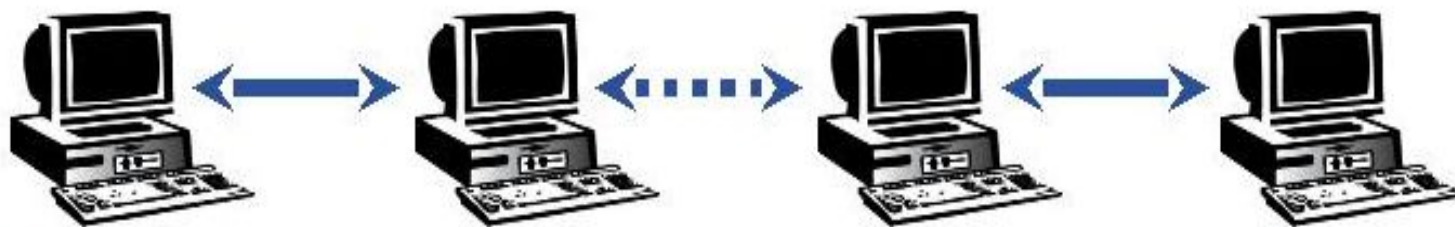


- **Объект (защиты):**
 - Пользователь, программа или компьютер
- **Параметры доступа**
 - Некоторые данные, обеспечивающие доказательства идентичности объекта
- **Аутентификация**
 - Проверка идентичности объекта защиты
- **Авторизация**
 - Определение множества прав и привилегий для объекта защиты
- **Конфиденциальность**
 - Шифрование сообщений для того чтобы только получатель мог его расшифровать
- **Целостность**
 - Гарантия того, что сообщение не было изменено во время передачи



- Как Пользователь может получить безопасный доступ к Ресурсу, не являясь зарегистрированным пользователем промежуточных узлов или хотя бы самого Ресурса?
- Как Ресурс узнает, кто такой Пользователь?
- Как определять права Пользователя и как определить какой доступ ему разрешён?

- **Опасность атак с других узлов**
 - Большие распределённые кластеры – идеальная мишень для атак злоумышленников (“отказ в обслуживании”)
- **Незаконное или ненадлежащее распространение данных и доступ к конфиденциальной информации**
 - Огромные доступные ресурсы хранения данных могут быть использованы, например для хранения “пиратской информации”
 - Всё больше пользователей обладают данными, которые требуют являются конфиденциальными (медицина)
- **Опасность, связанная с проникновением вирусов, сетевых червей и т.п.**
 - Высокоскоростные сети являются более быстрым источником распространения, чем обычный Интернет

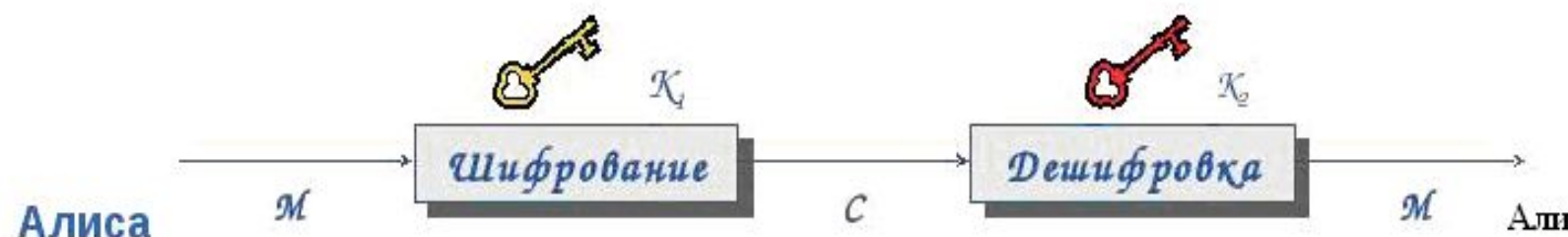
Три основных аспекта безопасности:

Privacy – Обмен сообщениями должен быть приватным.
(доступность передаваемых данных только участникам
диалога)

