Web Access Of Kerberized Services

Aniket P Kate, Sapna Jain, Kriti Puniyani

Oct 31, 2004
Under the Guidance Of
Prof. G. Sivakumar

Kerberos

Kerberos is a network authentication protocol that lets clients and servers reliably verify each others identities before establishing a network connection.

Kerberos

Kerberos is a network authentication protocol that lets clients and servers reliably verify each others identities before establishing a network connection.

Centralized Authentication System It authenticates users to servers and servers to users.

Kerberos

Kerberos is a network authentication protocol that lets clients and servers reliably verify each others identities before establishing a network connection.

Centralized Authentication System It authenticates users to servers and servers to users.

Provides Single Sign-on User has to log on to the Kerberos server only once and can then access all Kerberized services without repeated authentication.

Kerberos

Kerberos is a network authentication protocol that lets clients and servers reliably verify each others identities before establishing a network connection.

Centralized Authentication System It authenticates users to servers and servers to users.

Provides Single Sign-on User has to log on to the Kerberos server only once and can then access all Kerberized services without repeated authentication.

Authentication Server (AS) AS authenticates the user and creates a Ticket Granting Ticket (TGT).

Kerberos

Kerberos is a network authentication protocol that lets clients and servers reliably verify each others identities before establishing a network connection.

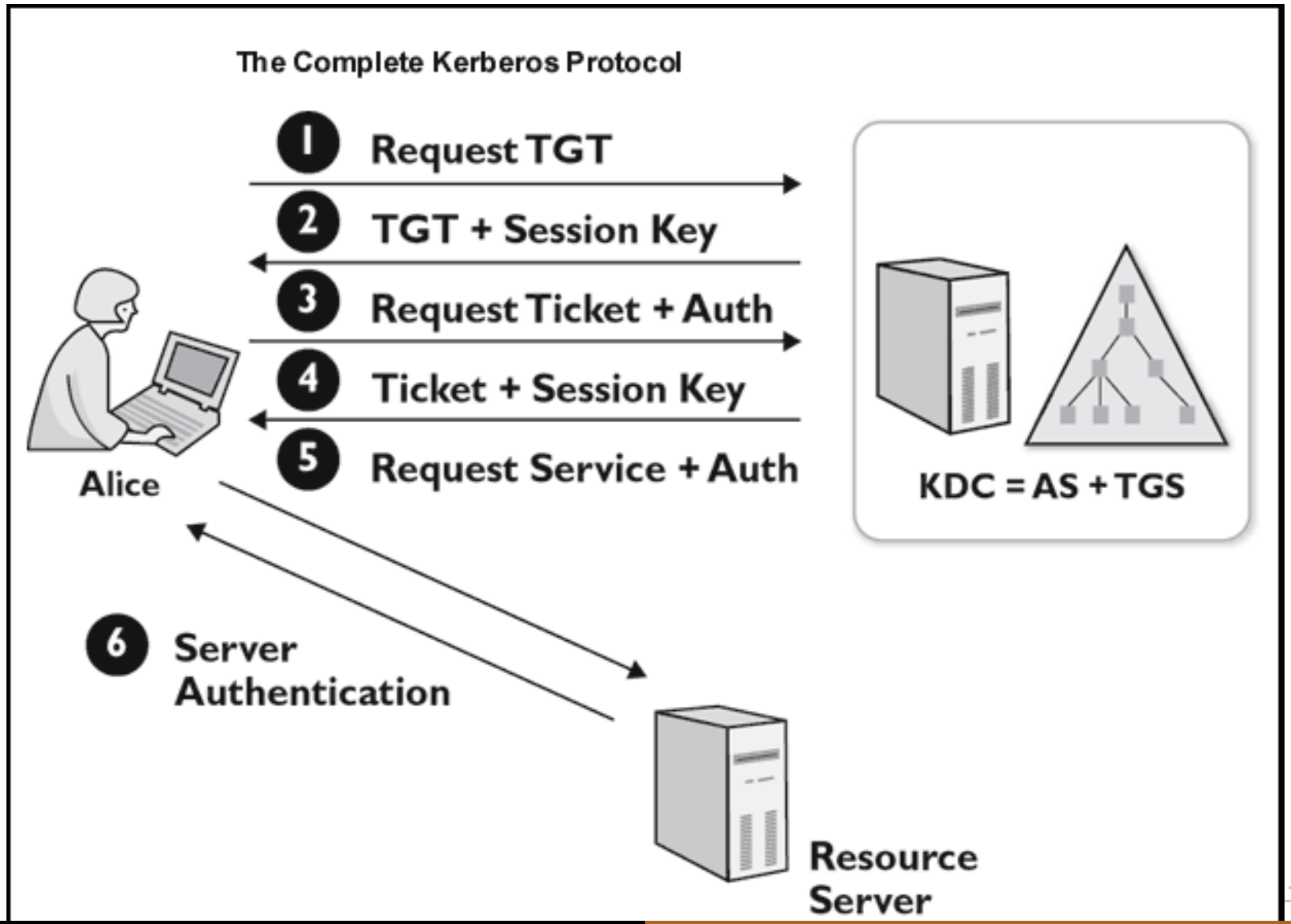
Centralized Authentication System It authenticates users to servers and servers to users.

