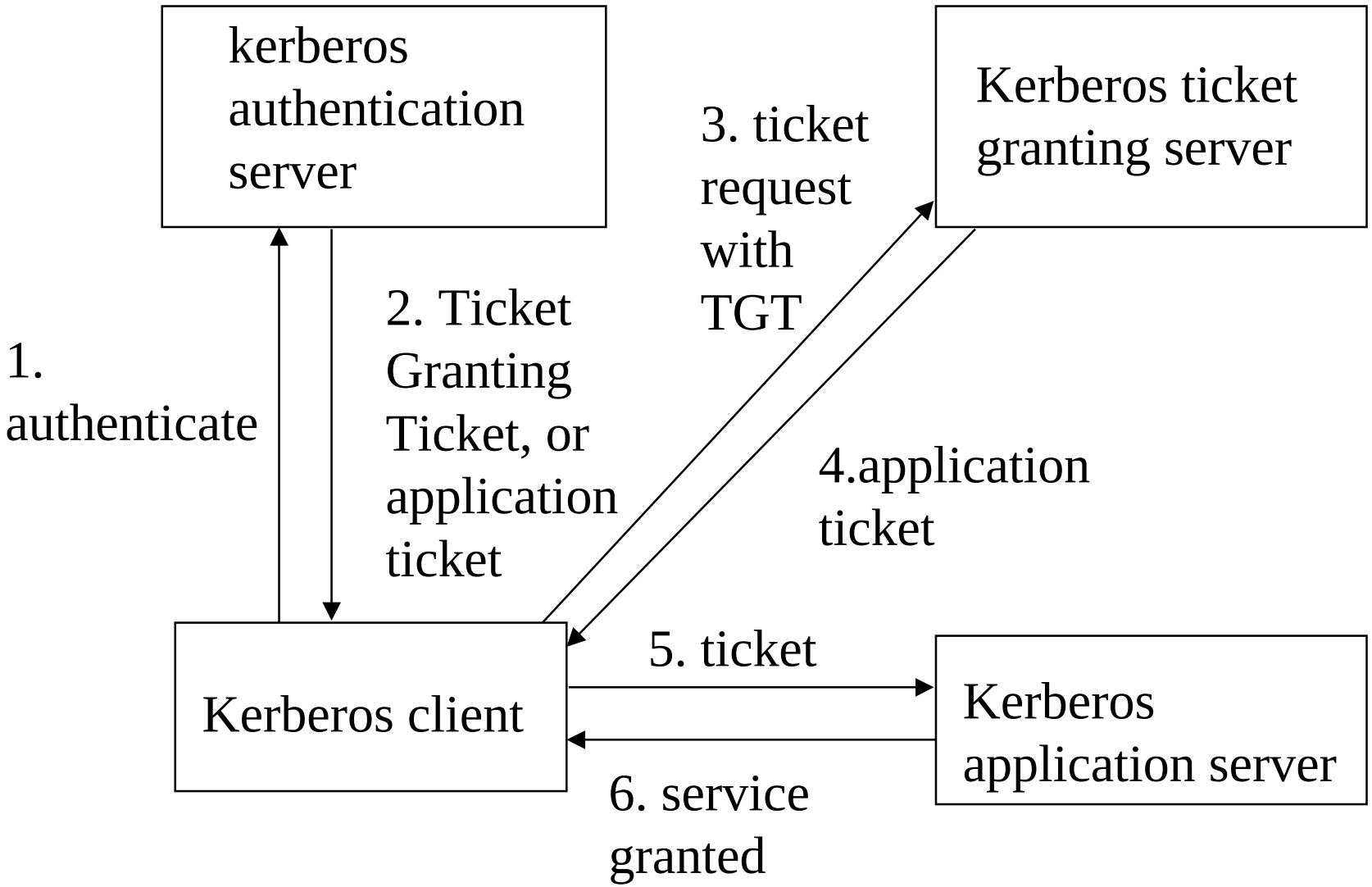
Kerberos

- Created in MIT Athena project 1988.
- Has been in wide use in the USA. It has been available for free to US and Canadian users, but under export restrictions because the cryptoalgorithms DES and RSA could not be exported.
- Export restrictions have been changed. However, for European users there has been an alternative, in fact improved, solution SESAME.
- Kerberos Version 4 had known limitations, the latest version of Kerberos is Kerberos V5 from about 1995 and some of the V4 problems are fixed. Some improvements are made, like use of ASN.1 for PDU definitions.
- Kerberos is mature technology.
- Kerberos is not very safe, but improves distributed Unix security considerably.
- Windows'2000 user authentication uses Kerberos.

