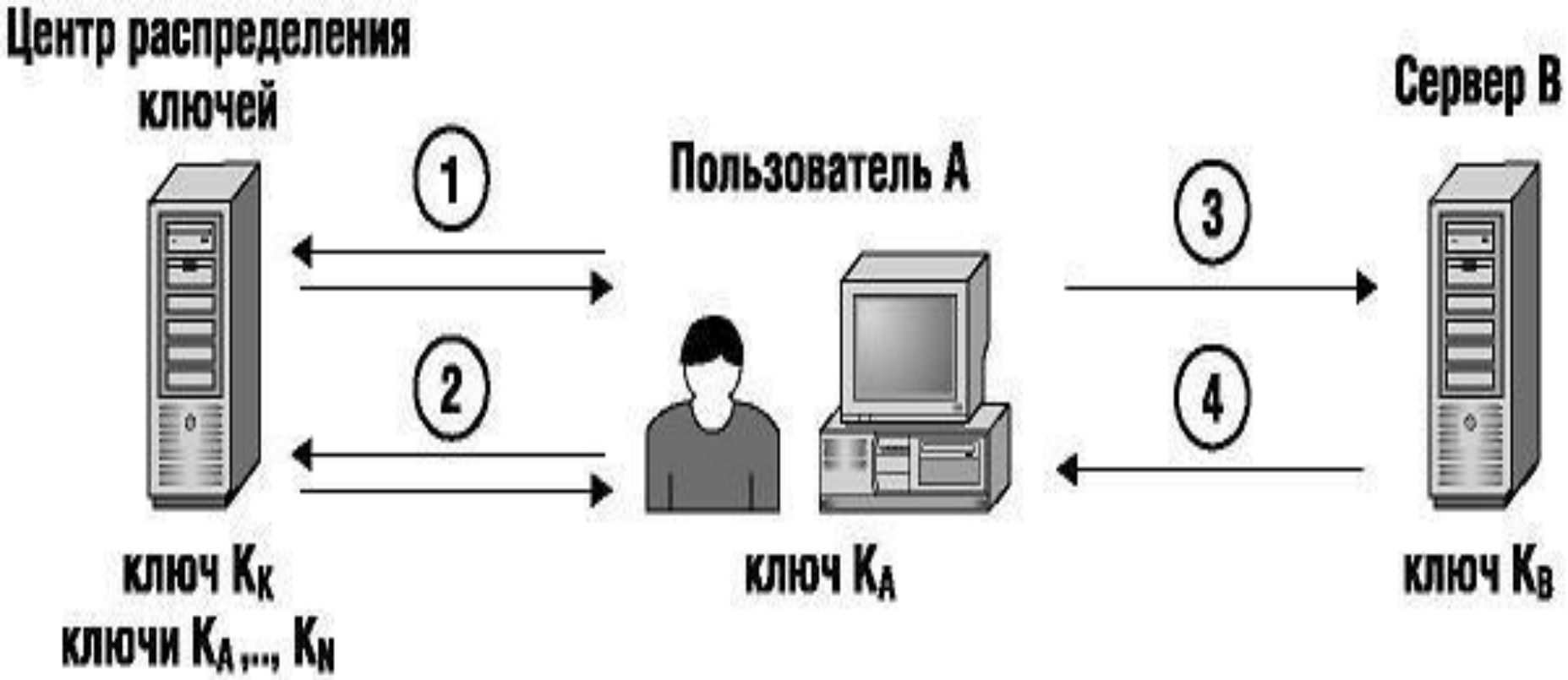
Таким образом оба пользователя теперь имеют одинаковый секретный ключ Kab.

Аутентификация Kerberos

Для решения проблемы аутентификации, которая базировалась бы на шифровании, в Массачусетском технологическом институте в 1985 году была разработана система защиты информационных систем от вторжений, с специальным сервисом выдачи билетов. Она была названа Kerberos по имени трехглавого пса Цербера, охранявшего ворота в ад в греческой мифологии.

Такое название было выбрано, потому что в аутентификации участвовали три стороны: пользователь, сервер, к которому желает получить доступ пользователь, и сервер аутентификации, или *центр распределения ключей (ЦРК)*. Специальный сервер аутентификации предлагался в качестве доверенной третьей стороны, услугами которой могут пользоваться другие серверы и клиенты информационной системы.

Управление ключами



K_A – секретный ключ пользователя А

K_B – секретный ключ сервера В

K_K – секретный ключ ЦРК

K_A, ... , K_N – секретные ключи всех субъектов системы Kerberos, хранимые ЦРК

Управление ключами

Система Kerberos владеет секретными ключами обслуживаемых субъектов и помогает им выполнять *взаимную аутентификацию*.

1. получение пользователем билета *TGT* на билеты;
2. получение пользователем *билета на доступ к серверу*;
3. аутентификация пользователя сервером;
4. аутентификация сервера пользователем.

