Network Attacks, part 2

Erland Jonsson
Magnus Almgren

General Problems

- TCP/IP is connectionless
- No authentication (v.4)
- Routers only look at destination addresses
- Trusted hosts (in UNIX)
- Users are unsuspecting (social engineering works)

- Port Scanning:
  is a general information-gathering activity. It is a way to find open ports in hosts that can be used for attacks.
  Variant: Stealth Scanning

Some Observations

- Use of security-enhancing tools (SATAN, ISS, …)
  - Can be used for good or bad purposes
- Why study intrusion methods?
  - Discover and report intrusions
  - Apply patches
  - Implement effective protection mechanisms

Spoofing

Spoofing means pretending to be the real owner of an address, which is incorrect.
It may also mean falsely providing a service instead of the real service provider.

- There are several types of spoofing:
  - **IP address** spoofing
  - **ARP** (Address Resolution Protocol) spoofing
  - **Web** spoofing
  - **DNS** (Domain Name Service or System) spoofing

ARP is used in LANs to map the host's IP address to the physical (MAC) address.
DNS translates alphabetic host addresses to IP addresses (is really a network).

IP Spoofing

- **IP address spoofing** exploits the trust relationship between two hosts, the trusted host and the victim host:
  - Send an attack packet to the victim host with a **false source address** (i.e. that of the trusted host)
  - The **victim's replies** would still go to the trusted host. Thus, the attacker does not see them.
  - **Disable the trusted host** in some way (DOS attack?), so that it does not interfere with the communication.
  - Find out the **sequence numbers** somehow, otherwise the spoofed packets will not be accepted by the victim.

Web spoofing

- **Web spoofing** fools the victim to think that he is visiting a legitimate site, whereas he is really visiting the attacker's site.
  - Can be achieved by providing a false link by compromising a common web page.
  - Can also be achieved by providing a false web address, that may be confused with the real one.
    E.g. www.bank.com or www.bank.nu instead of www.bank.se
  - This may cause all communication to pass through the attacker's server.

DNS Spoofing

- **DNS spoofing** means directing users to a false server. This can be accomplished in several ways:
  - By making a fake mapping between hostname and IP address at the victim's web server.
  - By IP spoofing, so that the IP address request goes to a false DNS server.
  - Attacking the real DNS server and changing entries in its cache memory (DNS Poisoning).

ARP Spoofing

- **ARP spoofing** (or poisoning) means sending faked ARP replies to a LAN.
  - The ARP request packet is broadcast to the network segment. Anyone can answer.
  - Giving a wrong answer will confuse network devices, e.g. routers.
  - This may result in that the communication will be directed to an incorrect host.
  - It may also result in that the correct host is unreachable, a DOS attack.

Man-in-the-middle attack

- A **man-in-the-middle attack** is an attack in which the attacker (logically) places himself *between* the two hosts that are communicating. Both hosts think that they are communicating with one another, but both of them are in fact communicating with the attacker's server.
- Can be accomplished in many ways:
  - Using web spoofing
  - By ARP poisoning (available tool: Hunt)
  - By ICMP redirection of packages
  - By DNS poisoning
- The attacker can also passively monitor the communication between the parties, e.g. to collect sensitive information (a passive attack).