Kerberos

- The problem Kerberos addresses is how to use computer resources over an insecure network.
- In Unix traditionally users log in with passwords which go in plain text, or user .rhosts to get access without password, or some resources, like printers, can be used without any authentication.
- These methods are insecure. Kerberos tries to solve the problem so that a user does not need to give a password for each server, but is once authenticated by one more reliable authentication server (AS). (single-sign-on)
- Then AS gives tickets for using the other servers.
- AS can give directly a ticket to an application server, but more usual solution is that AS gives a Ticket Granting Ticket (TGT) and another server, Ticket granting server gives a ticket to an application.

Kerberos

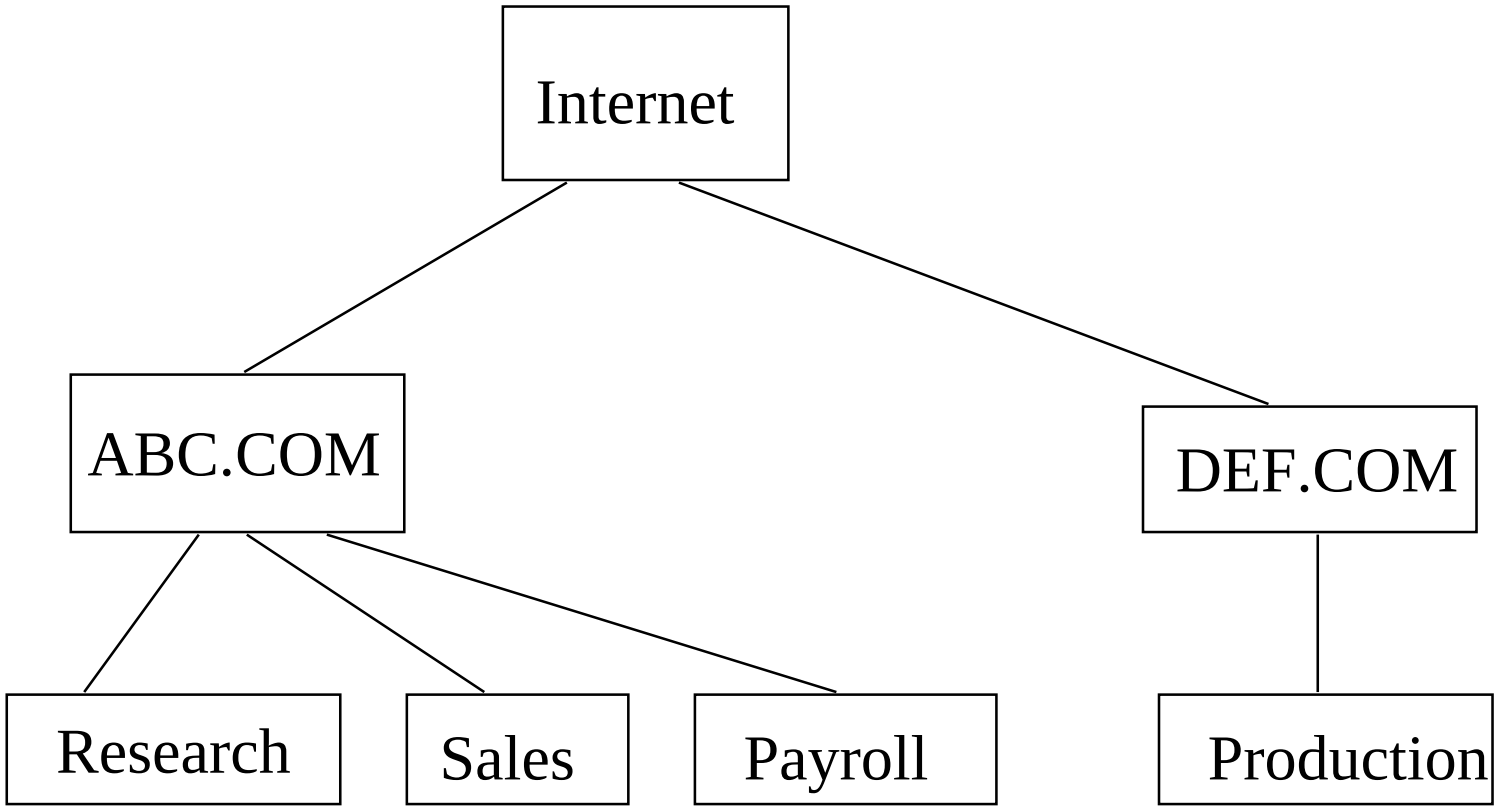

kerberos authentication server
Kerberos ticket granting server
3. ticket request with TGT
1. authenticate
2. Ticket Granting Ticket, or application ticket
4.application ticket
Kerberos client
5. ticket
Kerberos application server
6. service granted

Kerberos

- Kerberos has some limitations in design. In MIT the computer environment was such that users were using workstations.
- Workstations were used by only one user, so there was no danger of other users getting tickets from a authenticated user in a multiuser workstation.
- Dumb terminals were not used, they could not handle Kerberos message exchanges.
- Servers did not need to authenticate each other as there was not host-to-host communication. Kerberos is for human users to use servers.
- The computers run Unix and Kerberos is designed for Unix.
- SESAME is better designed for a heterogeneous computer environment, many other limitations are inherent in the basic Kerberos solution and are still in SESAME.

Kerberos

- For scalability reasons a network is divided into realms in Kerberos by a hierarchy. If Productions would like to use Payroll, it should be authenticated not only by DEF.COM Kerberos AS but also by ABC.COM Kerberos AS.



Kerberos

- **Kerberos roles:**
- Principal = Client - user’s workstation, no need to protect data on workstation disc when a user is not logged in.