Рассмотрим более подробно аутентификацию в системе Kerberos (рис.), которая выполняется за четыре шага:

1) Сеанс начинается с получения пользователем А *билета для получения билета - Ticket-Granting Ticket (TGT)* от *ЦРК*.
2) Когда пользователь желает получить доступ к некоторому серверу В, то сначала отправляет запрос на *билет для доступа к этому серверу* вместе со своим билетом *TGT* в *ЦРК*. *TGT* содержит информацию о сеансе регистрации пользователя А и позволяет *ЦРК* оперировать, не поддерживая постоянно информацию о сеансе регистрации каждого пользователя.

Управление ключами

3) В ответ на свой запрос пользователь A получает зашифрованный сеансовый ключ S_A и *билет на доступ к серверу* B. Сеансовый ключ зашифрован секретным ключом, известным только пользователю A и *ЦРК*. *Билет на доступ к серверу* B содержит тот же самый сеансовый ключ, однако он шифруется секретным ключом, известным только серверу B и *ЦРК*.

4) Аутентификация происходит тогда, когда пользователь A и сервер доказывают знание своего секретного ключа. Пользователь шифрует метку времени и отправляет ее на сервер B. Сервер расшифровывает метку, увеличивает ее значение на единицу, вновь зашифровывает и отправляет шифротекст пользователю A. Пользователь A расшифровывает ответ, и если в нем содержится значение метки времени с приращением, то аутентификация завершается успешно, в противном случае - неудачно. После *взаимной аутентификации* сеансовый ключ может использоваться для шифрования сообщений, которыми обмениваются пользователь A и сервер B. Очевидно, что стороны должны доверять *ЦРК*, поскольку он хранит копии всех секретных ключей.

Распределение ключей в асимметричных криптосистемах

Важной проблемой всей криптографии с открытым ключом, в том числе и систем ЭЦП, является управление открытыми ключами. Необходимо обеспечить доступ любого пользователя к подлинному открытому ключу любого другого пользователя, защитить эти ключи от подмены злоумышленником, а также организовать отзыв ключа в случае его компрометации.

- Задача защиты ключей от подмены решается с помощью сертификатов. Сертификат позволяет удостоверить заключённые в нём данные о владельце и его открытый ключ подписью какого-либо доверенного лица. В централизованных системах сертификатов (например PKI) используются центры сертификации, поддерживаемые доверенными организациями. В децентрализованных системах (например PGP) путём перекрёстного подписывания сертификатов знакомых и доверенных людей каждым пользователем строится сеть доверия.

Управление ключами

- Инфраструктура безопасности для распространения *открытых ключей*, управления электронными *сертификатами* и ключами пользователей получила название *инфраструктуры открытых ключей - Public Key Infrastructure (PKI)*.
- Термин *"PKI"* является производным от названия базовой технологии - криптографии с *открытыми ключами*, обладающей уникальными свойствами и являющейся основой для реализации функций безопасности в распределенных системах.
- *Инфраструктура открытых ключей* реализуется не ради нее самой, а для поддержки безопасности других приложений. Существуют, безусловно, и другие механизмы безопасности, которые не используют криптографию *открытых ключей*, и они менее сложные, чем *PKI*. Однако *PKI* не только предлагает наиболее комплексное решение, но и снимает остроту многих проблем, свойственных более традиционным механизмам безопасности.
- Управлением ключами занимаются центры распространения сертификатов. Обратившись к такому центру пользователь может получить сертификат какого-либо пользователя, а также проверить, не отозван ли ещё тот или иной открытый ключ.

Управление ключами

Управление открытыми ключами может быть организовано с помощью оперативной или автономной службы каталогов. Основными проблемами являются аутентичность, целостность и достоверность. Аутентичность позволяет убедиться, что это ключ именно этого пользователя.

Во всех случаях обмена ключами должна быть обеспечена подлинность сеанса связи, которая обеспечивается с помощью: механизма «запрос-ответ» (процедура «рукопожатие» = установка виртуального канала); механизма отметки времени.

Задача распределения ключей сводится к построению протокола распределения, обеспечивающего:

1. Взаимное подтверждение подлинности участников сеанса.
2. Подтверждение достоверности сеанса механизмом «запрос-ответ» или отметки времени.
3. Использование минимального числа сообщений при обмене ключами.
4. Возможность исключения злоупотребления со стороны ЦРК (вплоть до отказа от его услуг).

Управление ключами

Целесообразно отделить процедуру подтверждения подлинности партнеров от собственно процедуры распределения ключей.

Метод достижения одновременно аутентичности и целостности при распределении открытых ключей заключается в использовании сертификатов.

Система, основанная на сертификатах, предполагает, что имеется центральный орган, и каждый пользователь может осуществить безопасное взаимодействие с центральным органом, для этого у каждого пользователя должен быть открытый ключ центрального органа.

Сертификатом открытого ключа C_A называют сообщение центрального органа, удостоверяющего целостность некоторого открытого ключа объекта A (может быть бумажный, электронный документ).