Provides Single Sign-on User has to log on to the Kerberos server only once and can then access all Kerberized services without repeated authentication.

Authentication Server (AS) AS authenticates the user and creates a Ticket Granting Ticket (TGT).

Ticket Granting Server (TGS) Once the user acquires a TGT, the TGS creates tickets for the requested service. Using that ticket, the user can access the Kerberized service.

Overview of Kerberos Protocol



Secure Socket Layer (SSL)

Secure Socket Layer (SSL)

Securing Web Communication Secure Sockets Layer is a protocol developed by Netscape for transmitting data securely via the Internet.

Secure Socket Layer (SSL)

Securing Web Communication Secure Sockets Layer is a protocol developed by Netscape for transmitting data securely via the Internet.

Provides Authentication The SSL security protocol provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection.

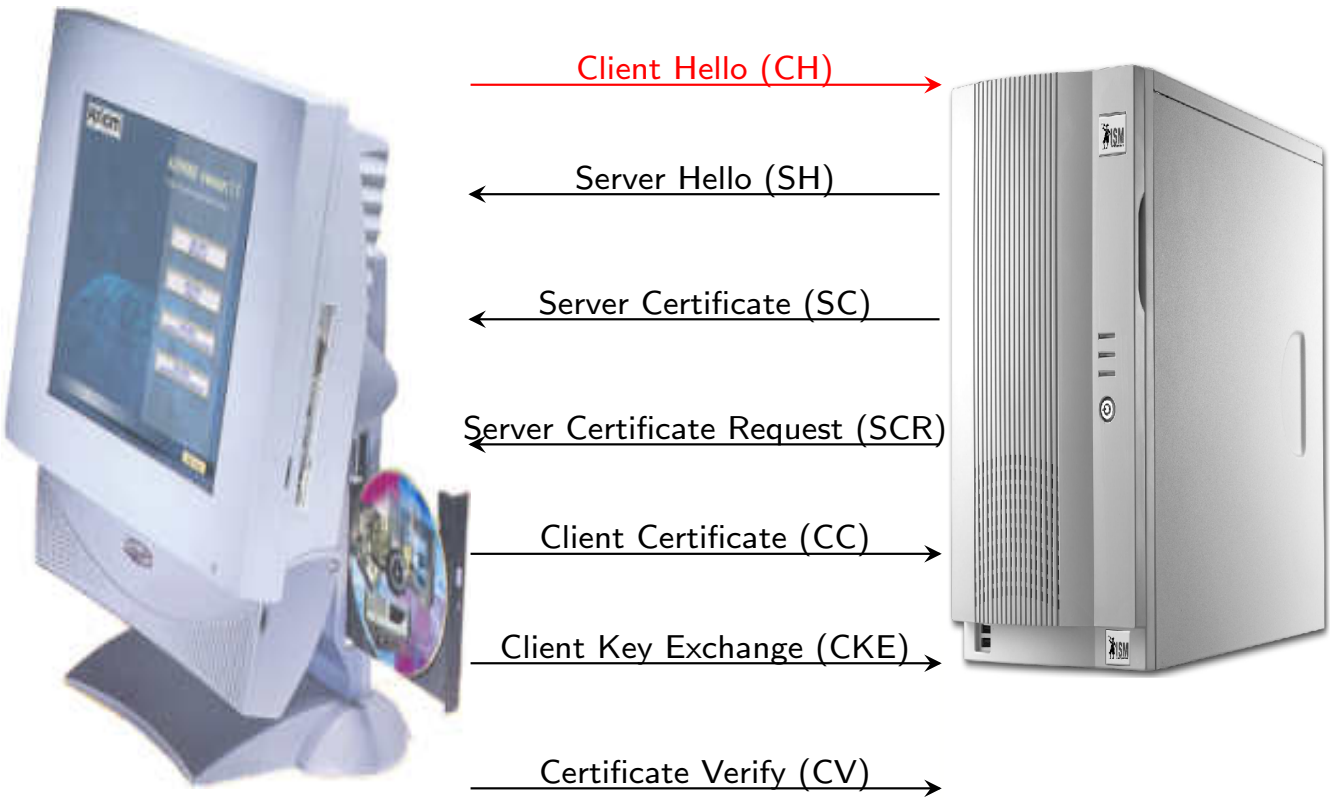
Secure Socket Layer (SSL)

Securing Web Communication Secure Sockets Layer is a protocol developed by Netscape for transmitting data securely via the Internet.