- (Application) Server - a resource to be used, access to data and processing should be protected.
- KDC - Key Distribution Center
- AS - Kerberos Authentication Server. User uses login name and password to be authenticated by AS. AS gives a TGT (Ticket Granting Ticket)
- TGS - Kerberos Ticket Granting Server, when TGT is presented to TGS, TGS gives a ticket to an (application) server, then ticket is encrypted by the server’s key, so the server can authenticate the ticket..
- Kerberos Administration Server - adds new principals.

Kerberos

- **Obtaining TGT**
- A principal sends to AS the following message (in Kerberos V4, in Kerberos V5 the message is in ASN.1):

Field	contents	length
1	Protocol Version Number	1 byte
2	Message Type Identifier	1 byte
3	Username	string
4	Requested Ticket Instance	string
5	Kerberos Realm	string
6	Timestamp	4 bytes
7	Requested Ticket Lifetime	1 byte
8	Requested Service	string
9	Requested Service Instance	string

Kerberos

- TGT Request is in plain text. AS cannot authenticate the TGT request, an attacker can easily create a TGT request with anybody's name. AS sends the answer:

Field	contents	length
1	Session Key	8 byte
2	Service name	string
3	Instance	string
4	Realm, or domain	string
5	Ticket Lifetime	1 byte
6	Version Number	1 byte
7	Length of Encrypted Ticket Block	1 byte
8	Encrypted Ticket Block	(given by 7)
9	Timestamp	4 bytes

Kerberos

- TGT reply is encrypted by the principal's (clients) secret key, which the AS derives from the user password with a publicly known one-way function.
- In Kerberos the function is : calculate from the user password the DES key and encrypt TGT with DES.
- If the principal can open the TGT reply, he will get the TGT, which can be presented to TGS. It he cannot open it, he will get only encrypted data. If the user enters the correct password, the Kerberos software derives the correct DES key in the user's workstation and TGT reply can be opened.
- The TGT is encrypted with TGS key, which the AS also knows.
- TGT will grant tickets, which are encrypted with a secret key of the server. Then the server can check the tickets.
- We see that AS and TGS together know all secret keys.