Например: сертификат открытого ключа для пользователя А , обозначаемый C_A, содержит отметку времени Т , идентификатор Id_A открытый ключ K_A, зашифрованный секретным ключом ЦРК $К_{црк}$:

$$C_A = E\ К_{црк}\ (T, Id_A, K_A)$$

Отметка времени Т используется для подтверждения актуальности сертификата и тем самым предотвращает повторы прежних сертификатов. Секретный ключ $К_{црк}$ известен только менеджеру ЦРК, а открытый ключ ЦРК известен всем абонентам

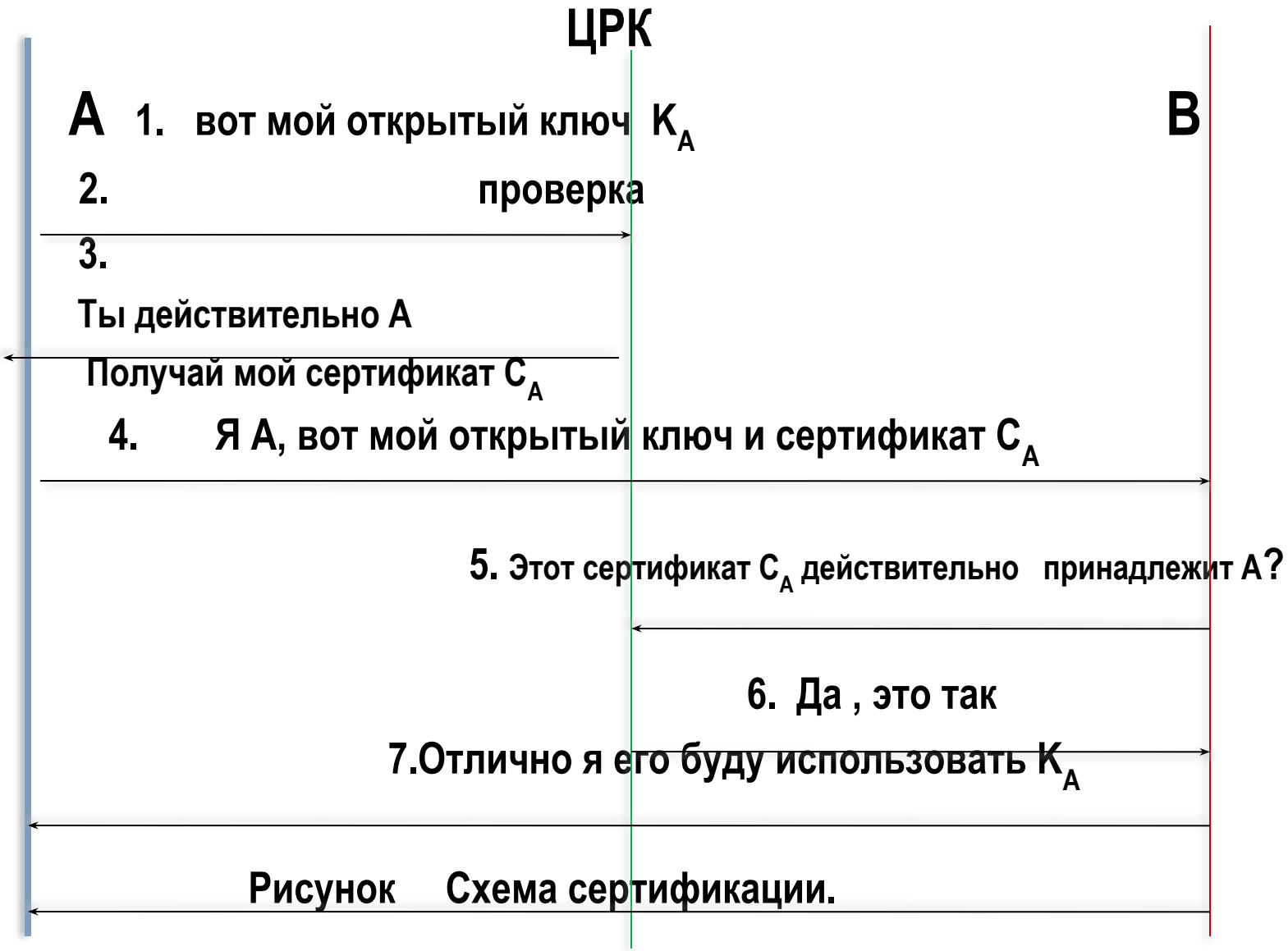


Рисунок Схема сертификации.

Управление ключами

Центр сертификации или Удостоверяющий центр (УЦ, англ. *Certification authority, CA*) — это организация или подразделение организации, которая выпускает сертификаты ключей электронной цифровой подписи, это компонент глобальной службы каталогов, отвечающий за управление криптографическими ключами пользователей.

