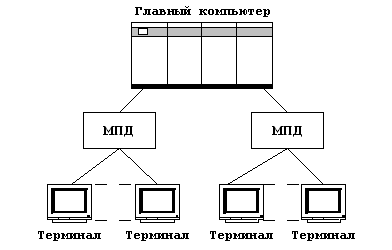
**53. Системы безопасности. Архитектуры систем безопасности (одноранговые и централизованные системы, распределенные службы каталогов). Примеры реализаций. Контроль доступа к ресурсам (ACL и мандатный доступ). Виды аутентификации. Аппаратные средства аутентификации. Алгоритмы аутентификации по открытым каналам.**

**Централизованная -**  Архитектура терминал-главный компьютер (terminal-host computer architecture) – это концепция информационной сети, в которой вся обработка данных осуществляется одним или группой главных компьютеров.



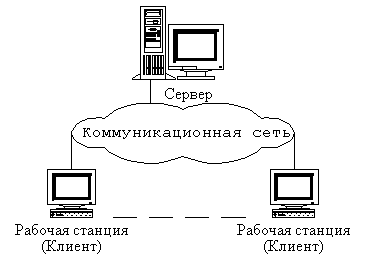
Мпд – модуль передачи данных

**Одноранговая архитектура -** (peer-to-peer architecture) – это концепция информационной сети, в которой ее ресурсы рассредоточены по всем системам. Данная архитектура характеризуется тем, что в ней все системы равноправны. К одноранговым сетям относятся малые сети, где любая рабочая станция может выполнять одновременно функции файлового сервера и рабочей станции. В одноранговых ЛВС дисковое пространство и файлы на любом компьютере могут быть общими.

Одноранговые ЛВС являются наиболее легким и дешевым типом сетей для установки. При соединении компьютеров, пользователи могут предоставлять ресурсы и информацию в совместное пользование.

Минусы – отсутствие центрального управления. Отдельная настройка доступа на каждый ресурс.

**Архитектура клиент-сервер** (client-server architecture) – это концепция информационной сети, в которой основная часть ее ресурсов сосредоточена в серверах, обслуживающих своих клиентов. Рассматриваемая архитектура определяет два типа компонентов: серверы и клиенты.

* 

В сетях с выделенным файловым сервером на выделенном автономном ПК устанавливается серверная сетевая операционная система. Этот ПК становится сервером. ПО, установленное на рабочей станции, позволяет ей обмениваться данными с сервером. Наиболее распространенные сетевые операционная системы:

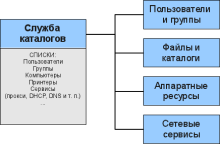
-    NetWare фирмы Novel;

-    Windows NT фирмы Microsoft;

-    UNIX фирмы AT&T;

-    Linux.

**Служба каталогов** — это сетевой сервис, представляющий централизованные средства управления ресурсами автоматизированной системы. Под ресурсами подразумеваются все компоненты сетевой инфраструктуры, которые используются для выполнения функций АСУ: пользователи, файлы и каталоги, устройства, сетевые сервисы и т.д



Как правило, служба каталогов состоит из базы данных, в которой размещены сведения о сетевых ресурсах и серверного ПО, представляющего механизмы доступа к этой базе.

Основными функциями службы каталогов являются следующие:

* Управление пользователями и группами (создание/удаление, настройка прав доступа).
* Управление ресурсами (представление в общий доступ, установка ограничений, удаленное администрирование и т.п.).
* Разграничение прав доступа (как правило, на уровне пользователей, групп и отдельных ресурсов).

Среди дополнительных функций сервиса каталогов можно указать, например, такие:

* поиск ресурсов;
* распространение сетевых политик;
* интеграция с другими сервисами.

**Сетевая политика** — совокупность правил, определяющих методы и средства взаимодействия с общими ресурсами в корпоративной сети.

Примеры службы каталогов

**NIS** (Сетевая Информационная Служба) — служба каталогов, разработанная и реализованная Sun Microsystems для систем на основе UNIX. NIS первоначально назывались Yellow Pages (YP), но из-за проблем с торговым знаком Sun изменила это название. Старое название (yp) используется в названиях утилит NIS.