Kerberos

- This is intentional, Kerberos only uses symmetric cryptoalgorithms.
- KDC (Key Distribution Center) is a generic name for a central point which knows all symmetric keys and distributes them. (KDC name is used only for symmetric cryptosystems). Kerberos needs KDC.
- We see one of the main security problems of Kerberos: an attacker can forge TGT requests from any users, possibly a large number of users, obtain encrypted TGT replies, and try password cracking on them.
- There is no salt used, so this is easier than cracking Unix password files. A password cracker tries dictionaries and transforms on them and if the attacker has obtained a large number of encrypted TGT Replies, he is almost sure to crack some.

Kerberos

- A real-life test is described in
- http://theory.stanford.edu/~tjw/krbpass.html
- TGT Replies were obtained for 25.000 Kerberos users and it took only nine seconds to crack the first password, in total 2045 passwords were cracked in two weeks.
- This was for Kerberos V4, Kerberos V5 has some improvements, but is in no way safe against password cracking.
- Users give poor passwords, no hope of fixing it. 283 of the cracked passwords were derivations of the login name or the full name. About half of the cracked passwords were words in some dictionary, the remaining hits were obtained with one transform to a password candidate. The best transform for the cracker was to convert all letters to lower case. With experience, password crackers can be hand tuned to be very good in cracking.

Kerberos

- Kerberos has an option to encrypt data (KRB_PRIV), but typically the purpose of encryptation in Kerberos is only to authenticate users to servers.
- For DES FIP 81 standard defines four modes: ECB(electronic code book), CBC(cipher block chaining), CFB (cipher feedback mode), OFB (output feedback mode). Kerberos V4 had created an own a variant of CBC where more than one block was xored to a block, but that had very bad error propagation properties. Kerberos V5 has the standard CBC, but with an addition - there is a confounder (random bits in the beginning) intended to make a selected plaintext attack more difficult.
- Kerberos supports several encryptation methods and a user can define his own encryptation method.

Kerberos

- Kerberos encryptation alternatives:
- **NULL** - no encryptation algorithms used
- **DES in CBC Mode with CRC-32 Checksum**
- In this method data is encrypted with DES in CBC. A Cyclic Redundancy Check (CRC) is calculated from the confounder and message sequence and placed in the checksum field. Notice, that CRC-32 is not collision free, so an attacker may find other messages which give the correct checksum.
- **DES with CBC Mode with an MD4 Checksum**
- **DES with CBC Mode with an MD5 Checksum**
- Similar, but Message Digest 4 or 5 is used instead of CRC. Message Digests are reasonably collision free.

Kerberos

- Kerberos has several alternatives for a checksum:
- CRC-32 - not recommended as it is not collision free
- RSA MD4 Checksum
- RSA MD4 Cryptographic Checksum using DES
- RSA MD5 Checksum
- RSA MD5 Cryptographic Checksum using DES
- DES Cipher Block Chained Checksum
- RSA MD4 Cryptographic Checksum Using DES Alternative
- DES Cipher-Block Chained Checksum Alternative
- What about these? It is possible to use many alternatives in Kerberos and though DES is not considered safe any more, there is no problem changing the algorithms to more safe ones. Same is true for SESAME, algorithms can be replaced.

Kerberos

- Kerberos is basically a protocol with a number of message exchanges. In Kerberos V5 all PDU structures are described in ASN.1. ASN.1 helps to improve software quality and PKI data structures are described in ASN.1 also.
- As an example, the ticket structure in V5 is probably as follows (Internet Security p. 513 has rather incomprehensible ASN.1, I try to fix it to make some sense).

  Ticket::= SEQUENCE {

tkt-vno	[0] INTEGER,
realm	[1] Realm,
sname	[2] PrincipalName,
enc-part	[3] IMPLICIT OCTET STRING }

- The value of enc-part is encrypted data structure of type EncryptdData. (use IMPLICIT tags, it gives shorter PDUs)

Kerberos

```
EncryptdData::= SEQUENCE {
        flags           [0] TicketFlags,
        key             [1] EncryptationKey,
        crealm          [2] Realm,
        cname           [3] PrincipalName,
        transited       [4] TransitedEncoding,
        authtime        [5] KerberosTime,
        starttime       [6] KerberosTime OPTIONAL,
        endtime         [7] KerberosTime,
        renew-till      [8] KerberosTime OPTIONAL,
        caddr           [9] HostAddress OPTIONAL,
        authorization-data [10] AuthorizationData
        OPTIONAL }
```