Удостоверяющий центр - юридическое лицо или индивидуальный предприниматель, осуществляющие функции по созданию и выдаче сертификатов ключей проверки электронных подписей, а также иные функции, предусмотренные настоящим Федеральным законом (N 63-ФЗ «**ОБ ЭЛЕКТРОННОЙ ПОДПИСИ»**)

Открытые ключи и другая информация о пользователях хранится удостоверяющими центрами в виде цифровых сертификатов

Сертификат ключа проверки электронной подписи должен содержать следующую информацию:

1) даты начала и окончания срока его действия;

2) фамилия, имя и отчество (если имеется) - для физических лиц, наименование и место нахождения - для юридических лиц или иная информация, позволяющая идентифицировать владельца сертификата ключа проверки электронной подписи;

3) ключ проверки электронной подписи;

4) наименование используемого средства электронной подписи и (или) стандарты, требованиям которых соответствуют ключ электронной подписи и ключ проверки электронной подписи;

5) наименование удостоверяющего центра, который выдал сертификат ключа проверки электронной подписи;

6) иная информация, предусмотренная частью 2 статьи 17 настоящего Федерального закона, - для квалифицированного сертификата.

Сертификат ключа проверки электронной подписи прекращает свое действие:

1) в связи с истечением установленного срока его действия;

2) на основании заявления владельца сертификата ключа проверки электронной подписи, подаваемого в форме документа на бумажном носителе или в форме электронного документа;

3) в случае прекращения деятельности удостоверяющего центра без перехода его функций другим лицам;

4) в иных случаях, установленных настоящим Федеральным законом, другими федеральными законами, принимаемыми в соответствии с ними нормативными правовыми актами или соглашением между удостоверяющим центром и владельцем сертификата ключа проверки электронной подписи.

Удостоверяющий центр обеспечивает следующую функциональность (ФЗ №63 «Об электронной подписи):

1) создает сертификаты ключей проверки электронных подписей и выдает такие сертификаты лицам, обратившимся за их получением (заявителям);

2) устанавливает сроки действия сертификатов ключей проверки электронных подписей;

3) аннулирует выданные этим удостоверяющим центром сертификаты ключей проверки электронных подписей;

4) выдает по обращению заявителя средства электронной подписи, содержащие ключ электронной подписи и ключ проверки электронной подписи (в том числе созданные удостоверяющим центром) или обеспечивающие возможность создания ключа электронной подписи и ключа проверки электронной подписи заявителем;

Управление ключами

5) ведет реестр выданных и аннулированных этим удостоверяющим центром сертификатов ключей проверки электронных подписей (далее - реестр сертификатов), в том числе включающий в себя информацию, содержащуюся в выданных этим удостоверяющим центром сертификатах ключей проверки электронных подписей, и информацию о датах прекращения действия или аннулирования сертификатов ключей проверки электронных подписей и об основаниях таких прекращения или аннулирования;

6) устанавливает порядок ведения реестра сертификатов, не являющихся квалифицированными, и порядок доступа к нему, а также обеспечивает доступ лиц к информации, содержащейся в реестре сертификатов, в том числе с использованием информационно-телекоммуникационной сети "Интернет";

7) создает по обращениям заявителей ключи электронных подписей и ключи проверки электронных подписей;

8) проверяет уникальность ключей проверки электронных подписей в реестре сертификатов;

9) осуществляет по обращениям участников электронного взаимодействия проверку электронных подписей;

10) осуществляет иную связанную с использованием электронной подписи деятельность.

ПО удостоверяющего центра выполняет следующие действия:

1) Генерация секретных и открытых ключей Главных абонентов УЦ, сертификатами которых заверяются сертификаты пользователей. Сертификаты главных абонентов могут быть самоподписанными или заверенными вышестоящим УЦ. Первое издание сертификата подписи абонентов происходит в УЦ вместе с генерацией секретного ключа для него.
2) Издание и регистрация сертификатов ЭП по запросу абонентов сети. Запрос на сертификат представляет собой шаблон сертификата, содержащий информацию об абоненте, его новый открытый ключ подписи, предполагаемый срок действия сертификата, а также другие параметры, соответствующие стандарту X.509. После заполнения полей сертификата сертификат через Центр управления отправляется к пользователю на компьютер.
3) Отзыв сертификатов (например, в случае компрометации ключей).
4) Приостановление действия сертификатов, возобновление действия сертификатов ЭП абонентов сети. Эти действия выполняются администратором УЦ. Справочник отозванных сертификатов рассылается абонентам сети.
5) Регистрация справочников сертификатов ЭП главных абонентов других УЦ .

Управление ключами

6) Регистрация справочников отозванных сертификатов ЭП из других УЦ.
7) Издание и регистрация сертификатов ЭП для внешних пользователей выполняется только по запросу из Центра регистрации. Запрос может быть зарегистрирован или отклонен.
8) Отзыв сертификатов, приостановление действия сертификатов, возобновление действия сертификатов ЭП внешних пользователей может происходить по запросу из ЦР, или самим Администратором УЦ без запроса из ЦР. Справочники отозванных сертификатов рассылаются по узлам сети.
9) Просмотр запросов и сертификатов ЭП. В программе УЦ возможен просмотр любых запросов, сертификатов, действующих списков отзыва, сохранение их в файл или вывод на печать.
10) Сервисные функции УЦ.