**Домен NIS** — это совокупность доверенных ресурсов с уникальным в пределах сети именем. Имя домена NIS и способ именования ресурсов напоминает адресацию в [системе доменных имен](http://www.4stud.info/networking/dns.html) (DNS), но никакого отношения к DNS не имеет. Информация о домене хранится на основном сервере NIS и реплицируется на вторичные сервера наравне с прочими ресурсами. Один основной сервер может вести базы нескольких доменов NIS.

**Active Directory** от майкрософт

Больше инфы <http://www.4stud.info/networking/directory-service.html>

**Access Control List** или **ACL** — список контроля доступа, который определяет, кто или что может получать доступ к конкретному [объекту](http://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82), и какие именно операции разрешено или запрещено этому [субъекту](http://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_%28%D1%84%D0%B8%D0%BB%D0%BE%D1%81%D0%BE%D1%84%D0%B8%D1%8F%29) проводить над объектом.

Список доступа представляет собой структуру данных (обычно таблицу), содержащую записи, определяющие права индивидуального пользователя или группы на специальные системные объекты, такие как [программы](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), процессы или файлы. Эти записи также известны как ACE ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Access Control Entries*) в [операционных системах](http://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) [Microsoft Windows](http://ru.wikipedia.org/wiki/Microsoft_Windows) и [OpenVMS](http://ru.wikipedia.org/wiki/OpenVMS). В операционной системе [Linux](http://ru.wikipedia.org/wiki/Linux) и [Mac OS X](http://ru.wikipedia.org/wiki/Mac_OS_X) большинство файловых систем имеют расширенные атрибуты, выполняющие роль ACL. Каждый объект в системе содержит указатель на свой ACL. Привилегии (или полномочия) определяют специальные права доступа, разрешающие пользователю **читать** из ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *read*), **писать** в ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *write*), или **исполнять** ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *execute*) объект. В некоторых реализациях ACE могут определять право пользователя или группы на изменение ACL объекта.

**Мандатное управление доступом** ([англ.](http://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Mandatory access control, MAC*) — разграничение [доступа](http://ru.wikipedia.org/w/index.php?title=%D0%94%D0%BE%D1%81%D1%82%D1%83%D0%BF&action=edit&redlink=1) субъектов к объектам, основанное на назначении метки конфиденциальности для информации, содержащейся в объектах, и выдаче официальных разрешений (допуска) субъектам на обращение к информации такого уровня конфиденциальности. Также иногда переводится как **Принудительный контроль доступа**. Это способ, сочетающий защиту и ограничение прав, применяемый по отношению к компьютерным процессам, данным и системным устройствам и предназначенный для предотвращения их нежелательного использования.

*В чем фишка – в acl есть такой перец как «владелец документа» и на каждый ресурс можно составлять свою матрицу прав. При мандатном доступе никаких хозяев нет права выделяется и контролируется только системой, на основе первоначальных меток документа и выделенных прав пользователей. Короче acl более гибче, мандатный доступ – жестче.*

**Аутентификация** (с греч. реальный или истинный) - процедура установления принадлежности пользователю информации в системе предъявленного им идентификатора.

Не стоит путать с авторизацией - проверкой: имеет авторизованный объект права на работу в системе?.

## Виды аутентификации

### Слабая Аутентификация

### На основе одного не физического параметра, например пароля или секретного вопроса.

### Сильная Аутентификация

Параллельно, по необходимости, используется сильная или многофакторная аутентификация - на основе двух или более факторов. В этом случае для аутентификации используются не только информация известна пользователю, но и дополнительные факторы. Например:

* свойство, которым обладает субъект;
* знание - информация, которую знает субъект;
* владение - вещь, которой обладает субъект.

## Способы аутентификации

### Парольная

Осуществляется на основе владения пользователем определенной конфиденциальной информации.

### Биометрическая

Биометрическая аутентификация основана на уникальности определенных антропометрических характеристик человека.

1. Настройки голоса.
2. Узор радужной оболочки глаза и карта сетчатки глаза.
3. Черты лица.
4. Форма ладони.
5. Отпечатки пальцев.
6. Форма и способ подписи.

### С помощью уникального предмета

Осуществляется с помощью дополнительных предметов ( токен, смарт-карта ) или атрибутов (криптографический сертификат ).

