


6G7Z1009: Introduction to Computer Forensics and Security

Key management - II

Kerberos


Outline

- What is Kerberos?
- Why Kerberos?
- How Does Kerberos Work?
- Formal Description of Kerberos
- Kerberos Drawbacks
- Kerberos Realm
- Kerberos v4. Vs v5
- Kerberos Pro. vs Cons.


What is Kerberos ? -I

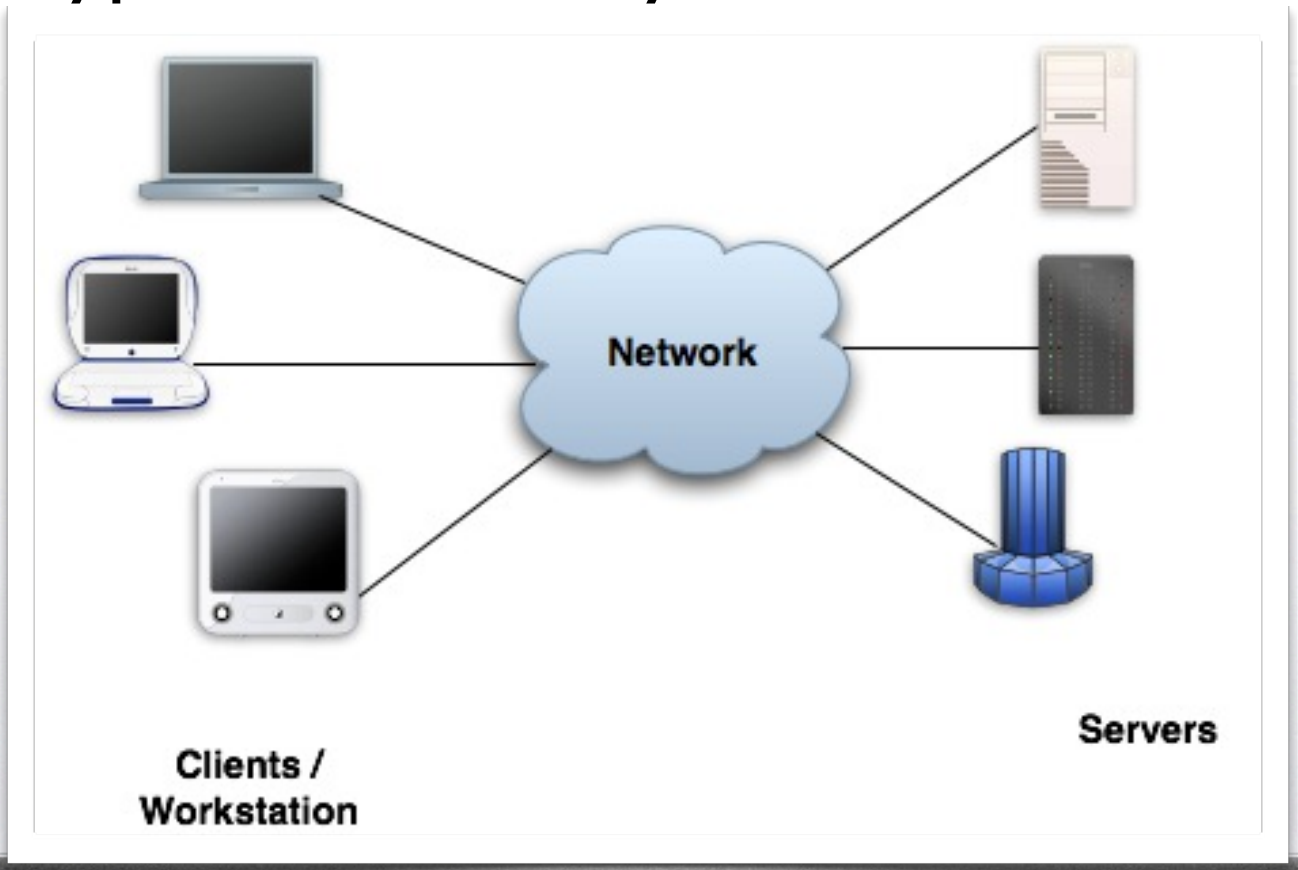
- Kerberos: in greek mythology, a many headed dog, the guardian of the entrance of Hades
- In computer security, Kerberos is a cryptographic based network authentication protocol
  - Developed by MIT for Athena Project (http://web.mit.edu/kerberos/www)
  - Uses symmetric cryptography and on-line authentication servers
  - Provides a centralised authentication server to authenticate users to servers and servers to users in a secure manner


What is Kerberos ? - II

- Provides strong security on physically insecure network
- Provides single-sign-on capability: only one password to remember and only need to enter it once per day
- Two versions: version 4 and 5
- Version 4 makes use of DES
- Kerberos is a secret key (private key) based service for providing authentication in a network
- It is based on work by Needham and Schroeder


Why Kerberos? - I

- A typical network system




Why Kerberos? - II

- The traditional authentication threats
  - A user pretends to be another user
  - Line eavesdropping on exchanges and replay attacks
- By using Kerberos,
  - To access all resources from anywhere on the network
  - Not need to enter a password to authenticate for each access to a network service


How Does Kerberos work? - I

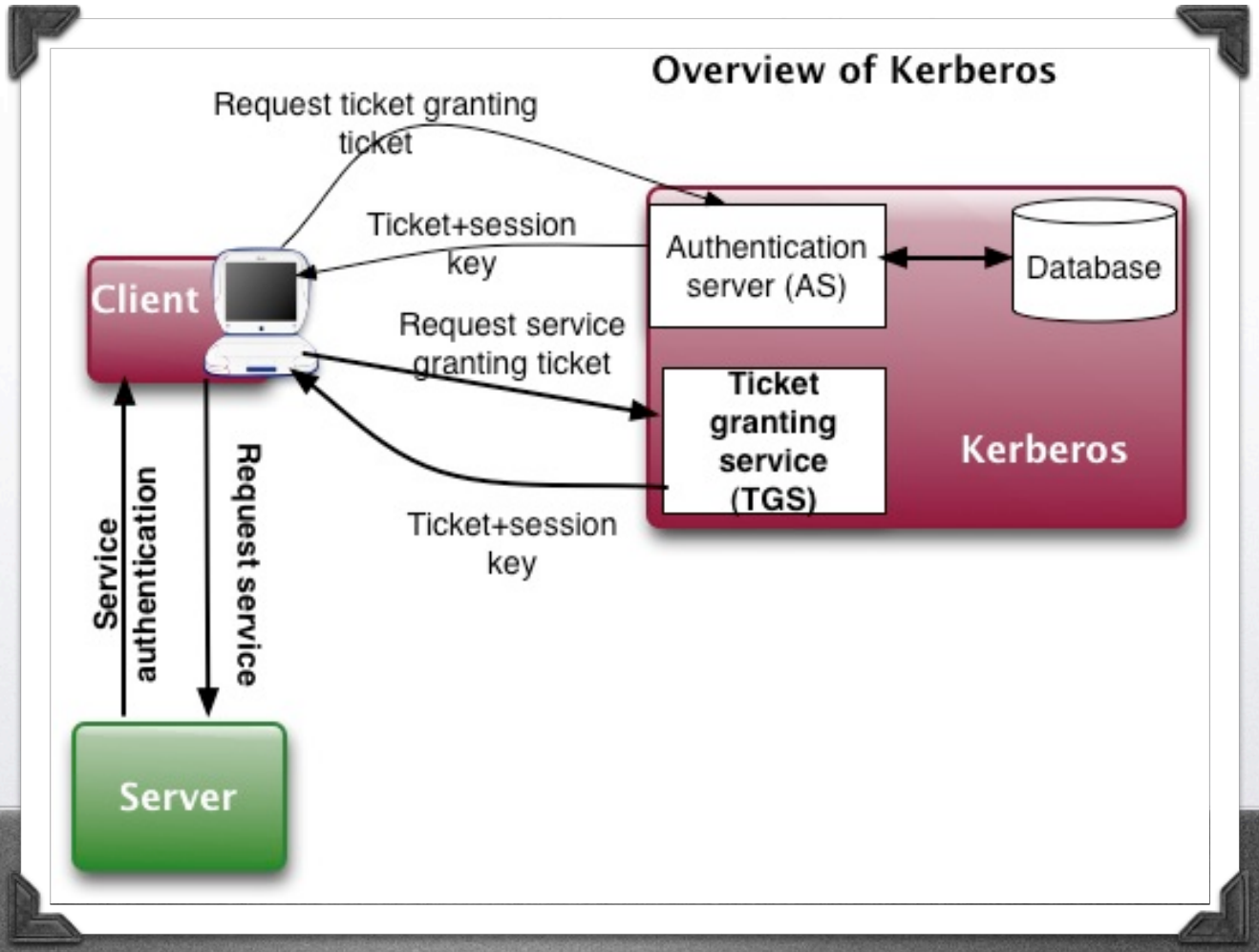
- Uses a trusted third party, Key Distribution Center (KDC) containing:
  - Authentication server (AS) and Ticket Granting Service (TGS)
- AS is responsible for handling a login request from a user. The AS maintains a database of secret keys:
  - Each entity (a client or a server) shares a secret key known only to itself and to the AS. Knowledge of this key serves to prove the entity's identity