Управление ключами

- Российские удостоверяющие центры

- Удостоверяющий Центр ЗАО "ПФ "СКБ Контур"
- Удостоверяющий Центр ФГУП НИИ "Восход"
- Удостоверяющий Центр DIP
- ООО Межрегиональный Удостоверяющий Центр
- НП "Национальный Удостоверяющий центр"
- ЗАО «АНК»
- ООО ПНК
- ЗАО Удостоверяющий Центр (Нижний Новгород)
- ЗАО Удостоверяющий Центр (Санкт-Петербург)
- ООО "Компания «Тензор»
- ОАО «Электронная Москва»
- Центр сертификации «Стандарт Тест»
- Всероссийский центр сертификации
- Удостоверяющий центр ЗАО "Сервер-Центр" (Владивосток)
- Удостоверяющий центр "ДСЦБИ "МАСКОМ" (Хабаровск)

Управление ключами

— Центры выдачи ЭП, Кемерово

- ОО Научно-технический центр Атлас

г. Кемерово, ул. Сарыгина, д. 29, к.517

8(3842)59-58-53 8(3842)59-58-71 8(3842)28-55-73 8(3842)44-11-00
atlas-k@regit.ru
http://atlas.regit.ru

- ООО СибНэт
- г. Кемерово, ул. Рукавишникова, д. 9А, оф. 9
- 8(3842)75-50-05 8(3842)75-16-32 8(961)705-66-62
  kontur@sib-net.ru

Аутентификация при помощи сертификатов

В том случае, когда пользователи имеют *сертификаты открытых ключей*, необходимость в *ЦРК* отпадает. Это не означает, что отпадает необходимость в доверии и третьих сторонах; просто доверенной третьей стороной становится УЦ. Однако УЦ не участвует в обмене протоколами, и в отличие от ситуации с *ЦРК*, если УЦ недоступен, аутентификация по-прежнему может быть выполнена.

Аутентификацию при помощи *сертификатов* обеспечивают несколько распространенных протоколов, в частности, наиболее известный и широко распространенный протокол Secure Socket Layer (SSL), который применяется практически в каждом web-браузере.

Помимо него применяются протоколы Transport Layer Security (TLS), Internet Key Exchange (IKE), S/MIME, PGP и Open PGP. Каждый из них немного по-своему использует *сертификаты*, но основные принципы - одни и те же.

Управление ключами

[Запрос на аутентификацию]	Необязателен
[Token ID]	Необязателен. Идентифицирует тип аутентификации (взаимная), версию и идентификатор протокола
CertA	Сертификат пользователя ***A***
CertB	Сертификат сервера ***B***
Token BA1	ran B, случайное число, сгенерированное сервером ***B***
Token AB	ran A, ran B, name B, подписанные пользователем ***A***, где ran A — случайное число, сгенерированное пользователем ***A***, ran B — значение, повторенное из Token BA1 и name B — имя сервера
Token BA2	ran A, ran B, name A, подписанные сервером ***B***, где ran A — случайное число, сгенерированное пользователем ***A***, ran B — значение, повторенное из Token BA1 и Token AB, и name A — имя сервера

Рисунок иллюстрирует типичный обмен сообщениями при аутентификации на базе *сертификатов*, использующий ЭП.

Обмен соответствует стандарту аутентификации субъектов на основе криптографии с *открытыми ключами*. Во многих протоколах предусматривается, что клиент направляет запрос серверу для того, чтобы инициировать аутентификацию. Такой подход, характерный, например, для дополнений аутентификации и шифрования к протоколу Internet File Transfer Protocol, гарантирует, что и пользователь, и сервер поддерживают один и тот же механизм аутентификации. Некоторые протоколы не требуют этого подготовительного шага.

Если сервер B поддерживает метод аутентификации, запрашиваемый пользователем A, то начинается обмен сообщениями.

1) Сообщение Token ID уведомляет о том, что будет выполняться *взаимная аутентификация*, а также содержит номер версии протокола и идентификатор протокола. Хотя этот идентификатор не обязателен, он намного упрощает процедуру и поэтому обычно используется. Пользователь A ожидает сообщение Token BA1 от сервера B.

2) Идентификатор протокола в Token ID позволяет пользователю A удостовериться, что сервер B отправляет ожидаемое сообщение. Token BA1 состоит только из случайного числа ran B, это - своего рода запрос, корректным ответом должна быть цифровая подпись числа ran B. Пользователь A подписывает ответ и отправляет свой *сертификат* ключа подписи, для того чтобы сервер B при помощи *открытого ключа* мог выполнить валидацию подписи.

3) Пользователь A подписывает последовательность из трех элементов: свой запрос ran A, запрос сервера ran B и имя сервера name B.

4) Получив ответ Token AB от пользователя A, сервер B проверяет, совпадает ли значение ran B с соответствующим значением в сообщении Token BA1, а по значению name B устанавливает, действительно ли пользователь A желает пройти аутентификацию сервера B.

5) Если какая-либо из проверок дает отрицательный результат, то и аутентификация завершается неудачно. В противном случае сервер B проверяет подлинность *сертификата* пользователя A и его цифровую подпись, если *сертификат* и подпись валидны, то аутентификация пользователя A сервером B прошла успешно. Ответ сервера B пользователю A завершает *взаимную аутентификацию*.