Internet Control Message Protocol

Kerberos Basics

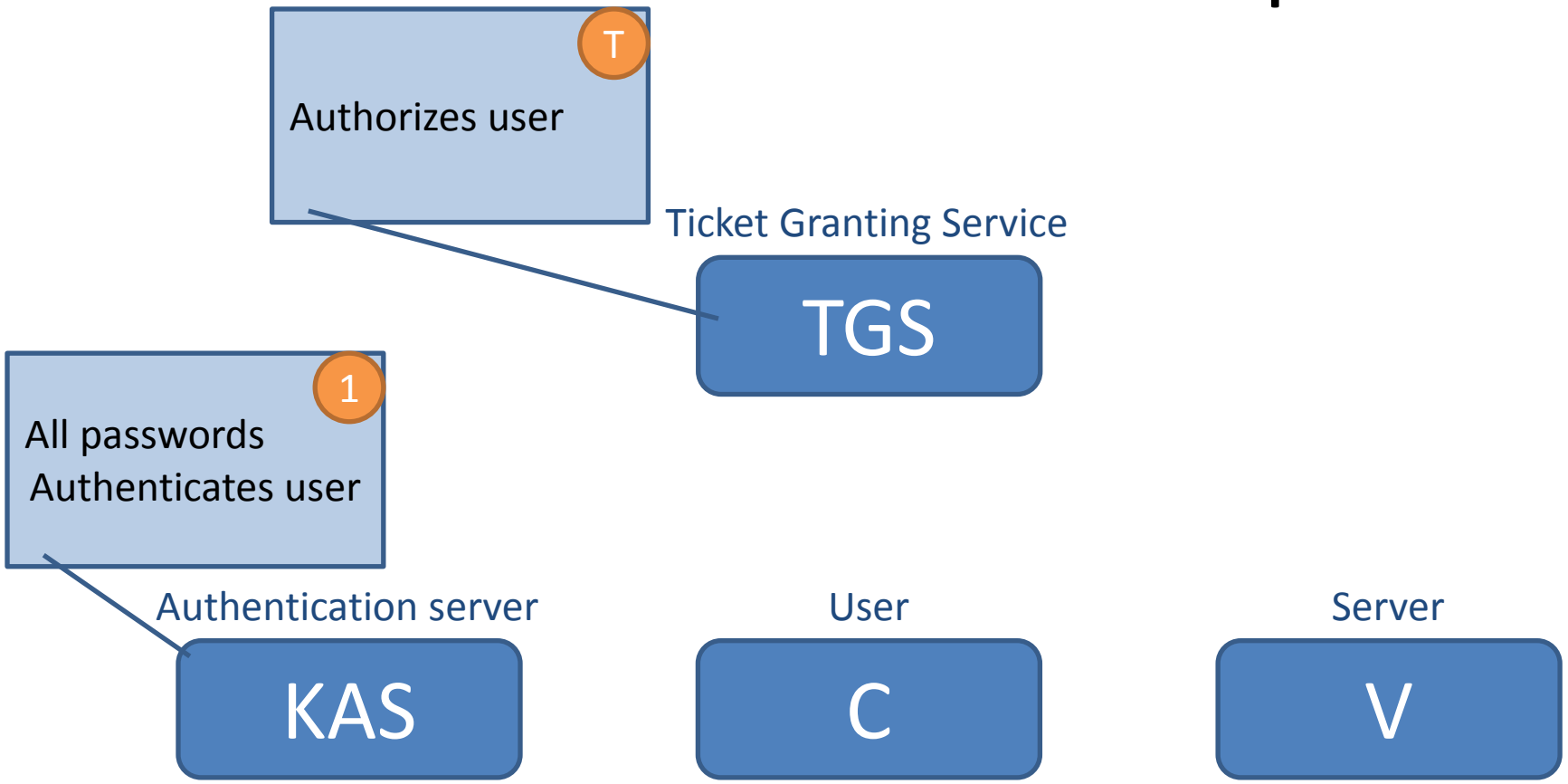
- Kerberos is an **authentication** system (client ←→ server)
- The original requirements were:
  - **Secure** (wrt eavesdropping – not the weakest link)
  - **Reliable** (service should be available when needed)
  - **Transparent** (to the user)
  - **Scalable**
- The system is based on a secure Kerberos installation, where the account name and password are stored.
- Other nodes in the network may be insecure
- Password stored in Kerberos (not locally)
- Passwords are never sent over the network
- The authentication gives a basis for **authorizations**