How Does Kerberos

- TGS sets up secure channels. TGS hands out special messages. Known as tickets that are used to convince a server that the client is really who he claims to be.
- A ticket is an unforgeable, non-replayable, authenticated object. It is an encrypted data structure naming a user and a service that the user is allowed to obtain. It also contain a time value and some control information.

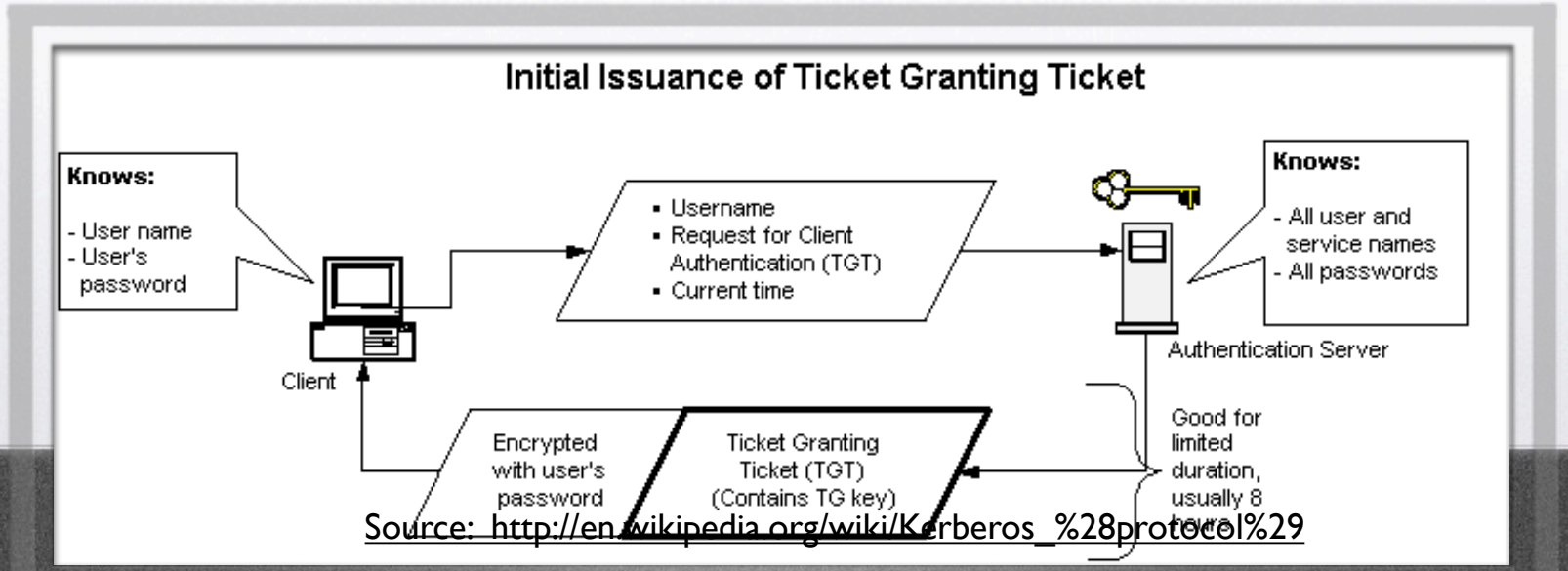

How Does Kerberos

- Overview of Kerberos



How Does Kerberos work? - III

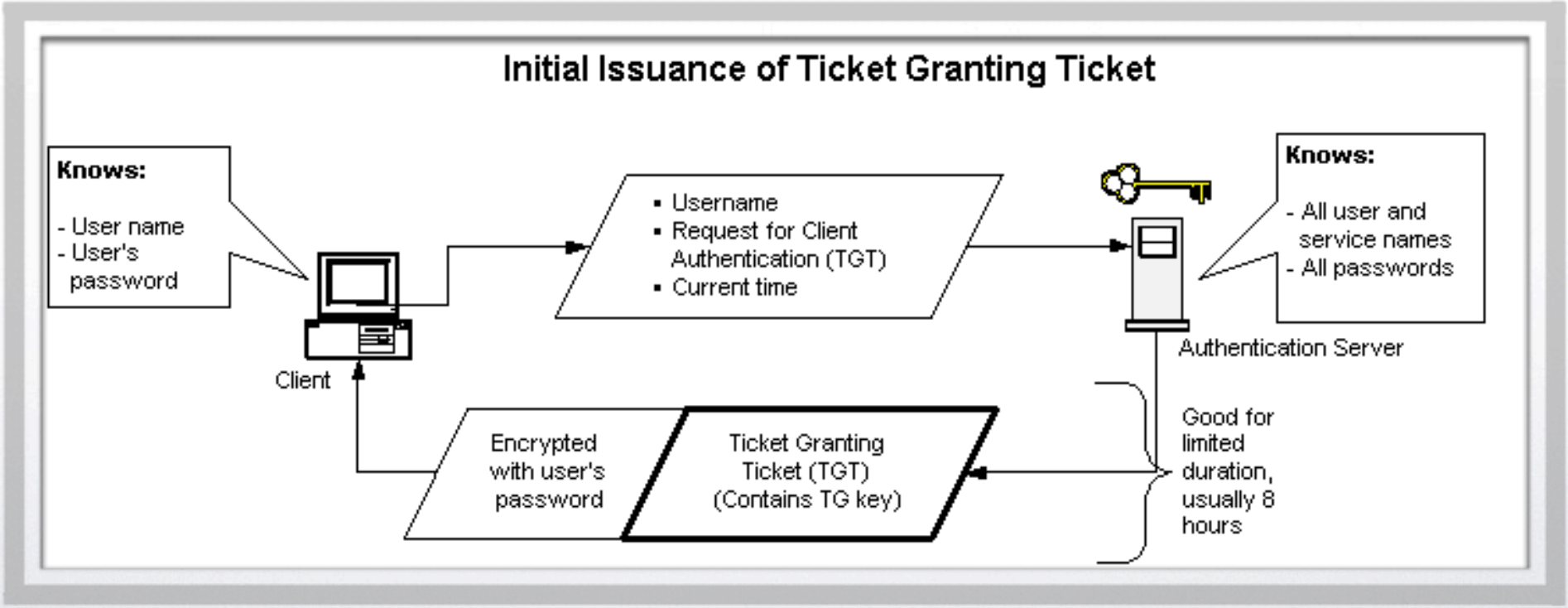
- The Operational steps:
  - User client logon: a user enters password and user name on the client and performs a one-way function and generate the secret key of the client (transforms a password into the key)



Source: http://en.wikipedia.org/wiki/Kerberos_%28protocol%29

How Does Kerberos work? - III

- The Operational steps:



Source: http://en.wikipedia.org/wiki/Kerberos_%28protocol%29


How Does Kerberos work? - III

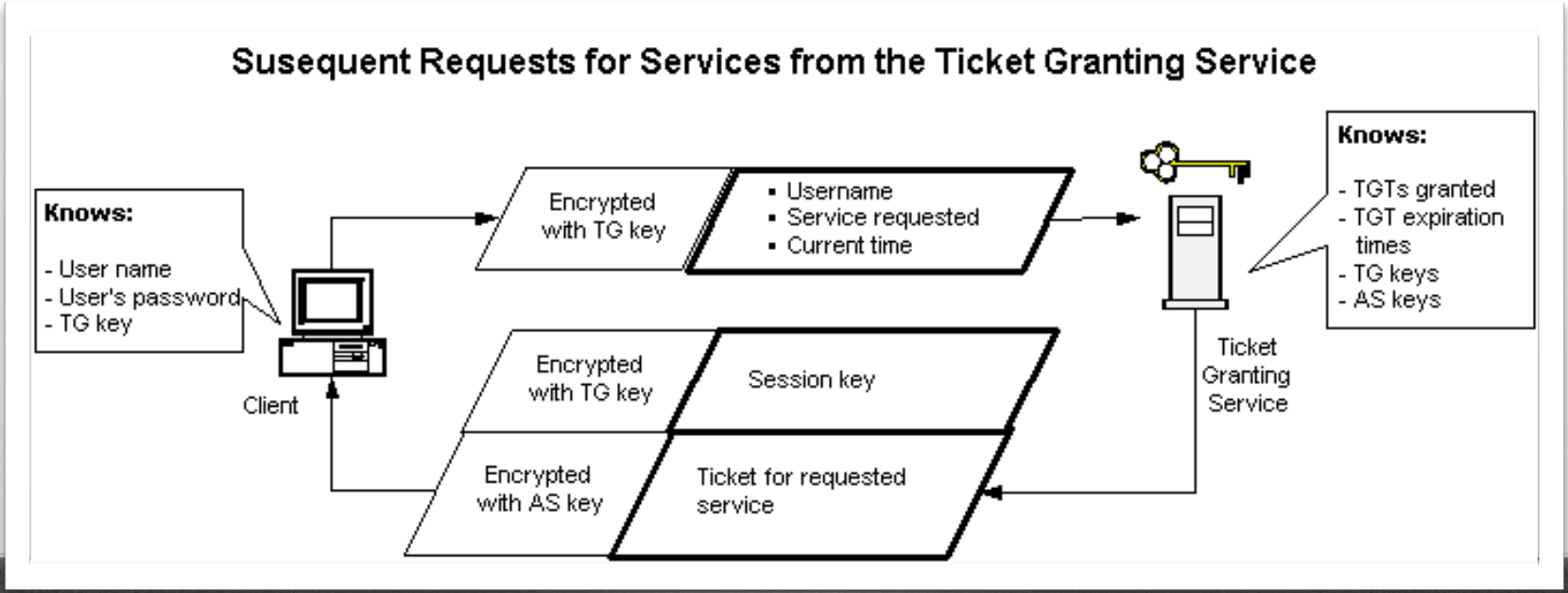
- The Operational steps:
  - Client to authentication server : the client send a cleartext message to the AS requesting services on behalf of the user (the secret key and the password are never sent to the AS). The AS generates the secret key through hashing the password of the user found at database.
  - AS checks to see if the client is in its database. If it is, the AS sends back the two messages to the client: Message A (client/TGS session key encrypted using the secret key of the client) and Message B(Ticket-Granting-Ticket encrypted using the secret key of the TGS)


How Does Kerberos work? - III

- The Operational steps:
  - Once client receives these two messages, it will decrypt with the secret key generated from the password entered by the user to obtain the TGS session key which is used for further communications with TGS (The client can not decrypt the Message B which is encrypted using TGS's secret key).
  - Now the client authenticates him/herself to the authentication server and receives a ticket. All tickets are time-stamped

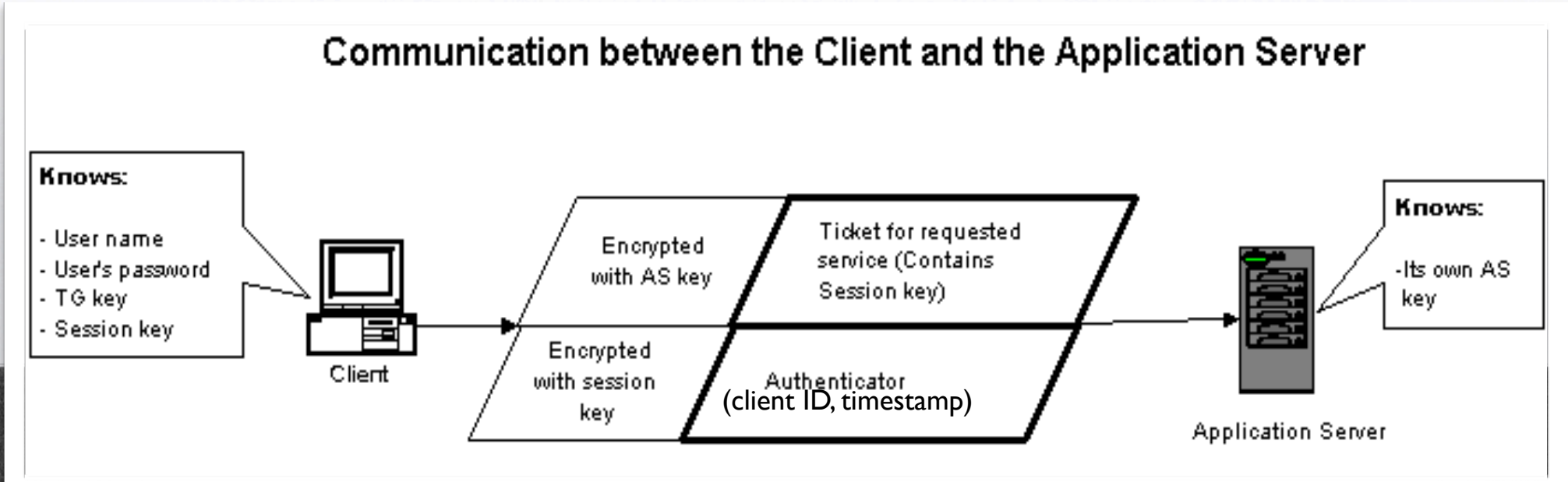
How Does Kerberos work? - IV

- The Operational steps:
  - Client to service authorisation: It then contacts the ticket granting server and sends messages to the TGS and obtain the ticket for requested service



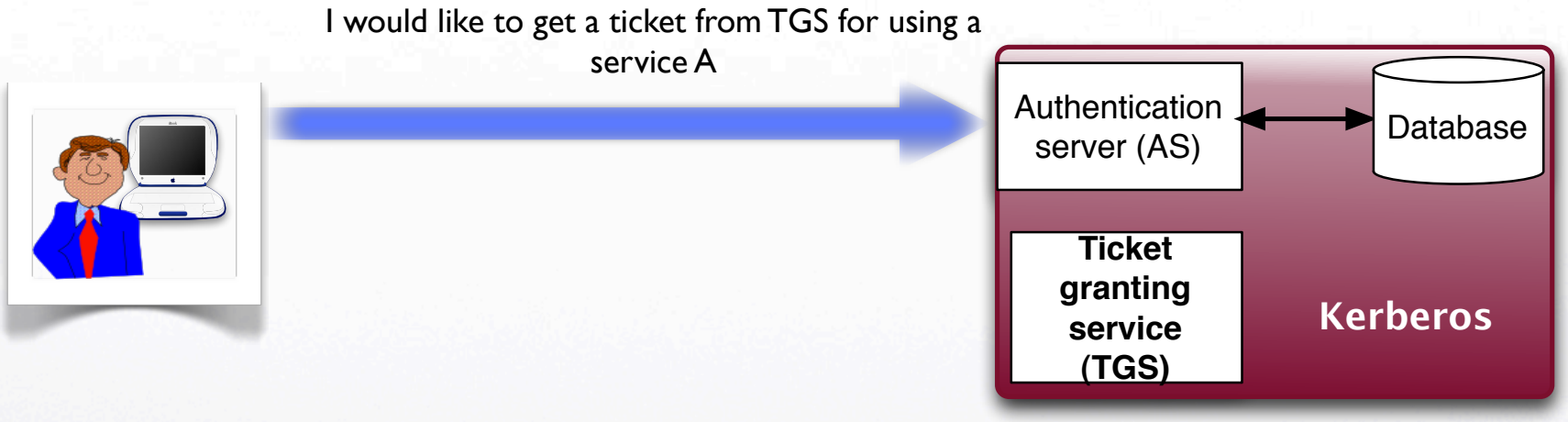
How Does Kerberos work? - V

- The Operational steps:
  - The client contacts the service server where the services are hosted and uses the ticket to demonstrate he/or she can use services on