6) Ответ сервера Token BA2 состоит из заверенной цифровой подписью последовательности трех элементов: ran A, ran B и name A, где ran A - запрос, сгенерированный A, ran B - исходный запрос сервера B, а name A - имя пользователя A. Получив ответ сервера, пользователь A убеждается, что ran A имеет то же самое значение, что и в сообщении Token AB, а проверяя значение name A - что сервер B намерен аутентифицировать именно его (пользователя A). Если какая-либо из проверок дает отрицательный результат, то и аутентификация завершается неудачно. В противном случае пользователь A проверяет подлинность *сертификата* сервера B и его цифровой подписи. Если они валидны, то пользователь A аутентифицировал сервер B, и *взаимная аутентификация* выполнена.

Итак, механизмы аутентификации при помощи *сертификатов* поддерживают аутентификацию в открытой сети, на многих удаленных серверах, и обеспечивают *взаимную аутентификацию*. В отличие от системы Kerberos. протоколы аутентификации на базе *сертификатов* не требуют активного участия третьих сторон. Для успешной аутентификации должны быть доступны только пользователь и сервер.

Слово "стеганография" в переводе с греческого буквально означает "тайнопись" (steganos - секрет, тайна; graphy - запись).

Стеганография - это метод организации связи, который собственно скрывает само наличие связи. В отличие от криптографии, где неприятель точно может определить является ли передаваемое сообщение зашифрованным текстом, методы стеганографии позволяют встраивать секретные сообщения в безобидные послания так, чтобы невозможно было заподозрить существование встроенного тайного послания.

Стеганография занимает свою нишу в обеспечении безопасности: она не заменяет, а дополняет криптографию. Сокрытие сообщения методами стеганографии значительно снижает вероятность обнаружения самого факта передачи сообщения. А если это сообщение к тому же зашифровано, то оно имеет еще один, дополнительный, уровень защиты.

Стеганографическая система или стегосистема - совокупность средств и методов, которые используются для формирования скрытого канала передачи информации

Процесс стеганографии условно можно разделить на несколько этапов:

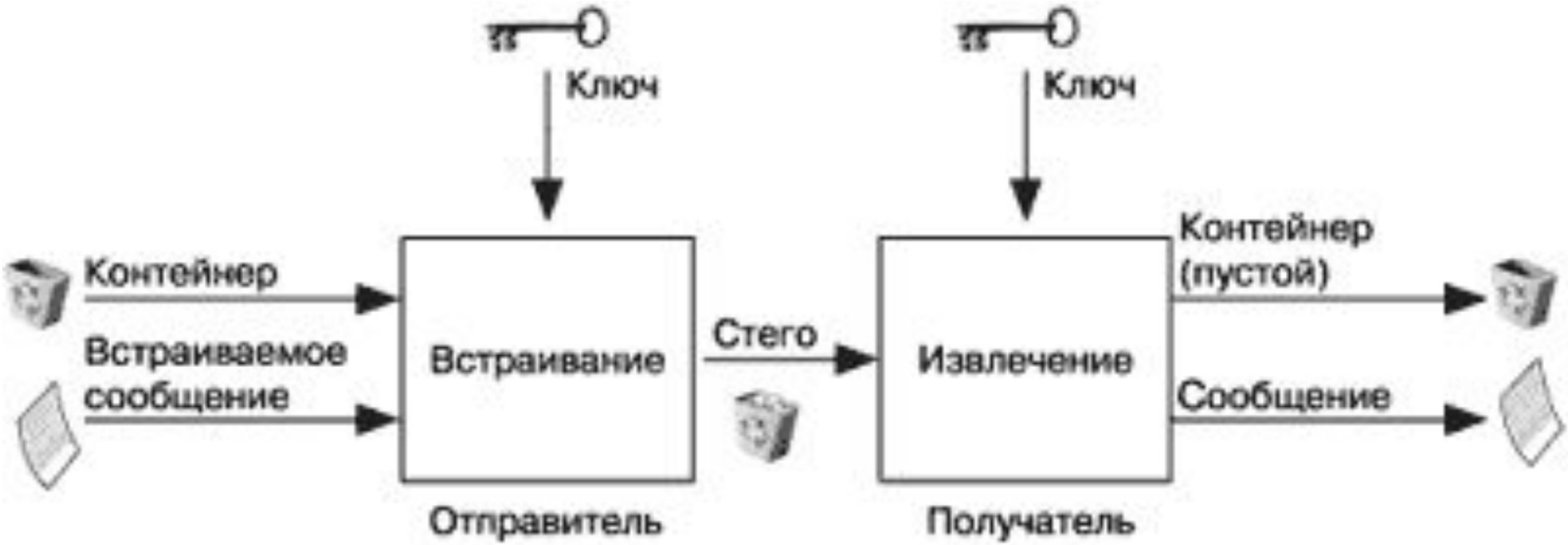
1) выбор файла, который необходимо скрыть (сообщение);
2) выбор файла, используемого для сокрытия информации (файл-контейнер);
3) выбор стеганографической программы;
4) кодирование (встраивание) файла. На новый файл устанавливается защита паролем;
5) отправление скрытого сообщения по электронной почте и его декодирование.