Authentication separated from authorizations

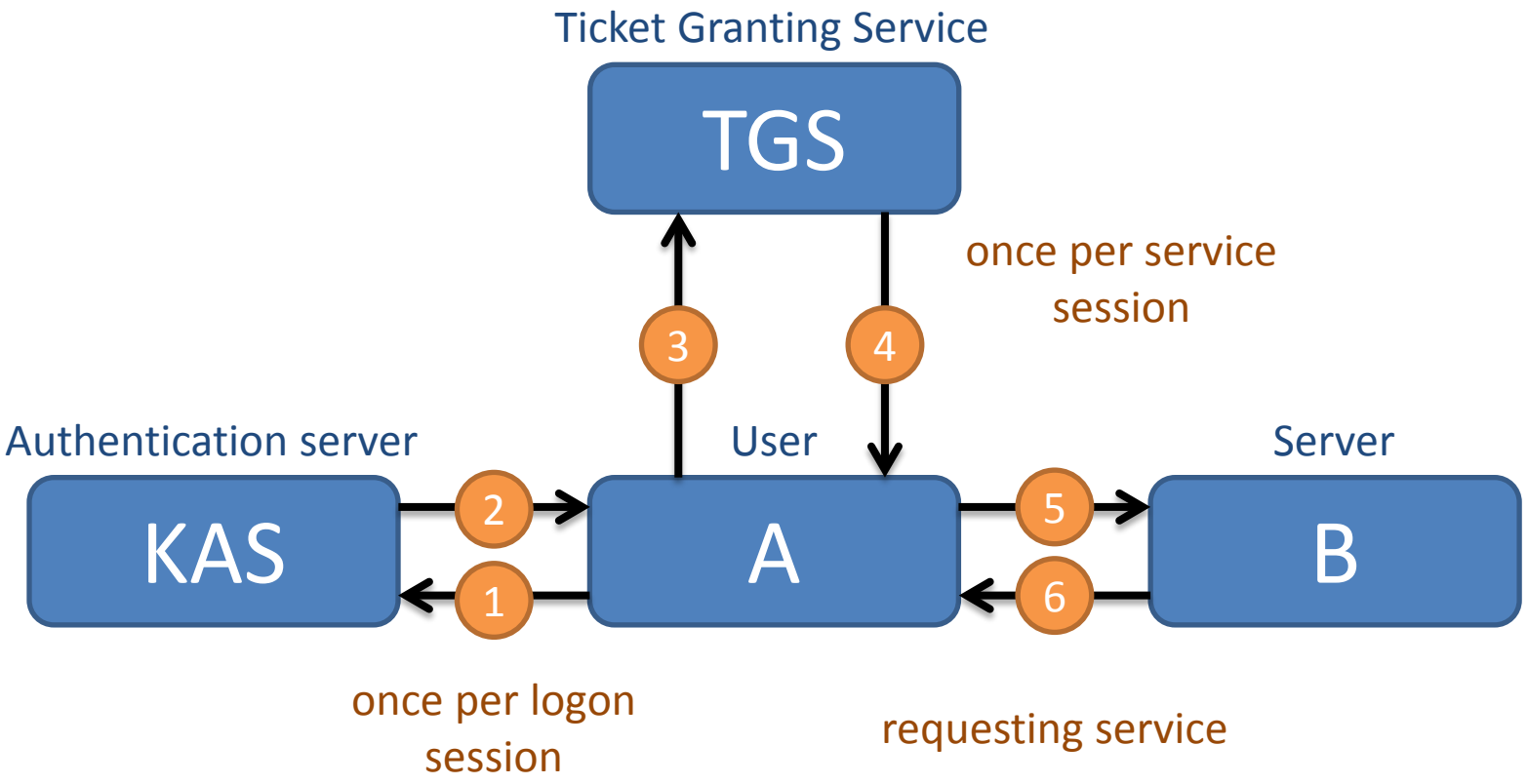
Kerberos Concepts

- **Ticket:** a token used by the user, so that his identity can be securely transferred to the server. It contains the necessary information needed for the user and server to be able to communicate (e.g. crypto keys). One ticket for each service is generated, often valid for hours.
- **Authenticator**: a one-time token showing that the user has the permission to use a service (one ticket per session, short lifetime ∼5 min, to prove user's identity)
- **Session key**: a temporary key for the communication between the user and the service
- **Life time**: the lifetime of a ticket
- **Time stamp:** the time when the ticket was created
- **Nonce:** a random number, to prevent replay attacks