Kerberos

- **Timestamps**
- In the ticket to the server there are several timestamps among the fields.
- Timestamps try to protect against replay. Ticket lifetime is typically 5 min in Kerberos (it can be assigned), and this may protect against replay.
- However, there are short sessions, like reading email from a server, when an attacker might wait ready to replay the ticket and 5 min may be quite enough.
- **Key**
- The key in the ticket is the session key which is used to encrypt the ticket to the server. The session key is shared by the client (principal) and the server, and naturally also known to TGS.

Kerberos

- **Ticket flags**
- In the ticket there is a field “ticket flags”. The flags differentiate between types of tickets. There are the following ticket types:
- **Initial tickets**
- These are tickets directly issued by AS, some servers accept only them because they are given only when a password is given.
- **Pre-Authenticated tickets**
- The client may have authenticated the ticket with some valid method before an initial ticket is given, then pre-authenticated flag is set.
- **Invalid tickets**
- You may give an invalid ticket for instance to a batch process which will start at some scheduled time can be given a ticket which is not yet valid, it will get valid later. Other reasons exist.

Kerberos

- **(Potentially) Postdated tickets**
- If TGT has MAY-POSTDATE flag set, TGS may issue a postdated ticket. A postdated ticket becomes valid later.
- **Renewable tickets**
- You may want to use the same ticket later. Then it can be renewed by sending a request to TGS. There is the maximum renewable time.
- **Proxy tickets**
- There are proxies for ftp and other applications which can use Kerberos. However, here the idea is that the client asks a server to act on behalf of the client and then the server is given a proxy ticket. Proxy ticket has restrictions.
- **Forwarded tickets** Like proxy tickets, a bit less restricted.

Kerberos

- **Message exchanges in Kerberos:**
- **KRB_AS_REQ, KRB_AS_REP**
- Principal authenticates to AS and obtains TGT.
- **KRB_TGS_REQ/REP**
- Principal connects to TGS and obtains a ticket.
- **KRB_KDC_REQ/REP**
- The messages are identical to either KRB_AS_REQ/REP or KRB_TGS_REQ/REP, the principal requests TGT or a ticket from KDC.
- **KBR_ERROR**
- A generic error message, which can come from any message exchange.

Kerberos

- **KRB_AP_REQ/REP**
- Message exchange between a principal and a server. A client sends a ticket to the server and the server grants usage.
- **KRB_SAFE**
- A client needs message integrity, a collision free checksum is used.
- **KRB_PRIV**
- A client needs privacy, messages are encrypted.
- **KRB_CRED**
- A client needs to send Kerberos credentials from one host to another may use this service.

Kerberos

- How to use Kerberos (from the WWW-site: www.isi.edu/gost/brian/security/kerberos.html)
- For a client Kerberos is rather transparent.
- You need to get TGT first. The command is
- **kinit**
- Give password for your_name@YOUR_REALM
- You can verify that you have got TGT by making
- **klist**
- Ticket cache: /var/tmp/krb5cc_1234
- Default principal: your_name@YOUR_REALM
- Valid starting Expires Service principal
- 24-Jul-95 12:58:02 24-Jul-95 20:58:15 krbtgt/realm

Kerberos

- Then you simply use normal Unix commands, like
- **rlogin newhost.domain**
- You can see that kerberos is working only by looking at the cache by klist.
- When you leave you would like the credentials in the cache to be destroyed, so use **kdestroy**
- For the systems administrator configuring Kerberos is more complicated. You must configure AS and TGS. You must register principals. You should make available services which use Kerberos. All of this is straightforward by following instructions. In general, there are many Kerberos products and different products do not necessarily work together.
- In Europe, you might be configuring SESAME, it has more components, notably the PKI infrastructure.