Integrity – Целостность данных, т.е. неизменность передаваемых
данных

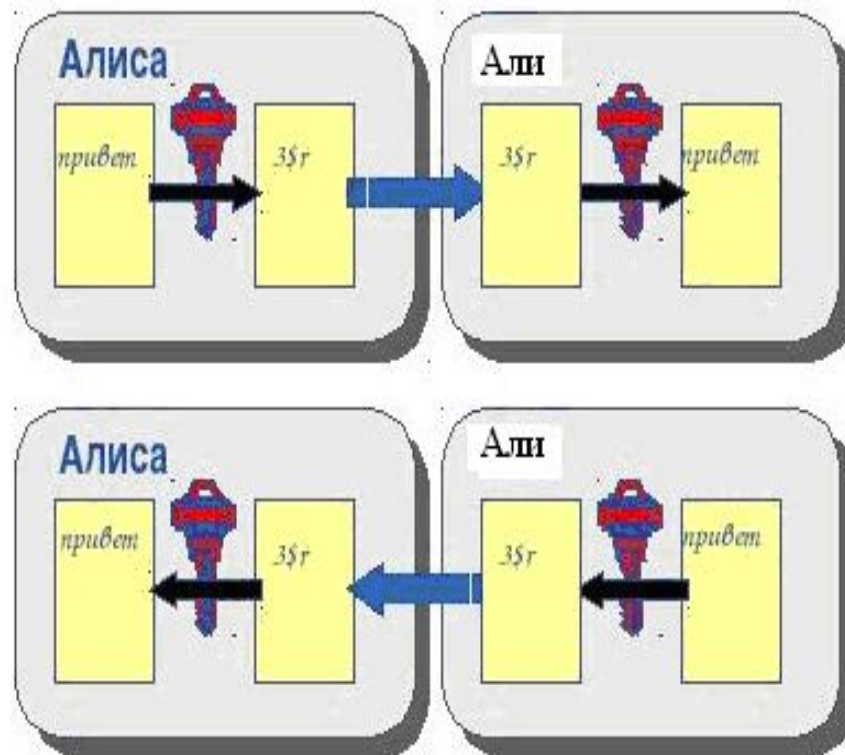
Authentication – Идентификация сторон, участвующих в диалоге
(проверка подлинности объекта)

Криптография – математическая дисциплина, которая занимается вопросами информационной безопасности и связанными с ней проблемами, особенно шифрованием, аутентификацией и контролем доступа