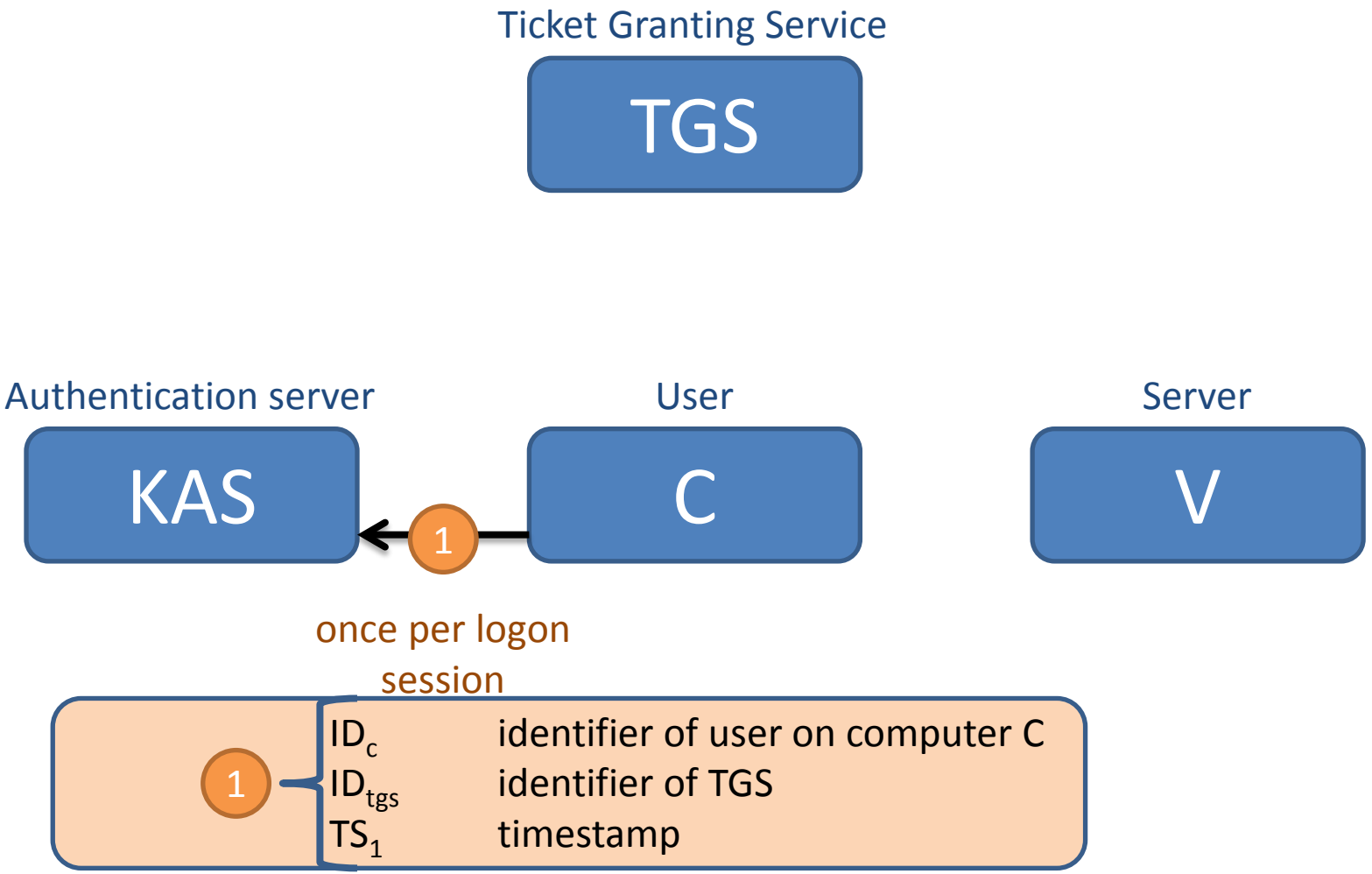
The Kerberos authentication protocol



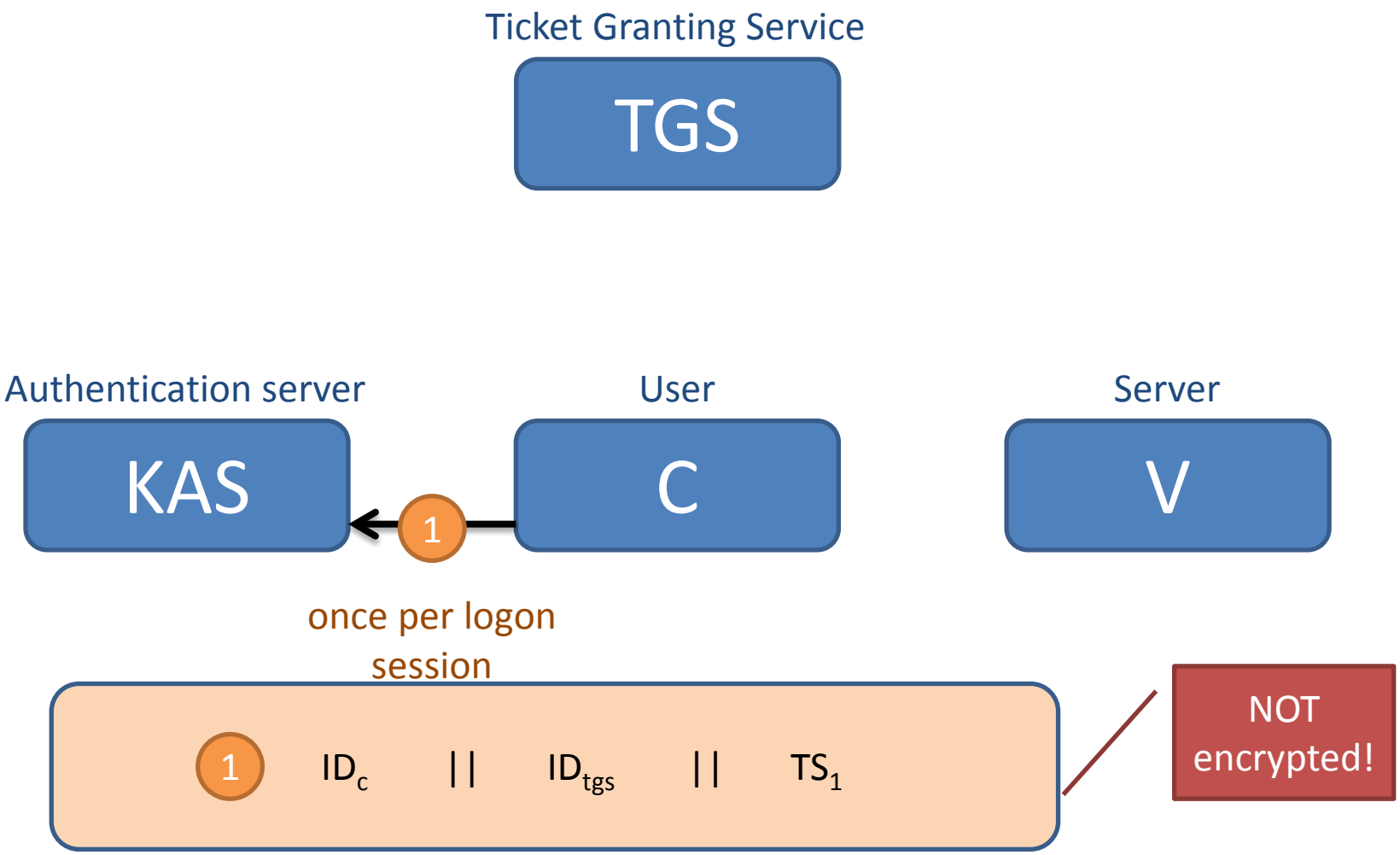
The Kerberos authentication protocol



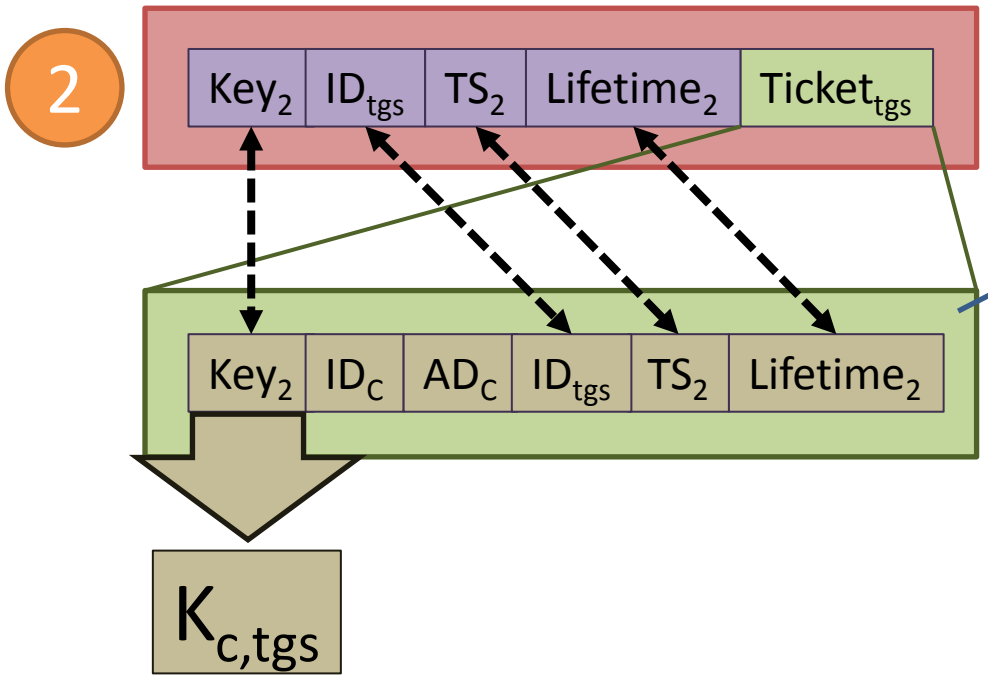
The Kerberos authentication protocol



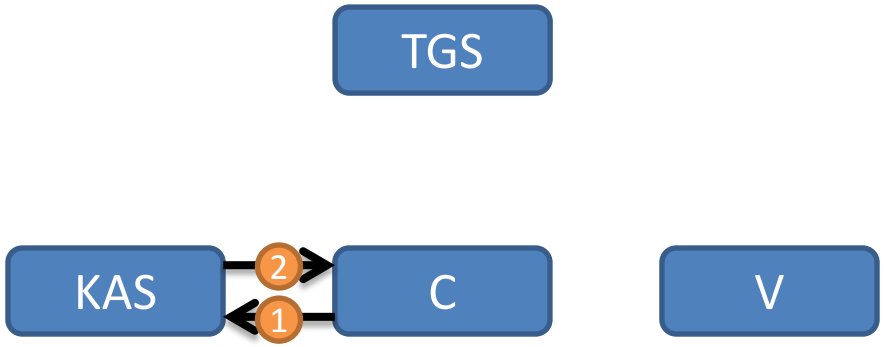
The Kerberos authentication protocol