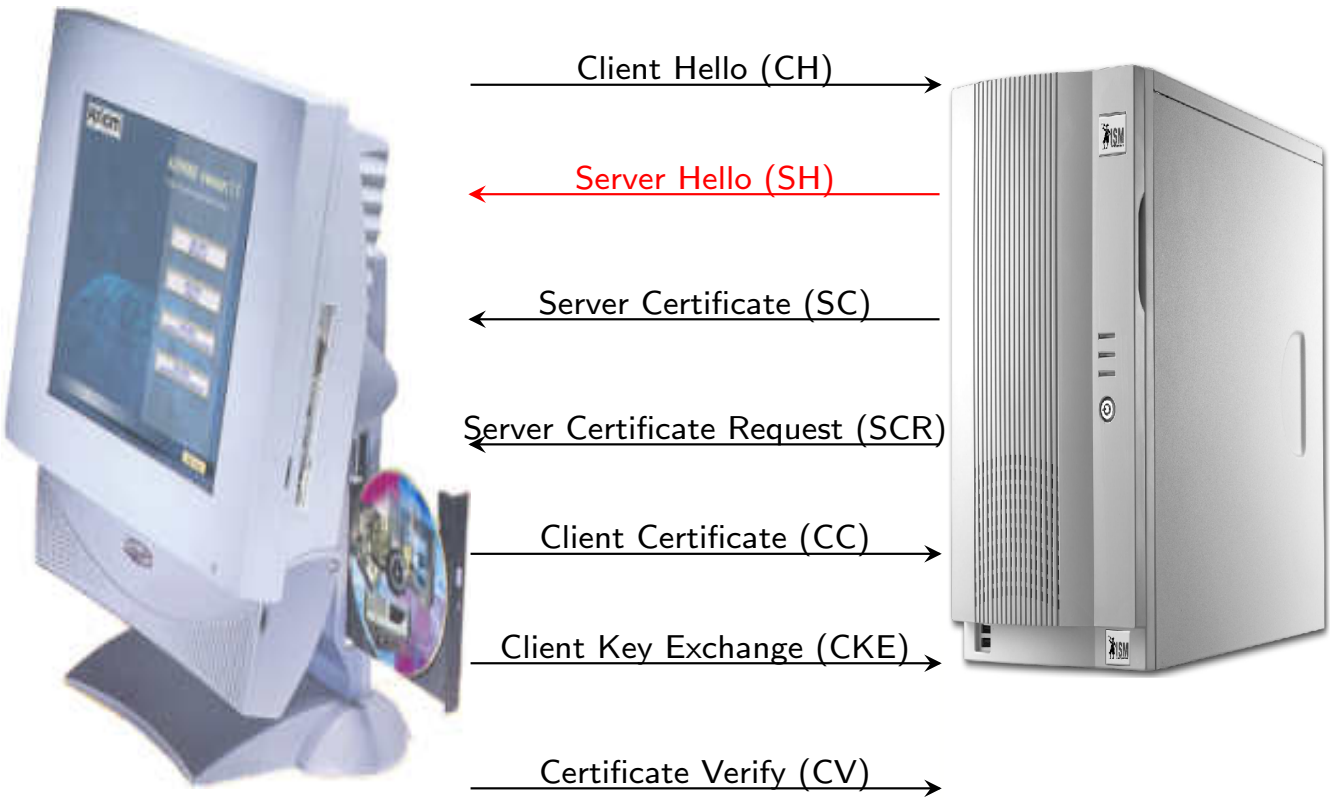
Provides Authentication The SSL security protocol provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection.

Uses Certificates SSL uses public key cryptography, in particular certificates for authentication, and secret key cryptography to provide confidentiality and integrity of message.