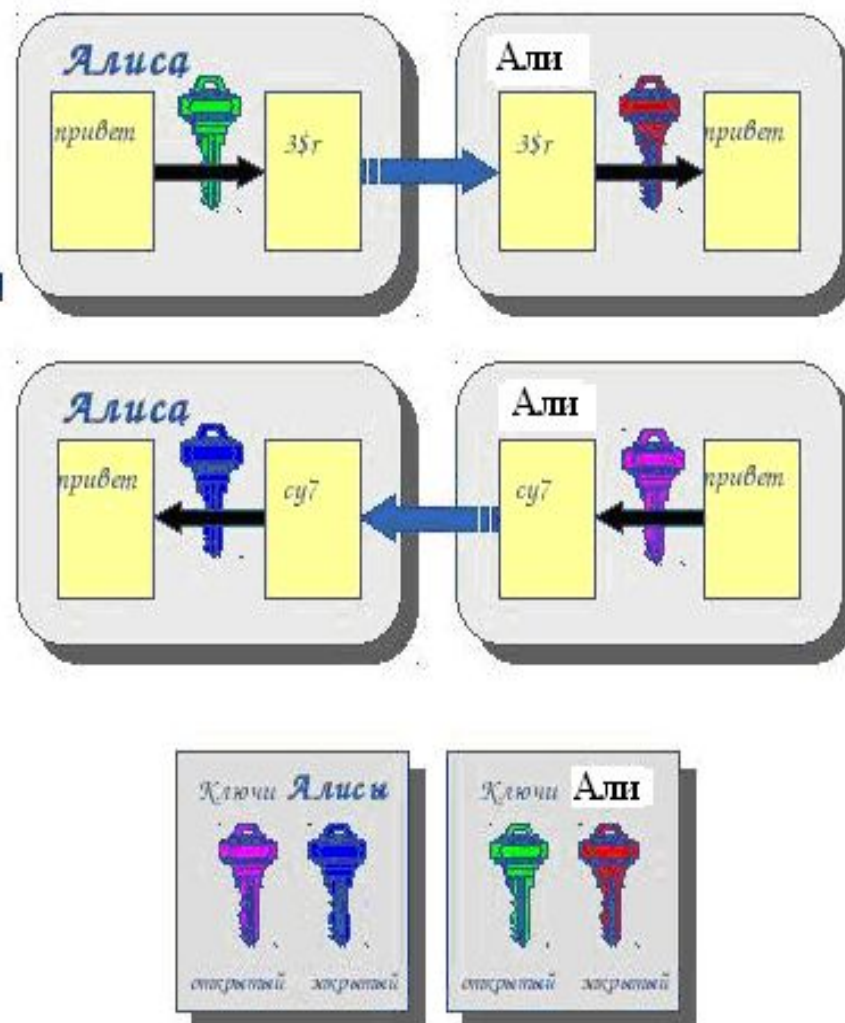


- Исходное сообщение: M
- Зашифрованное сообщение: C
- Шифрование с ключом K_1 : $E_{K_1}(M) = C$
- Дешифровка с ключом K_2 : $D_{K_2}(C) = M$
- Алгоритмы
 - Симметричный: $K_1 = K_2$
 - Несимметричный: $K_1 \neq K_2$

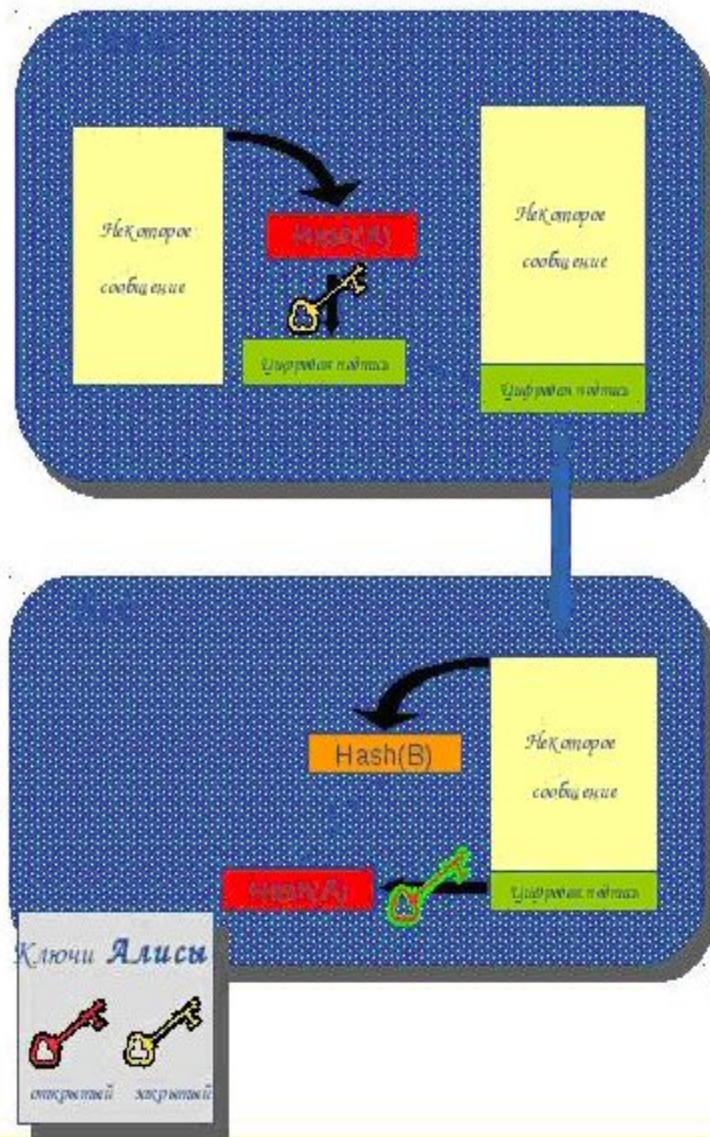
- Один и тот же ключ используется для шифрования и дешифровки
- Преимущества
 - Скорость
- Недостатки
 - Как безопасно передать ключ?
- Примеры
 - DES
 - 3DES
 - Rijndael (AES)
 - Blowfish
 - Kerberos



- У каждого пользователя 2 ключа: **открытый** и **закрытый**
 - “невозможно” вычислить значение закрытого ключа по открытому
 - сообщение, зашифрованное одним ключом может быть расшифровано *только* при помощи другого
- Нет необходимости обмениваться секретной информацией
 - отправитель зашифровывает при помощи *открытого* ключа получателя
 - получатель расшифровывает при помощи своего *закрытого* ключа
- Примеры
 - Diffie-Hellman (1977)
 - RSA (1978)



- Алиса вычисляет **дайджест** (hash) сообщения
- Алиса шифрует дайджест, используя свой **закрытый** ключ: зашифрованное значение и есть **цифровая подпись**
- Алиса отправляет подписанное сообщение
- Алис получает сообщение и вычисляет значение дайджеста
- Алис расшифровывает цифровую подпись при помощи **открытого** ключа Алисы и сравнивает его с вычисленным значением дайджеста
- Если оба значения равны, то сообщение не было изменено при передаче