How Does Kerberos work? - VI

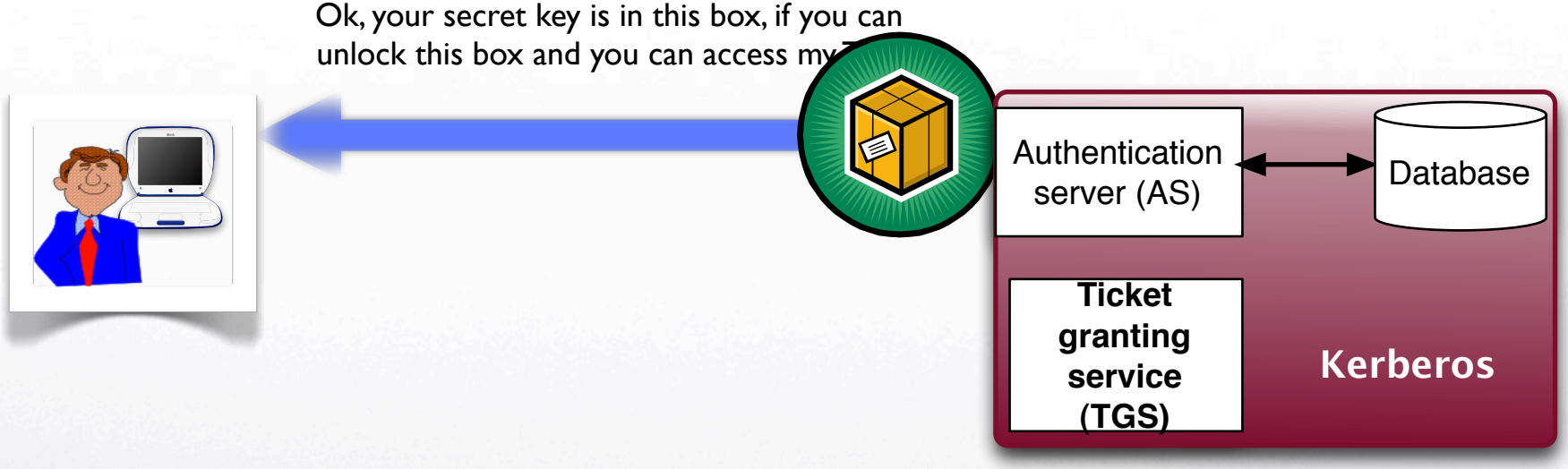
- The concrete operational scenarios:

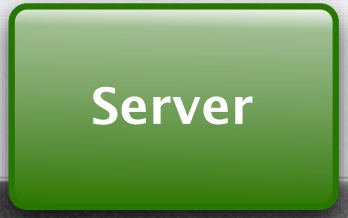


Services, e.g., FTP server, Web server

How Does Kerberos work? - VII

- The concrete operational scenarios:

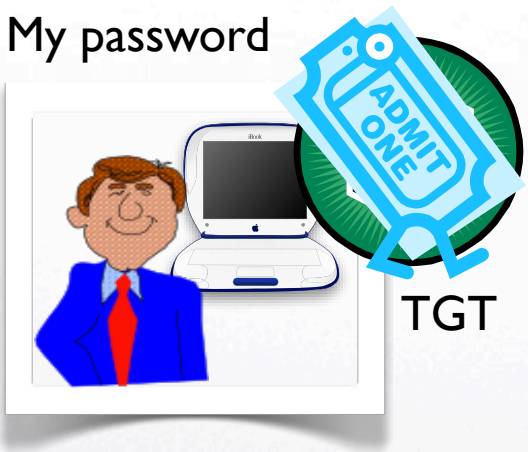


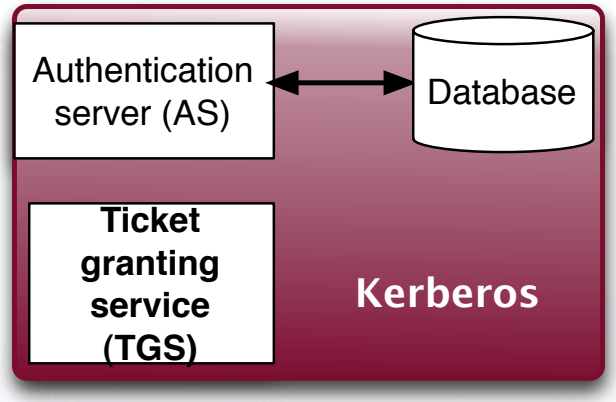


Services, e.g., FTP server, Web server

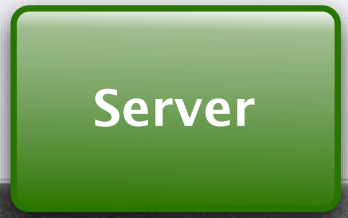
How Does Kerberos work? - VIII

- The concrete operational scenarios:





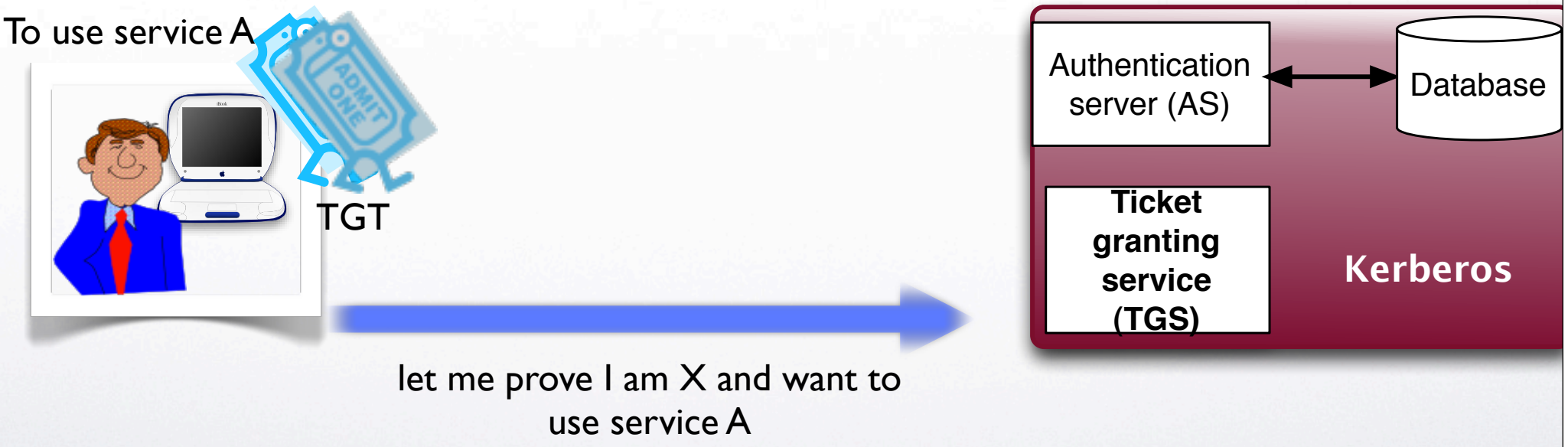
* The client now open the box using his secret key and now has TGT
* TGT must be presented to TGS to acquire a service tickets for using a certain service A. TGT has no password