Требования, предъявляемые к современной компьютерной стеганографии:

1) методы сокрытия должны обеспечивать аутентичность и целостность файла;
2) изначально предполагается, что преступнику знакомы все возможные стеганографические методы;
3) безопасность методов основывается на сохранении стеганографическим преобразованием основных свойств открыто передаваемого файла (контейнера) при внесении в него секретного сообщения и некоторой неизвестной преступнику информации - ключа;
4) если факт сокрытия сообщения стал известен противнику, то извлечение секретных данных представляет сложную вычислительную задачу

Обобщенная модель стегосистемы представлена на рис. 1.

- **Контейнер - любая информация, предназначенная для сокрытия тайных сообщений.**
- **Пустой контейнер - контейнер без встроенного сообщения; заполненный контейнер или стего - контейнер, содержащий встроенную информацию.**
- **Встроенное (скрытое) сообщение - сообщение, встраиваемое в контейнер.**



- **Стегоключ** или просто **ключ** - секретный ключ, необходимый для сокрытия информации. В зависимости от количества уровней защиты (например, встраивание предварительно зашифрованного сообщения) в стегосистеме может быть один или несколько стегоключей.

По аналогии с криптографией, по типу стегоключа стегосистемы можно подразделить на два типа:

- с секретным ключом;
- с открытым ключом.

В **стегосистеме с секретным ключом** используется один ключ, который должен быть определен либо до начала обмена секретными сообщениями, либо передан по защищенному каналу.

В стегосистеме с открытым ключом для встраивания и извлечения сообщения используются разные ключи, которые различаются таким образом, что с помощью вычислений невозможно вывести один ключ из другого. Поэтому один ключ (открытый) может передаваться свободно по незащищенному каналу связи. Кроме того, данная схема хорошо работает и при взаимном недоверии отправителя и получателя.

Пример встраивания

Предположим, что в качестве контейнера используется 24 битовое изображение размером 800x600 (графика среднего разрешения).

Оно занимает около полутора мегабайта памяти (800x600x24/3 = 1440000 байт).

Каждая цветовая комбинация тона (пиксела - точки) – это комбинация трех основных цветов –красного, зеленого и синего, которые занимают каждый по 1 байту (итого по 3 на точку).

Если для хранения секретной информации использовать наименьший значащий бит (Least Significant Bits – LSB) каждого байта, то получим по 3 бита на каждый пиксел. Емкость изображения носителя составит – 800x600x3/8=180000 байт. При этом биты в каких то точках будут совпадать с битами реального изображения, в других – нет, но, главное, что на глаз определить такие искажения практически невозможно.

Программные продукты стеганографии

Название программы	Описание
Gif-It-Up 1.0 for Windows	Прячет данные в GIF файлах, выполняя подстановку скрытых цветов в изображение
EZStego	Модифицирует наименее значимые биты (LSB) яркости точек GIF и PICT, изменяя их цветовую палитру
DiSi-Steganograph	DOS–приложение. Прячет данные в графических файлах PCX
Hide and Seek	Прячет данные в GIF файлах, скрываемые данные кодирует алгоритмом шифрования Blowfish. Осуществляет случайный выбор точек для хранения внедряемых данных.
MP3Stego	Внедряет данные в звуковые файлы формата MP3, который широко рапространен.
Steganos	Программа с ассистентом (Wizard) кодирует и прячет файлы в форматах DIB, BMP, VOC, WAV, ASCII и HTML
Steganography Tools 4	Предварительно кодирует данные с помощью алгоритмов шифрования IDEA, MPJ2, DES, TripleDES и NSEA, а затем прячет их в графических файлах, звуковых (WAV) файлах или свободных секторах флоппи-дисков

Steganography

Original Image

Modified Image

Hide Message

Original Message

Очень большой секрет

Extract Message

Extractedl Message

Очень большой секрет

Бесплатные программы, информацию об их получении, а также другую информацию о методах стеганографии можно найти

на веб сайтах –

http://www.cl.cam.ac.uk/~fapp2/steganography/,

http://www.demcom.com/english/steganos/,

http://www.signumtech.com,

http://www/digimark/com

В настоящее время можно выделить три тесно связанных между собой и имеющих одни корни направления приложения стеганографии:

- **сокрытие данных (сообщений),**
- **цифровые водяные знаки,**
- **заголовки.**

Обобщенная модель приложений стеганографии представлена на рис.

- **Сокрытие внедряемых данных,** которые в большинстве случаев имеют большой объем, предъявляет серьезные требования к контейнеру: размер контейнера в несколько раз должен превышать размер встраиваемых данных.
- **Цифровые водяные знаки** используются для защиты авторских или имущественных прав на цифровые изображения, фотографии или другие оцифрованные произведения искусства. Основными требованиями, которые предъявляются к таким встроенным данным, являются надежность и устойчивость к искажениям. Цифровые водяные знаки имеют небольшой объем, однако, с учетом указанных выше требований, для их встраивания используются более сложные методы, чем для встраивания просто сообщений или заголовков.
- **Заголовки,** используется в основном для маркирования изображений в больших электронных хранилищах (библиотеках) цифровых изображений, аудио- и видеофайлов. В данном случае стеганографические методы используются не только для внедрения идентифицирующего заголовка, но и иных индивидуальных признаков файла. Внедряемые заголовки имеют небольшой объем, а предъявляемые к ним требования минимальны: заголовки должны вносить незначительные искажения и быть устойчивы к основным геометрическим преобразованиям.