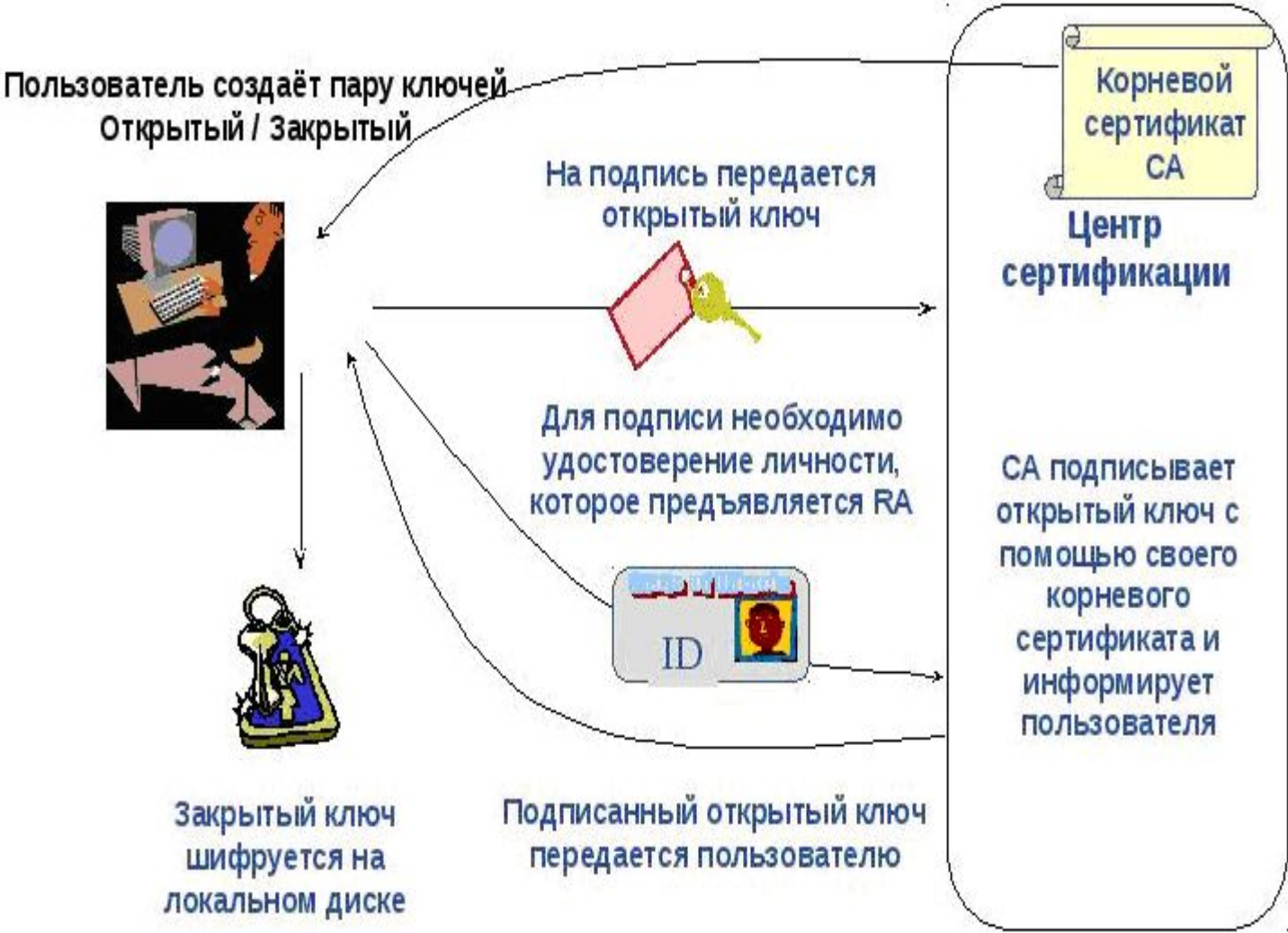
- Использование цифровой подписи Алисы безопасно, если:
 - Закрытый ключ Алисы остался секретным
 - Али знает её открытый ключ
- Но как Али может быть уверен, что открытый ключ, который он знает, на самом деле принадлежит Алисе, а не кому-то, кто выдаёт себя за неё?
 - Нужна некоторая третья сторона, которая будет гарантировать соответствие между открытым ключом и объектом, которому он принадлежит
 - Обе стороны, и Алиса и Али должны доверять этой третьей стороне

Эта третья сторона называется Сертификационный Центр - Certification Authority (CA).