Kerberos

- Security threats in Kerberos.
- Two main threats have been mentioned:
- It is possible to crack passwords
- It is possible to replay tickets is time service is messed up. (If time, like NTP, cannot be spoofed, it must be a secure environment, which needs no Kerberos in the first place).
- There are more: if anybody hacks into AS or TGS they get all secret keys. If anybody hacks into a server he will get the protected data. Anybody can play a man in the middle between a client and a server, the data is in any case not protected.
- If many users use one workstation, the tickets are in memory. They used to be in the disc, anyway, they are available. You can capture a user password with a login trapdoor.
- In short, it is not especially secure but better than nothing.

Alternatives to Kerberos

- **Bones**
- As Kerberos included DES and DES could not be exported, a version Bones of Kerberos was made. There the places where encryption was made are carefully removed and not marked so that insertion of cryptosoftware would not be easy.
- What for Bones, it does nothing useful? Many products expect that Kerberos is running and work only if there is Kerberos, If Bones is running, they work thinking Kerberos is running.Then these products could be exported.
- **E-Bones**
- Somebody put the encryption back to Bones and called the result Encrypted Bones.
- **DCE** (Distributed Computing Environment)
- DCE of OSF actually uses Kerberos, but it is not working with MIT Kerberos, so let us call it DCE Kerberos.

Alternatives to Kerberos

- **SESAME** (Secure European System for Applications in a Multi-vendor Environment)
- Created by ECMA (European Computer Manufacturer's Association)
- Implementation created in EU RACE project SESAM 1995.
- SESAME is similar to Kerberos and worked out from Kerberos ideas. Kerberos V5 has included some influences from SESAME and they interwork to some extent.
- SESAME is not only Kerberos.
- The most significant change is that there is added public key cryptography. There is Public Key Infrastructure with certificates and Certificate Authorities. Kerberos is based on symmetric cryptography.
- SESAME brought ASN.1 coding, now it is in Kerberos V5.

Alternatives to Kerberos: SESAME

- A design goal in SESAME is multi-vendor heterogeneous environment, not only Unix environment of a particular type.
- This goal is expressed by using ISO and ECMA standards.
- A significant design choice coming from multi-vendor environment is a generic access control scheme called PAC (Privilege Attribute Certificate).
- SESAME code originally contained parts from Kerberos, but as US export license could not be obtained, these parts were rewritten SESAME V4.
- SESAME does not seem to have wide usage, code is available. EU projects quite often just finish and stop.
- There are products, like ICL Access Manager and Bull SA's Integrated System Management AccessManager, using SESAME.

Alternatives to Kerberos: SESAME

- SESAME is an Open Systems solution following international standards. Actually it is also rather OSI using ASN.1, OSI Security Framework (ISO 10181) and the Directory Authentication Framework Standard (& X.509).
- This can be negative as OSI is not in fashion.
- SESAME uses the Generic Security Service API (GSS-API), which has some other usage in security applications.
- SESAME has developed SMIB Secure Management Information Base.
- What ever one says about SESAME, it has quite significant improvements to Kerberos. It uses better software development methods with more standard (or in other way in commonly accepted ways defined) protocols and APIs, it uses public key cryptography and in general, seems less ad hoc than the original Kerberos (which created the main ideas, though).

Kerberos: Summary

- Kerberos is widely used and a rather stable mature technology, maybe to some extent outdated also.
- The history of Kerberos shows a common trend in the Internet. An innovative project creates a system, it becomes popular and almost standard, later some faults and limitations are found and the solution is patched.
- The Kerberos solution decided to use symmetric cryptography because of performance problems in public key cryptography back in -88, these reasons have lost strength.
- The solution contained security faults, like vulnerability to clock modifications and most importantly password cracking, which was a well-known threat already 1979.
- There were also limitations in the solution for more general computing environments. Kerberos is for a Unix system with particular use of workstations as in MIT.

Kerberos: Summary

- Kerberos hardly would have become an international standard in this limited generality and unchecked security.
- As freeware software with good advertisement, it become quite popular. Only then it started to attract enough attention by analysts that the problems become apparent.
- Compare this to the AES (Advanced Encryptation Standard) creation process, the solution candidates were evaluated before the standard was chosen and before large investments.
- Efforts to improve Kerberos were made, like SESAM. The improvements were included in Kerberos V5. One of the improvements was use of ASN.1 which is not so important as such but points out to the usefulness of precise definition and automatic code generation in improving security. Another was using more standard security mechanisms.
- The result still has limitations inherited from the original model.