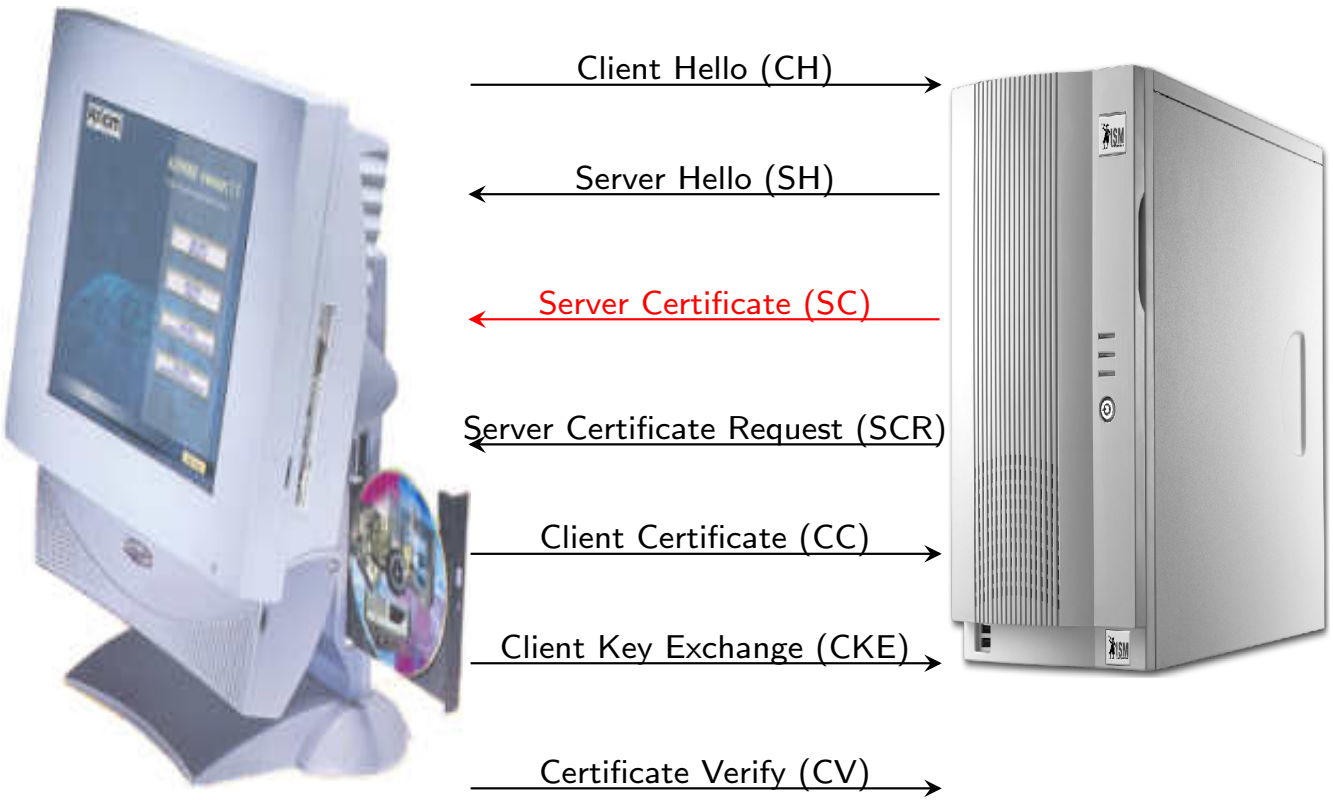
Overview of SSL Protocol



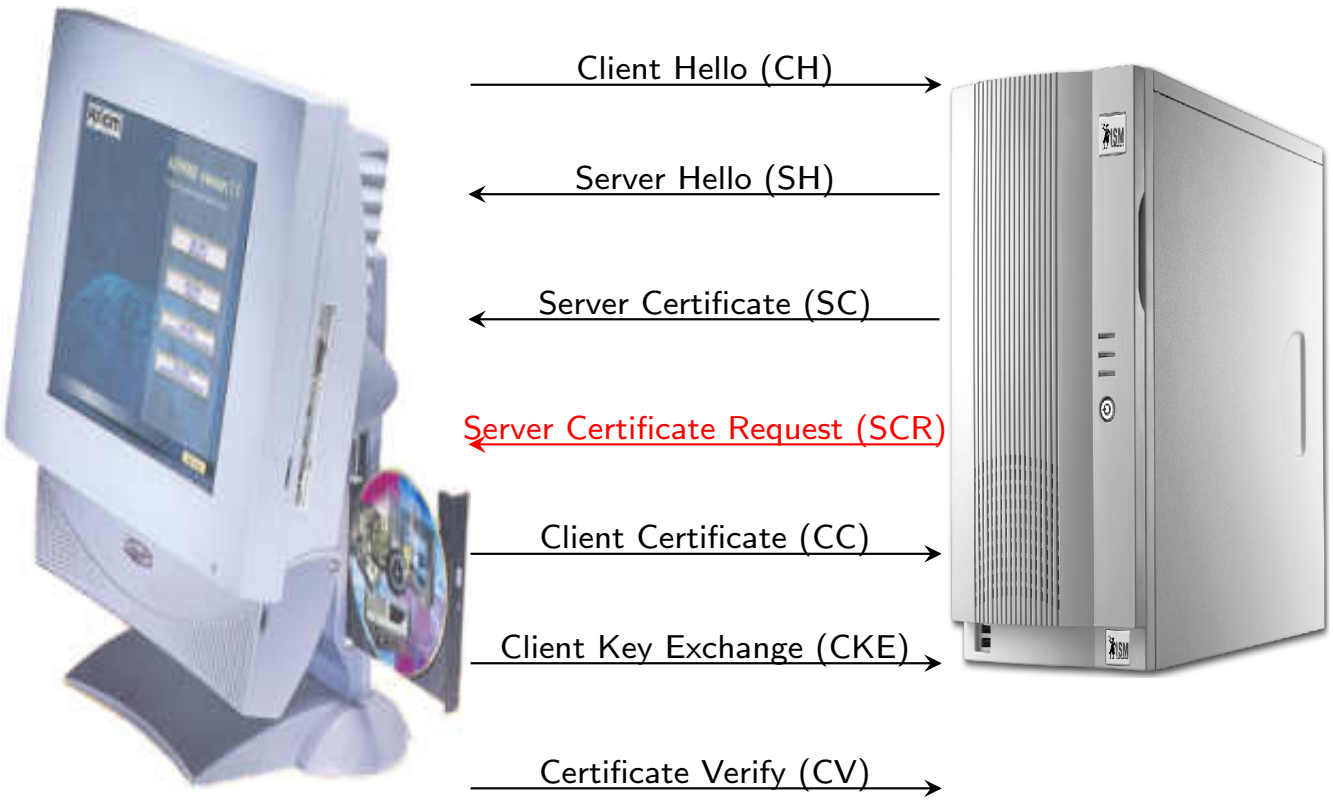
Overview of SSL Protocol



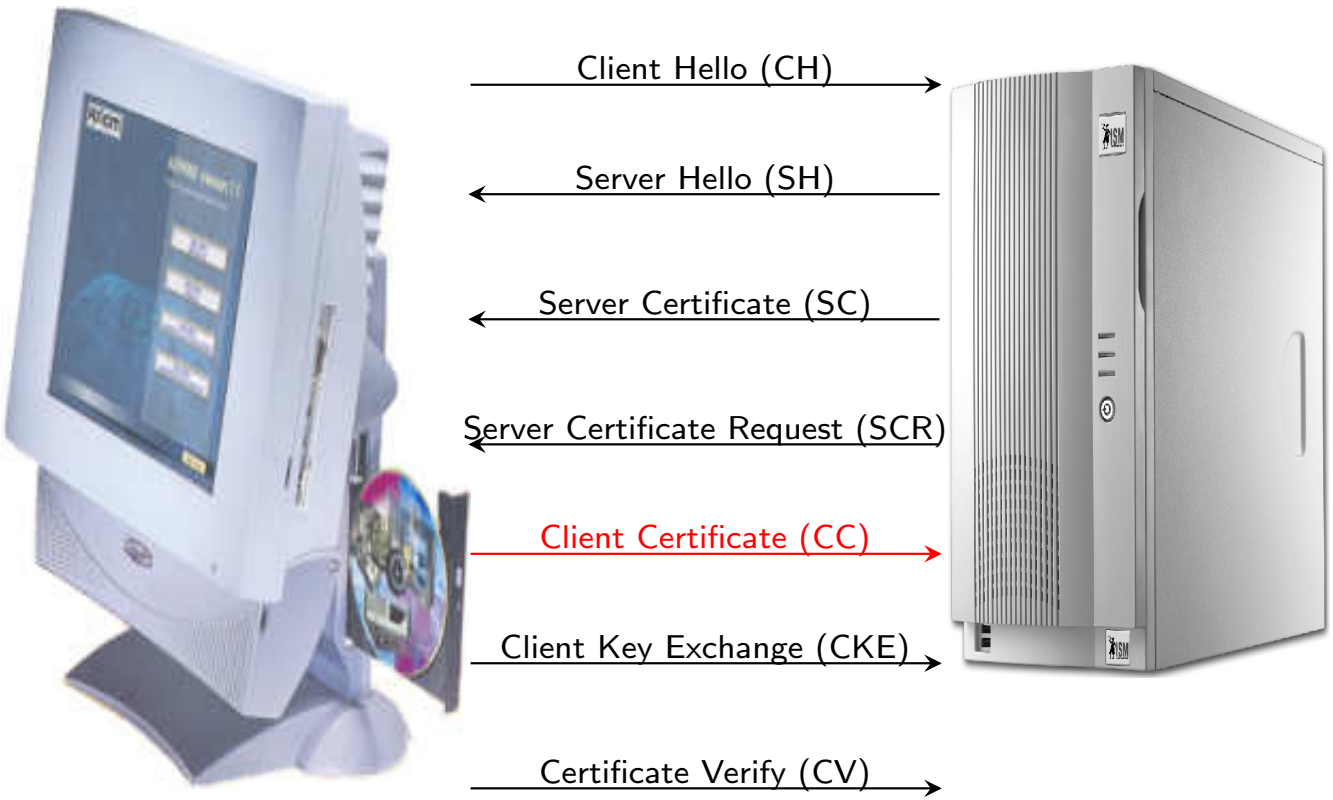
Overview of SSL Protocol



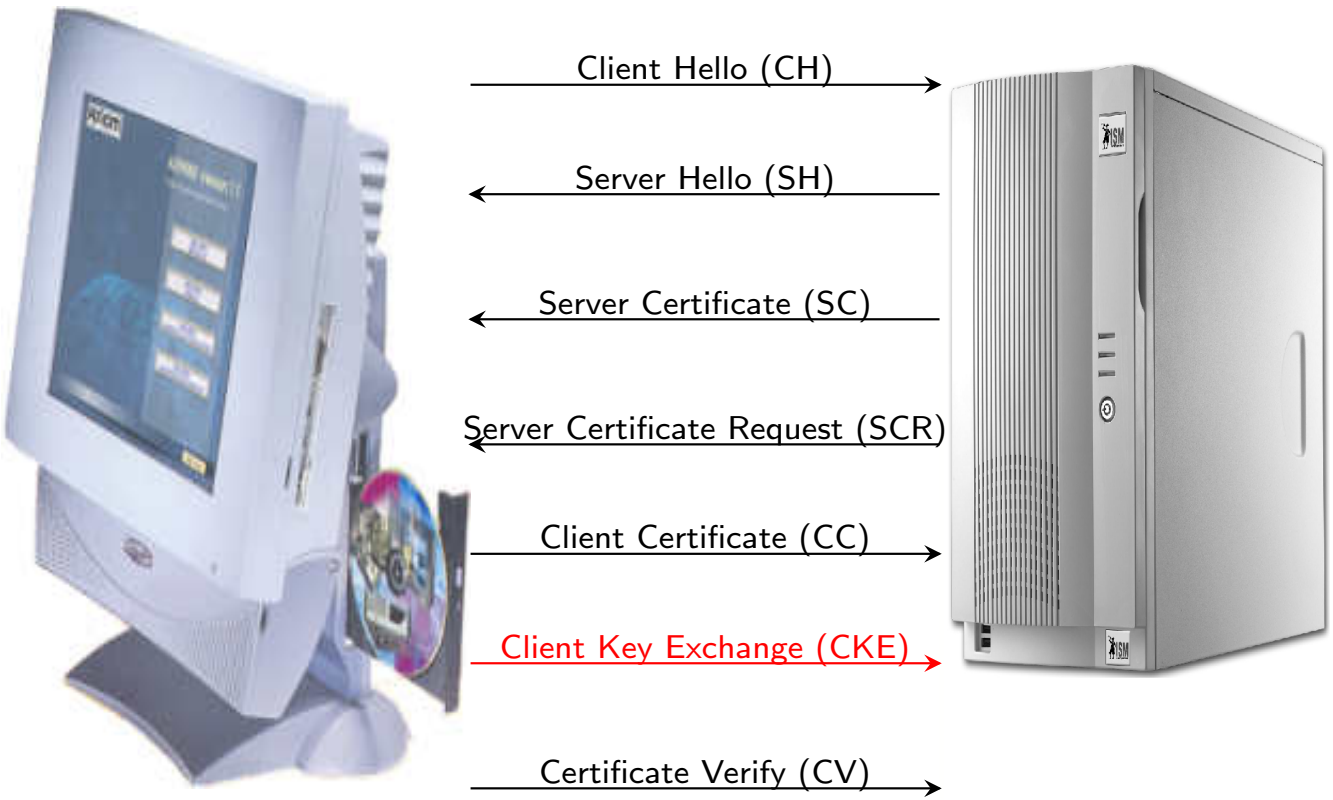
Overview of SSL Protocol



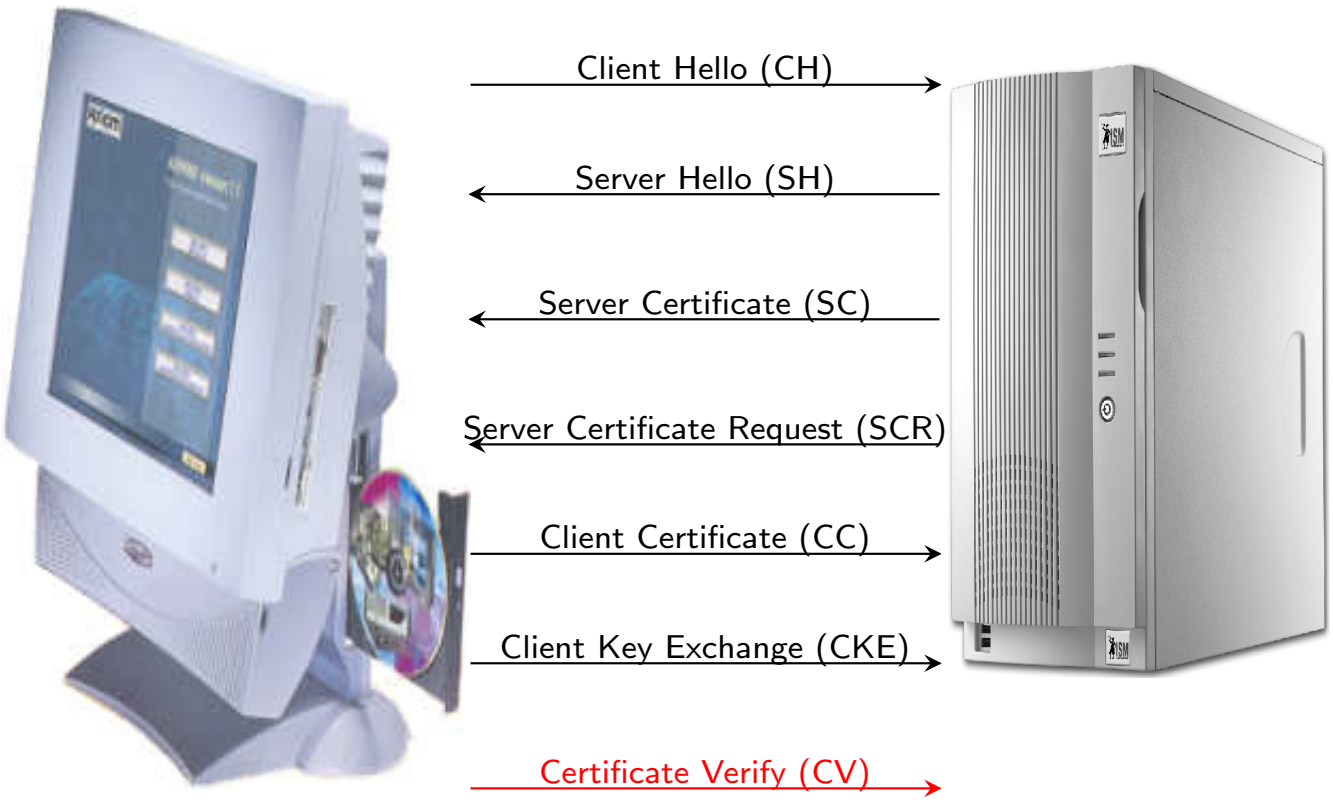
Overview of SSL Protocol



Overview of SSL Protocol



Overview of SSL Protocol



Need for Web based Access to Kerberized services

If you have a distributed system with multiple services being offered, using the single sign-on capability and security provided by Kerberos, then such services cannot be accessed securely over the internet.

- IIT system is Kerberized - print service, mail service, NFS
- Professor Siva wants to access tutorials on his NFS directory, set a question paper, and print it (and also check his mail simultaneously) from outside the campus.
- Allowing access to the Kerberized services directly over the internet will compromise the security in Kerberos.
- SSL is almost universally used for secure web transactions.
- Hence we aim to use SSL to provide secure web based access to Kerberized services.

Problem Definition

Problem Definition

We aim to provide support for secured Web-based access to Kerberized services, where SSL will be used for secure connection between the client and the web server.

Problem Definition

We aim to provide support for secured Web-based access to Kerberized services, where SSL will be used for secure connection between the client and the web server.

However, SSL provides authentication using public key credentials, while Kerberos utilises tickets for the same.