Надежность криптосистем

Стойкость криптосистемы зависит от:

- длины ключа
- объема ключевого пространства
- алгоритма
- криптографического протокола

Американские криптологи опубликовали результаты исследований взлома алгоритма симметричного шифрования DES. Атаки осуществлялись методом полного перебора. Бюджет определяет мощность специализированного компьютера.

Надежность криптосистем

кто атакует	бюджет	сложность атаки		стойкий ключ
		40 бит	56 бит	
хакер	$1000	1 нед.	никогда	45 бит
малый бизнес	$10000	12 мин.	556 дней	64 бит
крупная компания	$10 млн.	0,005 сек.	6 мин.	70 бит
федеральное агентство	$300 млн.	0,0002 сек.	12 сек.	75 бит

Выбор ключей

Если пользователь сам себе выбирает ключ, то скорее всего он выберет **Ivanov**, чем **&7)d/***

В атаке полного перебора используется словарь, включающий:

1. ФИО и данные отправителя в различных комбинациях;
2. Имена людей, героев книг, животных, мифов;
3. Слова, полученные после внесения изменений в слова 1,2 пунктов;
4. Слова полученные заменой строчных букв на заглавные;
5. Слова на различных иностранных языках и т.д.

Выбор ключей

Хороший ключ – случайный, но его трудно запомнить.

Более привлекателен подход, когда вместо отдельного слова используется достаточно длинное предложение на русском, английском или другом языке, которое преобразуется в ключ. Такое предложение в криптографии называется *паролем*.

Для преобразования пароля в псевдослучайный битовый ключ можно воспользоваться любой хэш-функцией.

Пароль следует выбирать достаточно длинным, чтобы полученный в результате преобразования ключ, был случайным.

Из теории информации известно, что каждая буква содержит ≈ 1,3 бита информации.

Чтобы получить 64 битовый ключ, пароль должен состоять из 49 букв, что соответствует фразе примерно из 10 слов.

Для выбора парольной фразы лучше воспользоваться творчеством мало известных поэтов.

Выбор ключей

Для хранения ключа (запомнить его сложно, если он случаен) можно использовать пластиковую карточку с ПЗУ.

С целью уменьшения вероятности компрометации ключа можно разделить его на две части:

- одну часть ключа реализовать в виде ПЗУ-ключа;
- другую часть разместить в памяти компьютера.

Задача: Как выбрать надежный криптоалгоритм?

Способы решения:

1. Воспользоваться надежным известным алгоритмом, если нет информации о его раскрытии.
2. Довериться специализированной сертифицированной фирме.
3. Обратиться к независимому эксперту.
4. Обратиться за поддержкой в соответствующее правительственное ведомство.
5. Создать собственный криптографический алгоритм.

Аппаратное или программное шифрование?

Большинство средств криптографической защиты данных реализовано в виде специализированных физических устройств. Эти устройства встраиваются в линию связи и осуществляют шифрование всей передаваемой информации.

Преобладание аппаратного шифрования обусловлено:

- более высокой скоростью;
- аппаратуру легче защитить физически от проникновения извне. Аппаратура помещается в специальные контейнеры, которые могут покрываться химическим составом. И в результате любая попытка преодолеть защитный слой приводит к уничтожению внутренней логической схемы чипа.
- аппаратура шифрования более проста в установке в различных местах (телефон, модем).

Надежность криптосистем

Аппаратное или программное шифрование?

Преимущества программного шифрования:

- любой алгоритм легко копируется (на другой компьютер);
- алгоритм прост в использовании;
- его не трудно модифицировать.

Надежность криптосистем

Симметричное или асимметричное шифрование шифрование?

Симметрические алгоритмы имеют меньшую длину ключа и работают быстрее, чем асимметричные системы, но для них существует проблема передачи секретного ключа.

Асимметричные криптосистемы используют два ключа, один из которых открытый.

У этих систем по мнению У. Диффи разное назначение:

- симметричные криптосистемы используются для шифрования данных,
- асимметричные системы - для ЭП, шифрования ключей.

Информацию перед шифрованием желательно сжимать. После шифрования сжать не удастся. Если удается сжать файлы после шифрования, то алгоритм шифрования недостаточно хорош.

Шифрование файлов

Особенности:

1. нередко после шифрования незашифрованная копия хранится там же или на другом диске.
2. Размер блока в блочных алгоритмах шифрования может значительно превышать размер отдельных порций в структурированном файле.
3. Скорость шифрования выбранного алгоритма должна быть больше скорости работы дисковых устройств.
4. Работа с ключами – лучше шифровать каждый файл на отдельном ключе, а затем зашифровать все ключи мастер-ключом.
5. Перед шифрованием лучше провести сжатие файла.