Services, e.g., FTP server, Web server

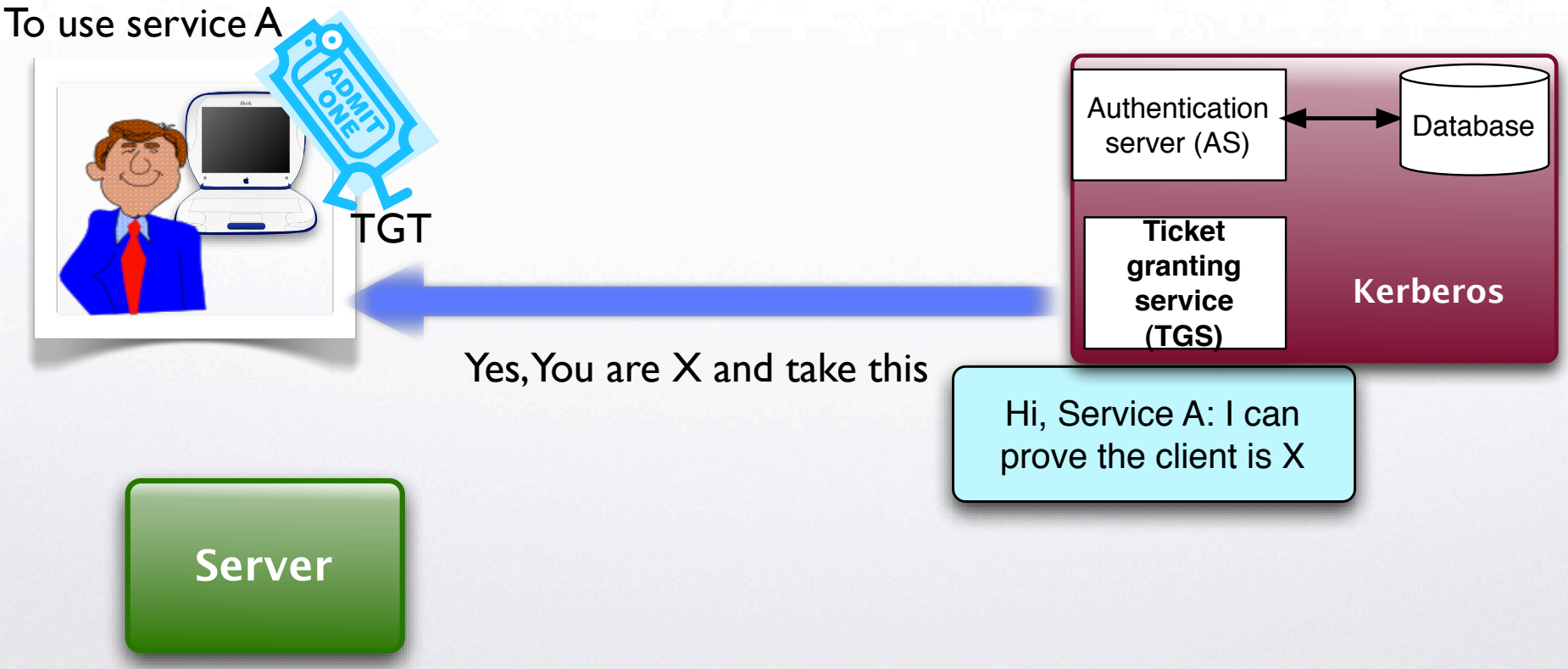
How Does Kerberos work? - IX

- The concrete operational scenarios:



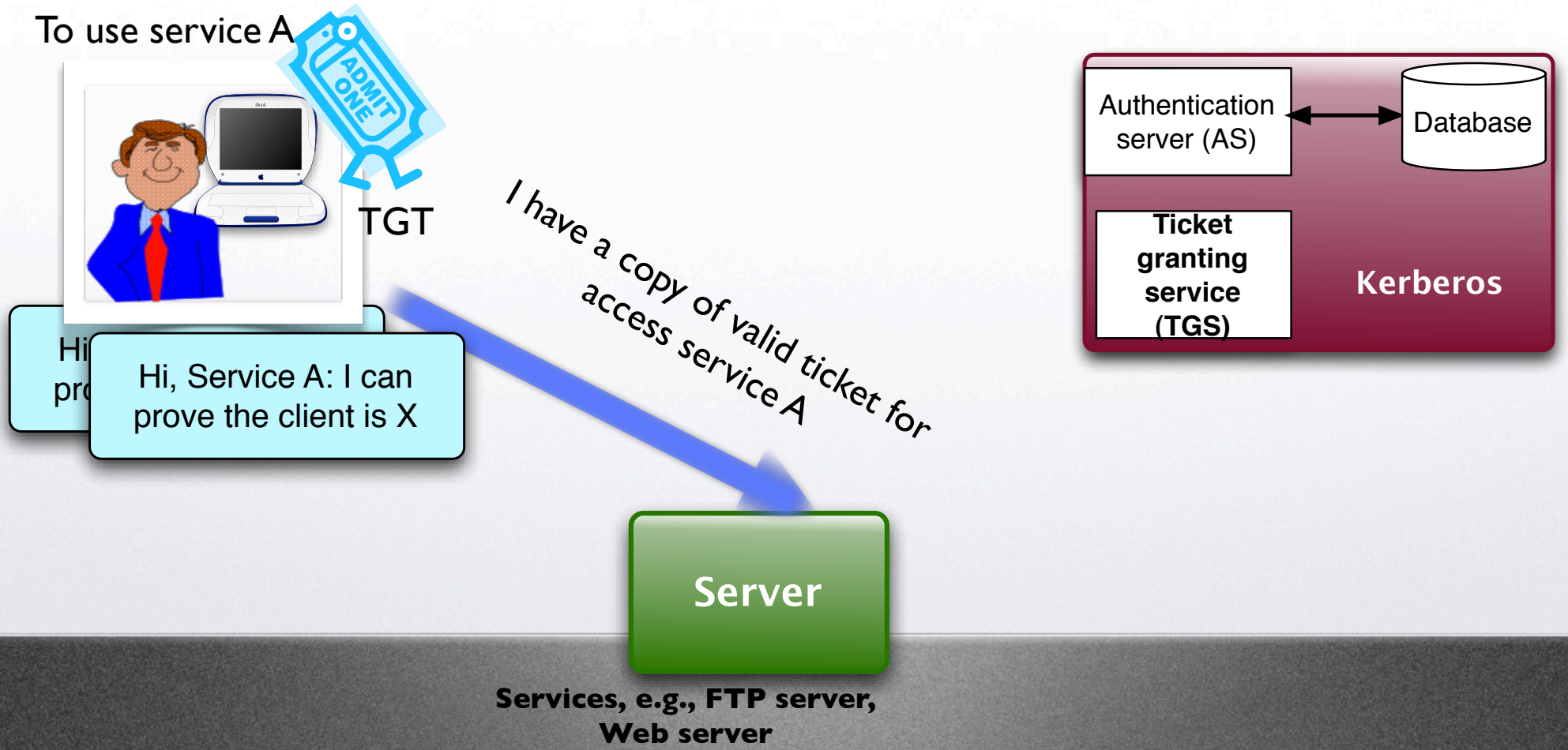
How Does Kerberos work? - X

- The concrete operational scenarios:



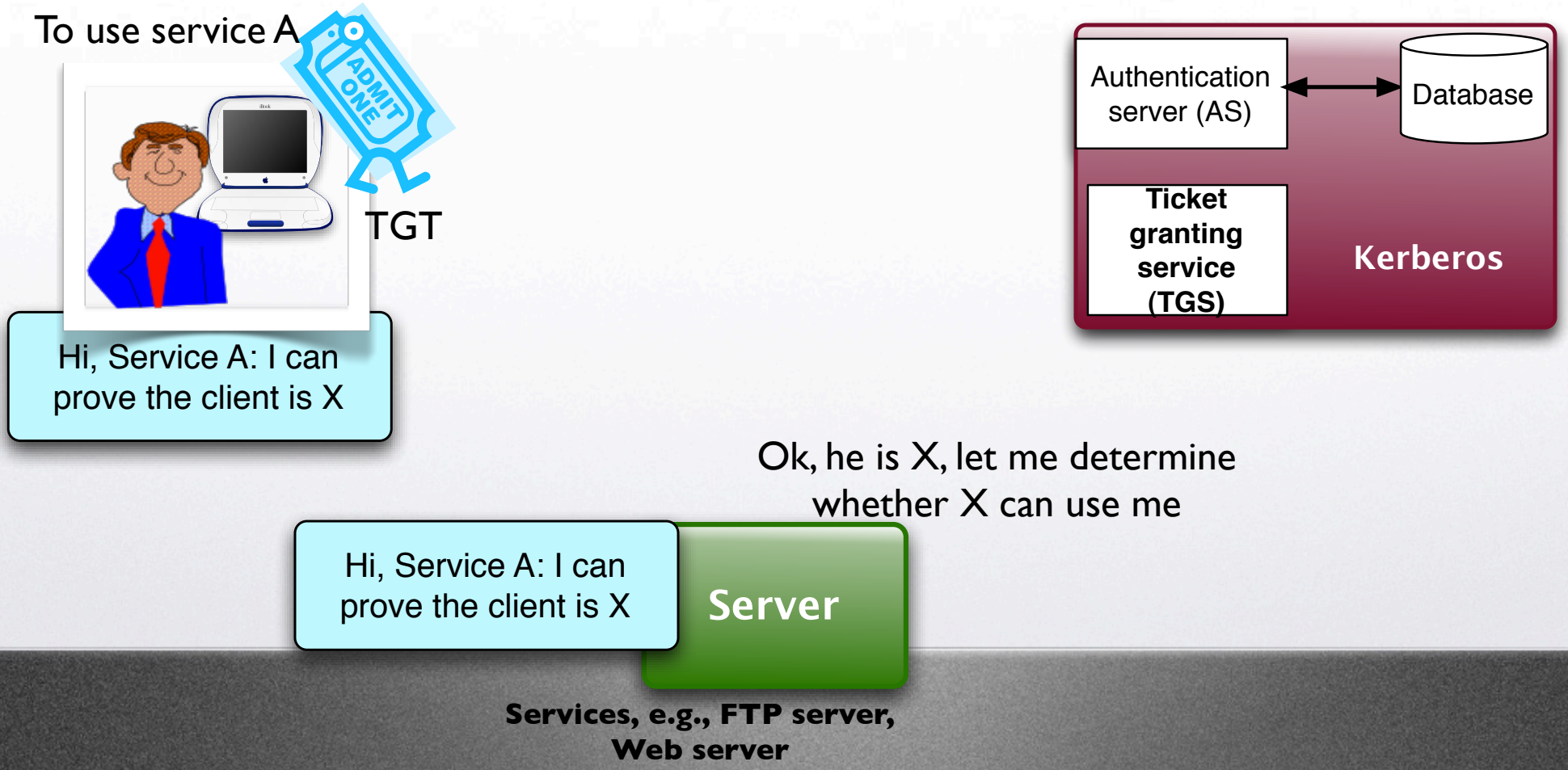
How Does Kerberos work? - XI

- The concrete operational scenarios:



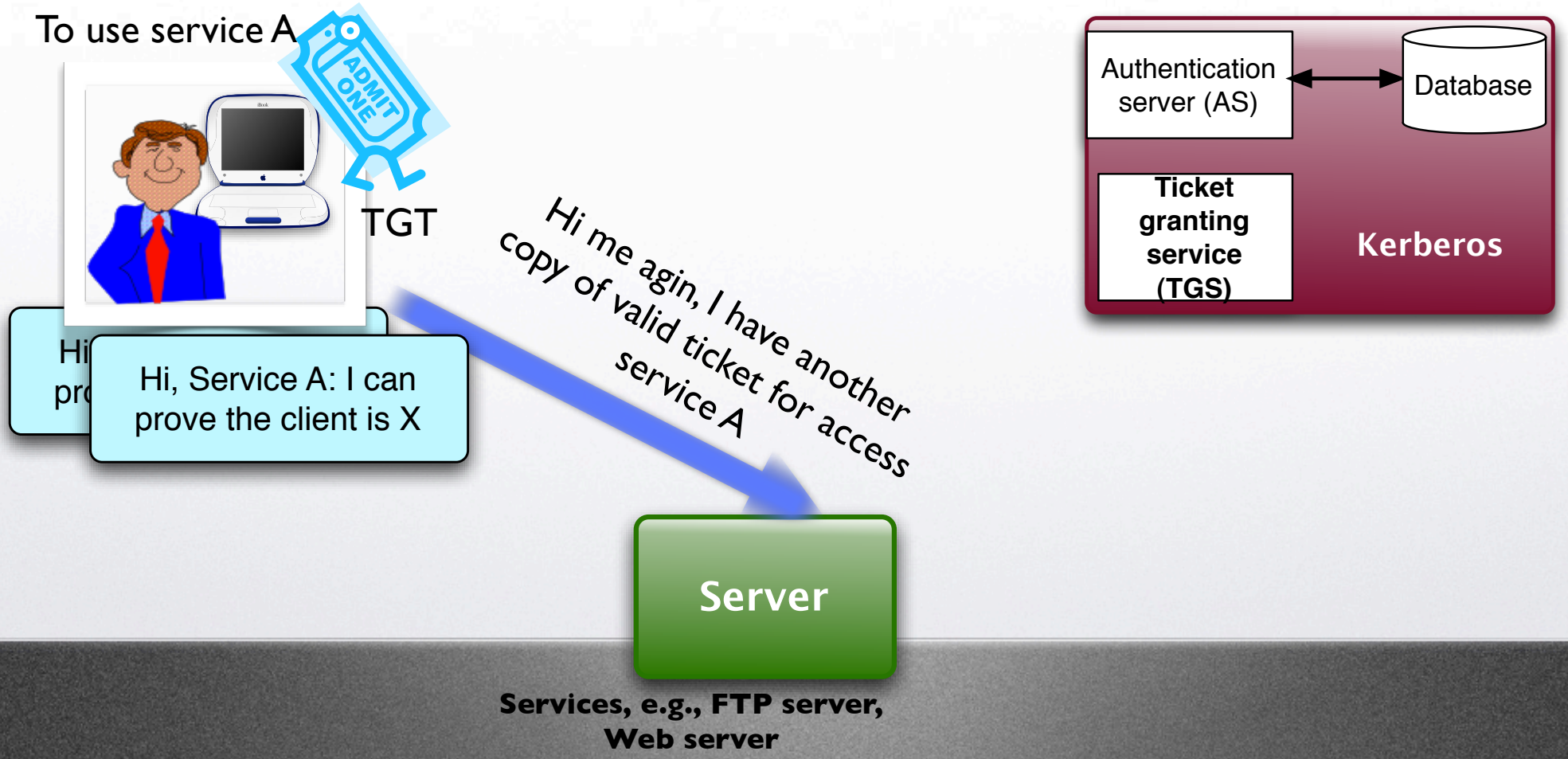
How Does Kerberos work? - XII

- The concrete operational scenarios:



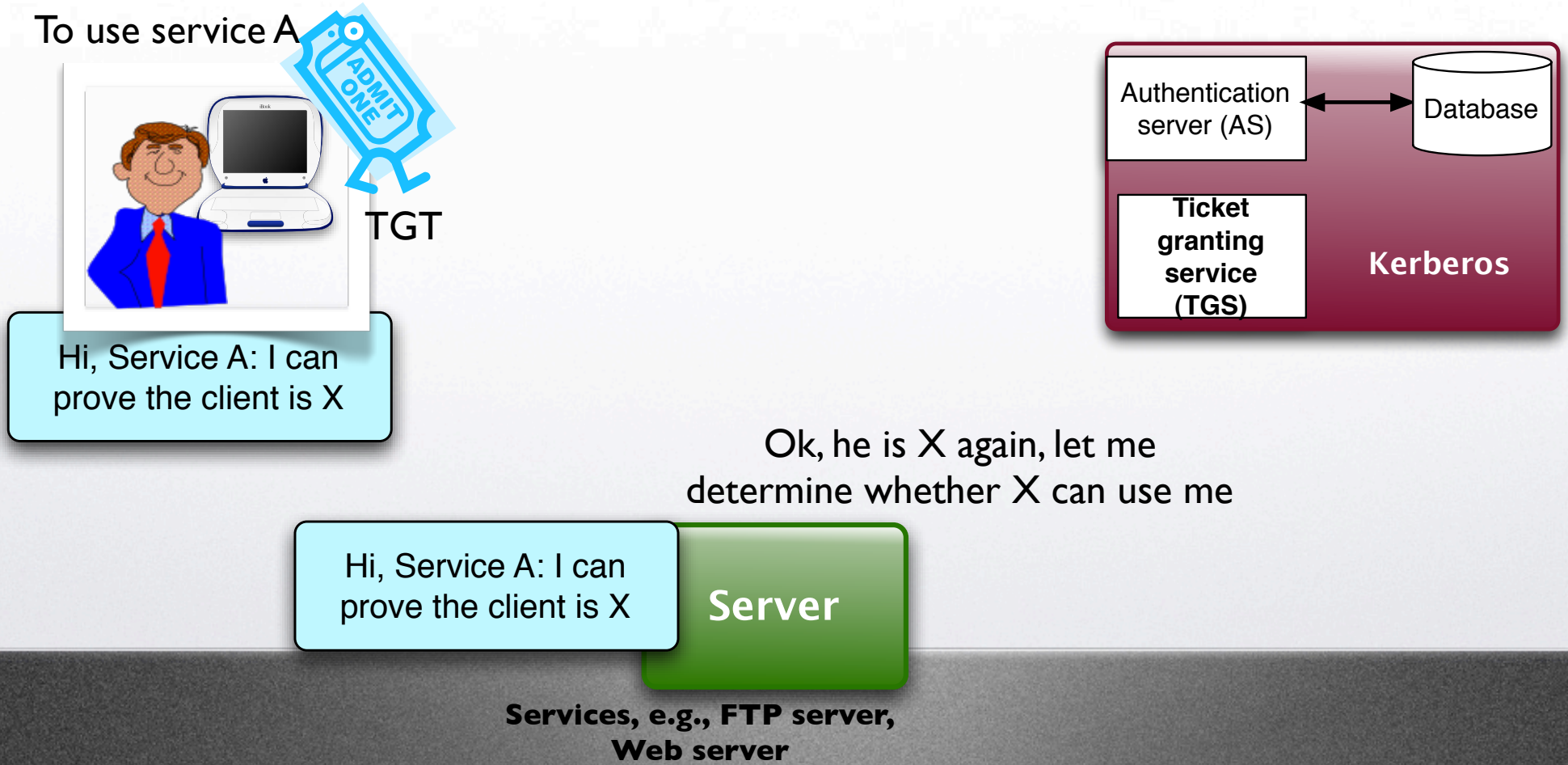
How Does Kerberos work? - XIII

- The concrete operational scenarios:



How Does Kerberos work? - XIV

- The concrete operational scenarios:




How Does Kerberos work?-XV

- The Secret key cryptography: DES in Kerberos v4
- Ticket
  - Each request for a service from a client requires a ticket
  - A ticket provides a single client with access to a single server
  - Ticket are dispensed by TGS, which knows all the encryption keys
  - TGS seals each ticket with the encryption key of the server.
  - Encrypted tickets can be sent safely over a unsafe network


How Does Kerberos work? - XVI

- Ticket contents
  - Client name (user login name)(C)
  - Server name (S)
  - Client host network address (Addr)
  - Session key for client/and server ($K_{c,s}$)
  - Ticket lifetime(lifetime)
  - Creation timestamp
  - Mathematically, a ticket Tc,s can be represented as (C, S, Addr, $K_{c,s,}$ lifetime, times)


How Does Kerberos work? - XVII

- Session keys
  - Random numbers for specific sessions
  - Used for secure communication between client /server.
- Authenticators: prove a client client's identity. It includes client user name, client network address and timestamp (the client must synchronise with the server). They are sealed with a session key.
- Tickets are reusable


Formal Description of Kerberos -I

- Step 1: Authentication service exchange to obtain ticket-granting ticket
- 1. C->AS: $ID_C||ID_{tgs}||TS_1$;
- 2. AS->C: $E(K_c, [K_{c,tgs}||ID_{tgs}||TS_2||Lifetime2||Ticket_{tgs}])$

  $Ticket_{tgs}=E(K_{tgs}, [K_{c,tgs}||ID_C||ID_{tgs}||TS_2||Lifetime2||AD_c])$

  AD_c means preventing use of ticket from the client other than one that initially requested ticket.


Formal Description of Kerberos -I

- Step 1: Authentication service exchange to obtain ticket-granting ticket

  AD_c means preventing use of ticket from the client other than one that initially requested ticket.

  Note: the client (C) send a request to Authentication server(AS), with a message including ID_C, ID_{tgs} for requesting a ticket for ticket granting service and current timestamp $TS_{1;}$. The AS responds with encrypted key derived from user's password (Kc), a copy of session key $K_{c,tgs}$. current timestamp of the AS, the ticket duration (lifetime2), and Ticket ($Ticket_{tgs}$). When the client receives this response, the client prompts the user for his or her password and generates the key and decrypts the message.

Formal Description of Kerberos -II

- Step 2: Ticket granting service exchange to obtain service-granting ticket
- 3. C->TGS: $ID_s||Ticket_{tgs}||Authenticator$;
- 4. TGS->C: $E(K_{tgs}, [K_{c,tgs}||ID_c||TS_4||Ticket_s])$

$Ticket_{tgs}=E(K_{tgs}, [K_{c,tgs}||ID_C||ID_{tgs}||TS_2||Lifetime2||AD_c])$

$Ticket_s=E(K_s, [K_{c,s}||ID_C||ID_s||TS_4||Lifetime4||AD_c])$

$Authenticator_c=E(K_{c,tgs}, [||ID_C||AD_c||TS3])$

ID_s is to confirm this tickets is for server; $K_{c,s}$ is a copy of session key created by TGS for secure exchange between client and server.


Formal Description of Kerberos -III

- Step 3: Client/server authentication exchange to service
- 1. C->S: $Ticket_s$ ||$Authenticator_c$;
- 2. S->C: E ($K_{c,s}$, [TS_5+1] for mutual authentication

$Ticket_s$=E (K_s, [$K_{c,s}$||ID_C||ID_s ||TS_4 ||Lifetime4||AD_c]

$Authenticator_c$ =E ($K_{c,s}$, [||ID_C||AD_c||TS5])

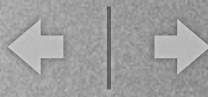
ID is to confirm this tickets is for server; $K_{c,s}$ is a copy of session key created by TGS for secure exchange between client and server.


Kerberos Summary

- Every service request needs a ticket
- Tickets come from the TGS (except for the ticket TGT)
- Every ticket has an associated session key
- Tickets have a finite lifetime
- Authenticators are only used once (new connection to a server)
- Authentication server maintains list of authenticators


Kerberos Realms - I

- A realm is defined by a kerberos server; a single administrative domain contains:
  - a Kerberos server, which stores user and application server password for a realm
  - a number of clients, all registered with the server,
  - application servers, sharing keys with the server