- выдаёт цифровые сертификаты (содержат открытый ключ и идентификационную информацию) для пользователей, программ и машин (подписанные цифровой подписью CA)
- при этом проверяет соответствие представленных персональных данных и объекта
 - Но как это сделать, если сертификационный Центр в Москве, а пользователь – в Санкт-Петербурге?
 - Возникает сообщество Ответственных за Регистрацию

Registration Authority (RA)

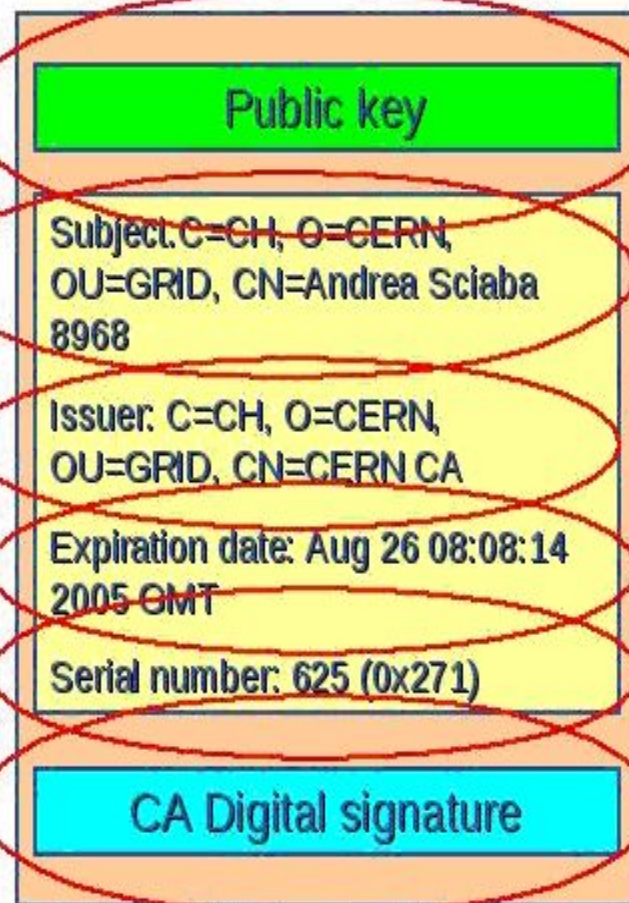
Получение сертификата



Х.509 сертификат содержит:

- открытый ключ владельца;
- данные владельца;
- информация о СА;
- срок действия;
- серийный номер;
- цифровая подпись СА

Структура сертификата Х.509



Али **Б**) хочет аутентифицировать Алису (**А**)

- **А** посылает свой сертификат **Б**
- Он проверяет правильность сертификата и подпись (имеет PK CA).
- **Б** посылает **А** произвольную фразу (challenge) с просьбой зашифровать её закрытым ключом **А**.
- **А** шифрует пришедшие данные
- **А** отсылает ответ (response) **Б**.
- **Б** расшифровывает ответ **А** с помощью переданного ранее открытого ключа
- **Б** сравнивает результат с эталонной фразой. Если сравнение успешно, то **А** действительно владеет закрытым ключом, соответствующим сертификату.



- В зависимости от способа получения сертификата он может быть получен в различных форматах:
 - *.pem формат: 2 файла: userkey.pem – закрытый ключ, usersert.pem – подписанный сертификат)
 - *.p12 формат (PKCS12): один файл - для загрузки в браузер Mozilla/Netscape/FireFox
 - *.pfx формат: один файл - для загрузки в браузер Internet Explorer
- Как правило, сертификат должен быть загружен в браузер (регистрация в ВО)
- Процедура экспорта/импорта зависит от типа используемого браузера и формата сертификата
- Сертификат имеет срок действия (от 2 недель до 1 года)
- По истечению срока действия он может быть продлён

Проблемы:

Single sign-on

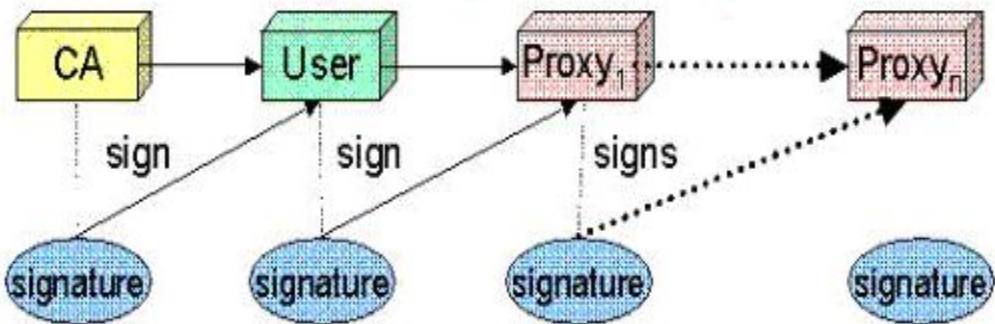
(однократное предъявление
первичного закрытого ключа)

Delegation

делегирование полномочий



Прoxy-сертификат
(расширение X.509)



Применение проxy-сертификата для аутентификации избавляет пользователя от необходимости вводить свой пароль при каждом взаимодействии с сервисами.

Можно передавать свои проxy-сертификаты другим субъектам для выполнения операций от своего имени.

Ограниченное время действия и ограниченное назначение

- Проху сертификат имеет достаточно короткое время жизни (обычно не более 24 часов). А как быть, если заданию требуется больше времени для выполнения?
 - в HEP Data Challenges в LCG некоторые задания выполнялись до 2 суток
- Выход – создание специального сервиса для автоматического обновления сертификатов (**MyProxy server**)
- Проху-сертификат можно зарегистрировать на сервере Мупроху и он будет обновляться в течение указанного периода времени (по умолчанию 7 суток)
- При этом соответствующий запрос будет проходить через Myproxy server

- «Динамическое собрание одиночек и организаций, гибко, безопасно и координировано разделяющее ресурсы»
- Пользователь Грид **обязан** принадлежать к одной из ВО
 - ВО согласовывают доступ к Грид-узлам и ресурсам
 - Авторизация проверяется на ресурсе
- **ВО с технической точки зрения:** ресурс, перечисляющий Distinguished Names сертификатов пользователей конкретной ВО
- Реализационно ВО ведёт список своих членов на специальном сервере (LDAP Server)
 - **этот список распространяется на все узлы, где поддерживается эта ВО**
 - **сопоставляется с локальными пользователями, зарегистрированными на этом узле (обычно выполняется через файл grid-mapfiles)**

"/C=CH/O=CERN/OU=GRID/CN=Simone Campana 7461" .dteam

"/C=CH/O=CERN/OU=GRID/CN=Andrea Sciaba 8968" .cms

Эволюция системы управления ВО

До VOMS

- Пользователь может быть членом только одной ВО
- Все члены ВО имеют одинаковые права
- Grid-mapfiles модифицируются только системой управления ВО
- `grid-proxy-init`

С VOMS

- Пользователь может быть членом нескольких ВО
 - Объединение прав
- ВО может иметь группы
 - Различные права для каждой
 - Различные группы экспериментаторов
 - Связанные группы
- ВО может иметь роли
 - Назначаются для особых целей
 - `Haap`, `sysadmin`
- При создании Proxy сертификата вводится дополнительный атрибут – имя ВО
- `voms-proxy-init -voms gilda`

VOMS – используется сейчас в Грид EGEE

- Аутентификация основывается на использовании сертификатов стандарта X.509
 - Устанавливаются отношения доверия между Certificate Authorities (CA) и узлами, между CAs и пользователями
 - CAs выдаёт/подписывает (долгоживущие) сертификаты, идентифицирующие узлы и пользователей (аналог паспорта)
 - Широко используется в браузерах для аутентификации сайтов
 - Для того, чтобы уменьшить уязвимость, в Грид для идентификации пользователей используются (короткоживущие) proxy их сертификатов
- Proxy сертификаты могут
 - Быть делегированы сервису для того чтобы он мог действовать от имени пользователя
 - Включать дополнительные атрибуты (например информацию о ВО для VOMS)
 - Быть зарегистрированными на внешнем хранилище (MyProxy)
 - Быть обновлены (в случае истечения срока действия)

• Аутентификация

- Пользователь получает сертификат от **Certificate Authorities (CA)**
- Соединяется с UI по SSH (UI – сервис пользовательского интерфейса)
- Загружает сертификат на UI
- “Входит” в Грид - создание проху
- GSI (Grid Security Infrastructure)

• Авторизация

- Пользователь вступает в VO
- VO согласовывает доступ к Грид-узлам и ресурсам
- Авторизация проверяется на ресурсе
- Права пользователя определяются информацией из его проху