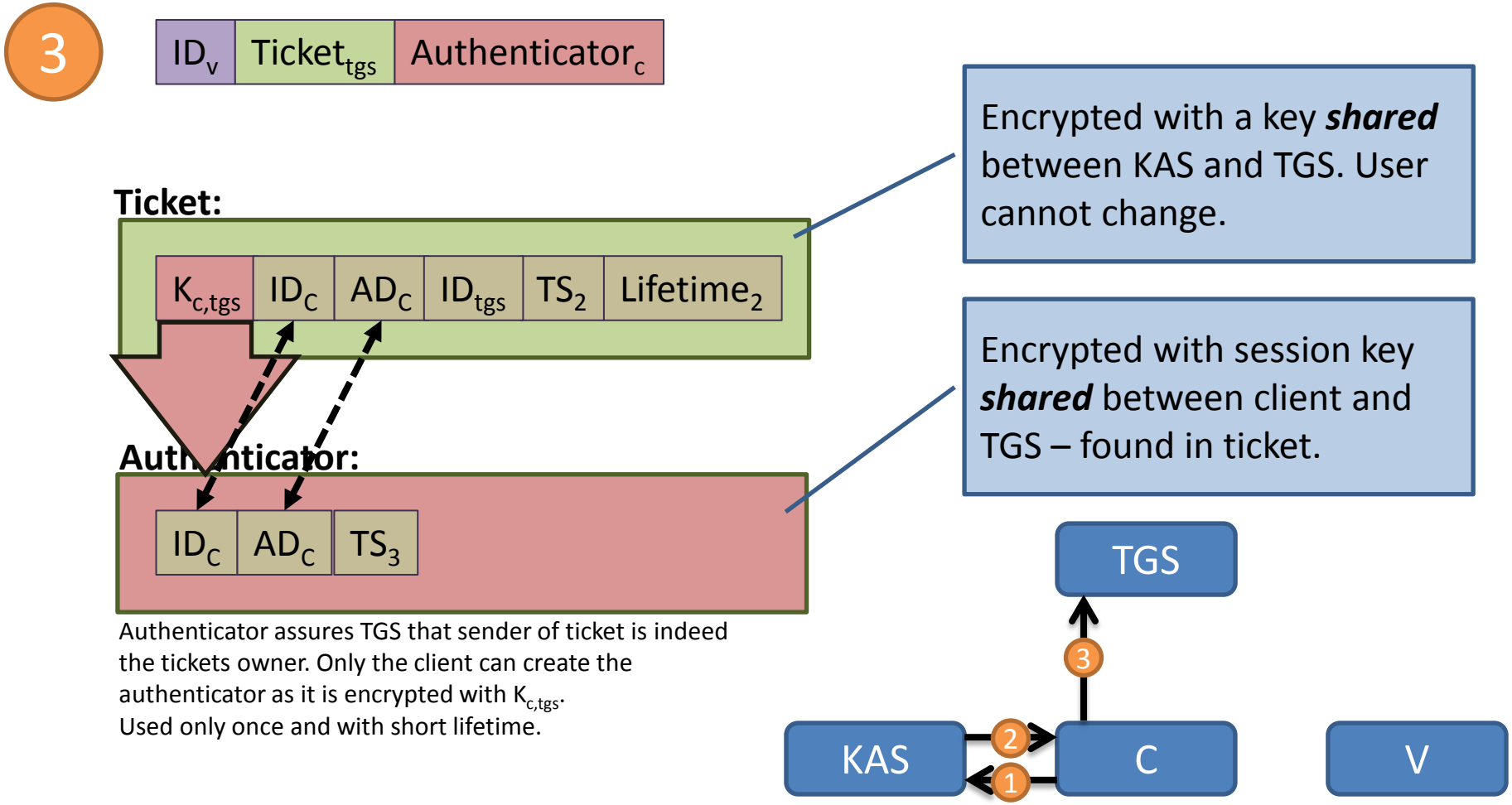
The Kerberos authentication protocol



Encrypted with a key ***shared*** between KAS and TGS. User cannot change.



The Kerberos authentication protocol



The Kerberos authentication protocol (3)

- A→TGS: ID_v || $Ticket_{tgs}$ || $Authenticator_c$
  - ID_v: Client wants access to server V (ID_v)
  - $Ticket_{tgs}$: Assures TGS user authenticated by AS
    E_{ktgs} [$K_{c,tgs}$||ID_c||AD_c||ID_{tgs}||TS_2||$Lifetime_2$]
    - Ticket encrypted with key from TGS / AS (client cannot touch)
      - Key shared with client (to create/read authenticator)
      - ID_c = identity of user, AD_c = network address
      - ID_{tgs} = decrypt OK?
      - TS_2, $Lifetime_2$ = when issued and how long valid
  - S: Authenticatorc = $E_{Kc,tgs}$[Id_c || Ad_c || TS_3]
    - Used only once, with short lifetime
    - Check info with that within ticket + incoming message
    - ”At time TS_3, I use $K_{c,tgs}$” – only client has access to $K_{c,tgs}$