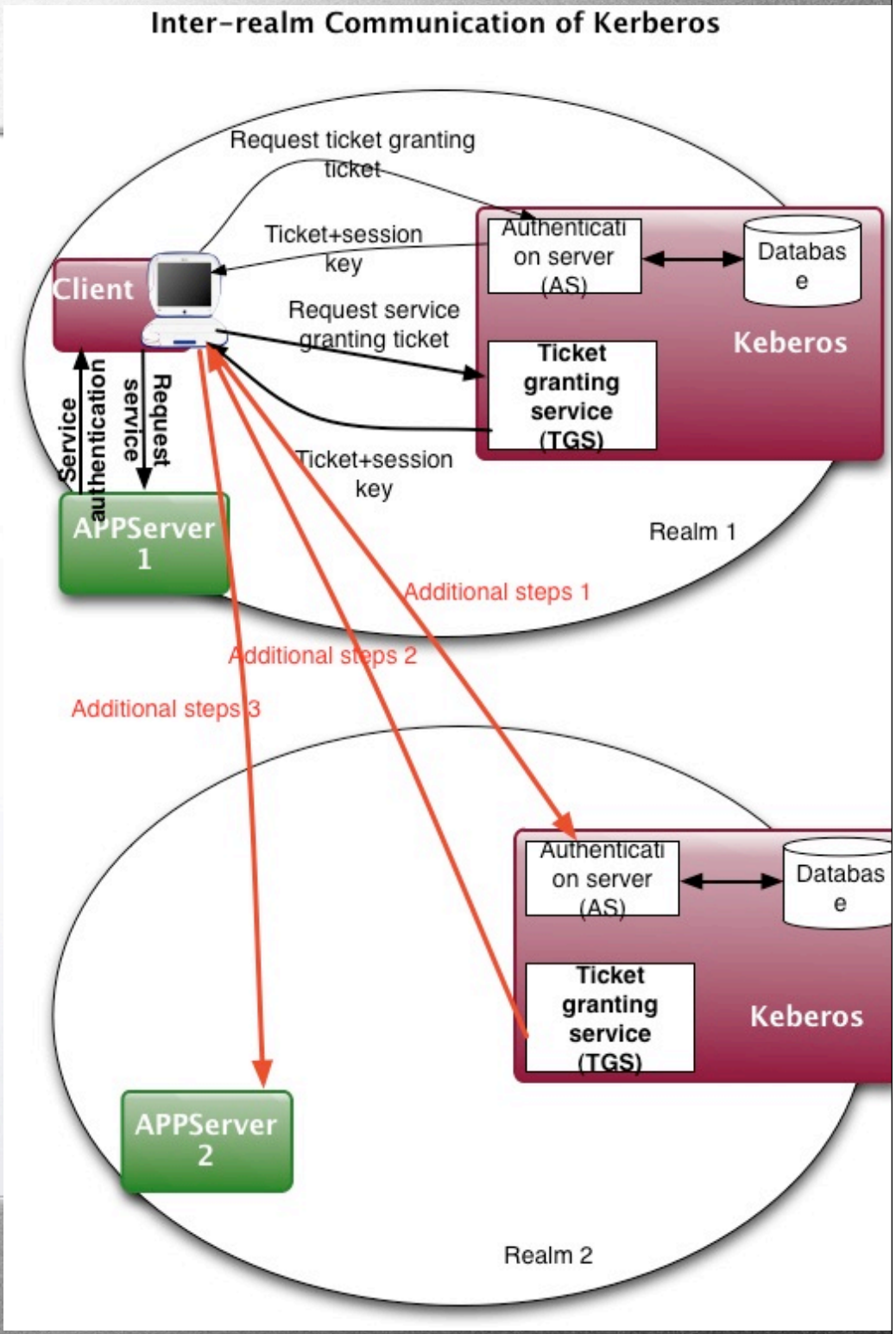



Kerberos Realms - II

- Kerberos provides inter-realm authentication when users in one realm want to access to services from other realms:
  - kerberos servers in each realm share a secret key and servers register with each other (mutual trust)
  - kerberos servers should trust each other so that each of them can authenticate users from other realms
  - Secure key exchange will be N*(N-1)/2
  - Needs additional steps

Kerberos Realms - III

- Inter-realm communication of Kerberos




Kerberos Realms - IV

- The steps for inter-realm communication
  - A client wants to use a service on APPserver2 in realm 2. He needs a ticket for that server
  - The client follows the usual steps to gain access to local TGS in realm 1 and then requests a ticket granting ticket for a remote TGS in realm 2 (Additional step 1)
  - The client can then apply to the remote TGS for a service-granting ticket for the APPserver2 in the realm 2
  - The ticket presented to Appserv2 indicates the user was originally authenticated in realm 1 and the server decides whether the client can use the service.


Kerberos 4 versus Kerberos 5

- Two versions of kerberos:
- Kerberos 4 aims to minimise the amount of time the user's password is stored on the workstation;Kerberos 5 has a longer ticket expiration time
- Kerberos 5 is an Internet standard, specified in RFC1510. It uses ASN.1 (a data representation language standardised by ISO). The Kerbeors 5 uses ASN.1 syntax with basic encoding rules to make it easy for fields to be optional, of varying length, and tagged with type information to allow future versions to include additional encodings
- Kerberos 4 messages had a largely fixed layout with variable length fields marked in an ad hoc way


Kerberos 4 versus Kerberos 5

- Kerberos 4's principal includes "Name, Instance, and Realm", each must be a null-terminated text string up to 40 Characters long.
- While Kerberos 5 has two components: Name and Realm. The name contains a type and a varying number of arbitrary strings. So the purpose served by V4 Instance field is accomplished by using an extra string in the 'Name' component. In V4, Realms are DNS standard names, whereas in V5 they can be DNS standard names or X.500 names, and the syntax allows for other name types as well.


Kerberos 4 versus Kerberos 5

- Delegation of rights are different (the ability to allow someone else access to things you are authorised to access. You can delegate time or scope). kerberos tickets (V4 and V5) contain the specific network layer address. In V4, the network layer address in the ticket is the address from which the ticket was requested. While in V5, it allows multiple addresses to be specified and allow multiple machines to access the resources. for example, as long as Alice specifies bob's or dave's machines addresses and they can act on behalf of Alice to access the resources)
- Ticket lifetimes are different. In V4, maximum lifetime of a ticket was about 21 hours, since the time in a ticket was encoded as a four-octet start time and a one -octet lifetime (in units of 5 minutes)


Kerberos 4 versus Kerberos 5

- Ticket lifetimes in V5 can be issued with virtually unlimited. The timestamp follows ASN.1 format and is 17 bytes long. Some key fields: start time, end time, authtime(first logged in) and renew-till.
- Renewable tickets: although in V5, it has a longer expiration time but unless you renew it (say a day).
- Cryptographic algorithms: V4 uses DES. V5 also uses DES but with modification by combination of different encryption algorithms, such as: rsa-md5-des, des-mac (message-authentication code), etc.


Kerberos 4 versus Kerberos 5

- Hierarchy of realms: in V4, in order for users in realm A to be authenticated in realm B, it is necessary for B's KDC to be registered as a principal in A's KDC. If there are many realms, it will become much complicated.
- While in V5, it allows realms to act as intermediary. For example, if a user in Realm A wants to be authenticated by Realm C. Realm C might not be registered in A. But Realm B is registered in Realm A and Realm C is registered in B. The user can first get a ticket for B and then ask B for a ticket to the KDC in C.


Kerberos 4 vs. Kerberos 5

- New elements in Kerberos 5
  - Indicates realm of the user
- Options
- Times
  - From: the desired start time for the ticket
  - Till: the requested expiration time
  - Rtime: requested renew-till time
- Nonce: a random value to assure the response is fresh

Kerberos 4 vs. Kerberos 5

- Kerberos 5

(1)C → AS : Options || IDc || Realmc || IDtgs ||Times ||

Nonce1

(2) AS → C : Realmc || IDc || Ticket tgs ||

EKc [Kc,tgs || IDtgs || Times || Nonce1 ||| Realm tgs **]**

Ticket tgs= EKtgs [Flags || Kc,tgs || Realm c ||

IDc || ADc || Times**]**

(3) C → TGS : Options || IDv || Times || Nonce2 || Ticket tgs ||

Authenticator c

(4) TGS → C : Realmc || IDc || Ticket v || **EK c,tgs [** Kc,v ||Times||

Nonce2 || IDv || Realm v**]**

Ticket tgs= EKtgs [Flags || Kc,tgs || Realm c || IDc || ADc ||

Times**]**

Ticket v : EK v [Kc,,v || Realmc || IDc || ADc || Times **]**

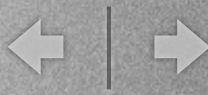
Authenticator c : **EK c,tgs [**IDc || Realmc || TS1**]**


Kerberos Pros vs. Cons -I

- Kerberos pros
  - User's passwords are never transferred across the network
  - Client and Server mutually authenticate
  - It limits the duration of the users' authentication
  - Authentications are reusable and durable
  - Security experts have scrutinised the kerberos


Kerberos Pros vs. Cons -II

- kerberos cons
  - Single point of failure: due to centralised model, if the kerberos server is down, no one can log on
  - Once an attack obtains information from KDC, there is a risk that the attacker can impersonate any user
  - Time of the hosts involved in the network must be synchronised since tickets has a time availability period.
  - Kerberos does not protect against Trojan horses, virus and worms

Further references

- An Introduction to Kerberos : http://www.upenn.edu/computing/pennkey/docs/kerbpres/200207Kerberos.htm
- MIT Kerberos Site : http://web.mit.edu/kerberos/
- The Moron's Guide to Kerberos : http://www.isi.edu/~brian/security/kerberos.html
- Kerberos: The Definitive Guide : http://www.oreilly.com/catalog/kerberos/cover.html