**Аппаратные средства аутентификации**

**Автономные токены** – это мобильные персональные устрой-ства, не подсоединяемые к компьютеру, которые имеют собст-венный источник питания. Эти устройства позволяют пользо-вателю аутентифицировать себя на серверах, используя или одноразовый пароль (токены с использованием ОТР - One-Time Password), или метод запрос/ответ.

Суть метода запрос/ответ в том, что:

􀂃 пользователь вводит свой ID на рабочей станции;

􀂃 ID передается по сети в открытом виде;

􀂃 сервер аутентификации генерирует случайный запрос, который передается поль-зователю по сети в открытом виде;

􀂃 пользователь вводит запрос в аутентификационный токен:

􀂃 токен пользователя зашифровывает этот запрос с помощью некого алгоритма шифрования и секретного ключа пользователя и результат отображается на экра-не;

􀂃 пользователь вводит результат на рабочей станции и ПО возвращает его серверу;

􀂃 cервер зашифровывает то же самое случайное число (запрос);

􀂃 при совпадении результатов процесс запрос/ответ в существующей системе аутен-тификации успешно завершается.

**USB-токены** – устройства, которые подключаются к стандартным портам USB и со-держат микроконтроллер и/или чип смарт-карты с операционной системой.

USB-токены:

􀂃 позволяют осуществлять строгую двухфакторную аутентифика-цию пользователя;

􀂃 обеспечивают функции шифрования и ЭЦП (цифровую подпись) пользователя;

􀂃 напрямую подключаются к USB-порту компьютера (не требуют считывателей);

􀂃 не нуждаются в дополнительном программном обеспечении, устанавливаемом на сервера (в отличие от OTP-токенов).

**Электронные ключи** для авторизации и лицензирования про-граммного обеспечения и его защиты от несанкционированного ис-пользования.

Ключи выпускаются как для параллельных, так и для USB-портов.

Кроме аппаратных токенов для аутентификации пользователей ши-роко применяются смарт-карты.

**Смарт-карты для IT-безопасности** – это пластиковые кар-точки размером с кредитку, содержащие чип (микропроцессор) для криптографических вычислений (ЭЦП, шифрование) и встроенную защищенную память для хранения информации (данные о пользователе, криптографические ключи, сертифи-каты и пр.).

# Алгоритмы аутентификации по открытым каналам.

* [Инфраструктура открытого ключа](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D1%80%D0%B0%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D1%85_%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%B9)

# Протокол Диффи — Хеллмана. Смысл – зная некий алгоритм несколько станций передают по открытым каналам случайные числа друг другу на основе которых генерят закрытый ключ используемый для шифрования на эту сессию.

При работе алгоритма каждая сторона:

1. генерирует случайное [натуральное число](http://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%82%D1%83%D1%80%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) *a* — *закрытый ключ*
2. совместно с удалённой стороной устанавливает *открытые параметры* *p* и *g* (обычно значения *p* и *g* генерируются на одной стороне и передаются другой), где

*p* является [случайным простым числом](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE)

*(p-1)/2* также должно быть [случайным простым числом](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D0%BE%D0%B5_%D1%87%D0%B8%D1%81%D0%BB%D0%BE) (для повышения безопасности)[[7]](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB_%D0%94%D0%B8%D1%84%D1%84%D0%B8_%E2%80%94_%D0%A5%D0%B5%D0%BB%D0%BB%D0%BC%D0%B0%D0%BD%D0%B0#cite_note-7)

*g* является [первообразным корнем](http://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B2%D0%BE%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D1%80%D0%B5%D0%BD%D1%8C_%28%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D1%87%D0%B8%D1%81%D0%B5%D0%BB%29) [по модулю](http://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8E) *p*

1. вычисляет *открытый ключ* *A*, используя преобразование над *закрытым ключом*

*A = ga* mod *p*

1. обменивается *открытыми ключами* с удалённой стороной
2. вычисляет *общий секретный ключ* *K*, используя открытый ключ удаленной стороны *B* и свой закрытый ключ *a*

*K = Ba* mod *p*

*К* получается равным с обеих сторон, потому что:

*Ba* mod *p = (gb* mod *p)a* mod *p =* ***gab*** **mod *p*** *= (ga* mod *p)b* mod *p = Ab* mod *p*

В практических реализациях для *a* и *b* используются числа порядка 10100 и *p* порядка 10300. Число *g* не обязано быть большим и обычно имеет значение в пределах первого десятка.