Problem Definition

We aim to provide support for secured Web-based access to Kerberized services, where SSL will be used for secure connection between the client and the web server.

However, SSL provides authentication using public key credentials, while Kerberos utilises tickets for the same.

Hence, we need a bridge between the secure Web communication using SSL and the secure Intranet authentication using Kerberos.

Related Work

Web Server impersonating Client

1. Client will provide kerberos identity and password to web server through SSL.
2. Web server will do the authentication for client at the Kerberos server.
3. Then client will request for the required Kerberized service along with parameters.
4. The Web server impersonating the user will acquire the service ticket, and provide the service to the user.

Related Work

Web Server impersonating Client

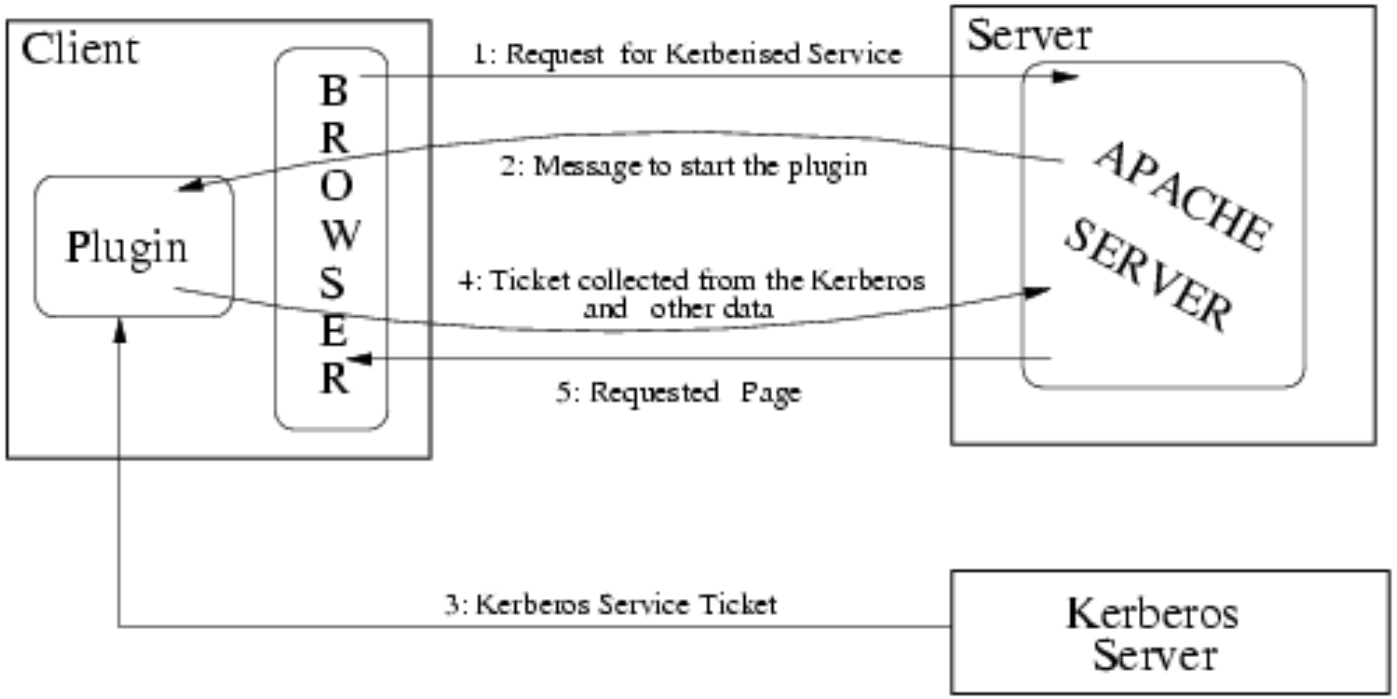
1. Client will provide kerberos identity and password to web server through SSL.
2. Web server will do the authentication for client at the Kerberos server.
3. Then client will request for the required Kerberized service along with parameters.
4. The Web server impersonating the user will acquire the service ticket, and provide the service to the user.

Risk Involved

This gives unlimited power to Web Server to impersonate users, which is a significant security risk.

Related Work Contd...

Plugin on client side



Design Criteria

The following points are hence taken into consideration during the design:

1. Use existing security infrastructure and mechanisms for authentication and authorization.
2. Restrict and Control Web Server actions through authorization mechanisms.
3. Use off-the-shelf software as much as possible, and modify the Web Server, rather than the browser.
4. Added features should not require additional user interaction, providing transparent access to resources.

Our Proposed Solution - Kerberos Credential Translator

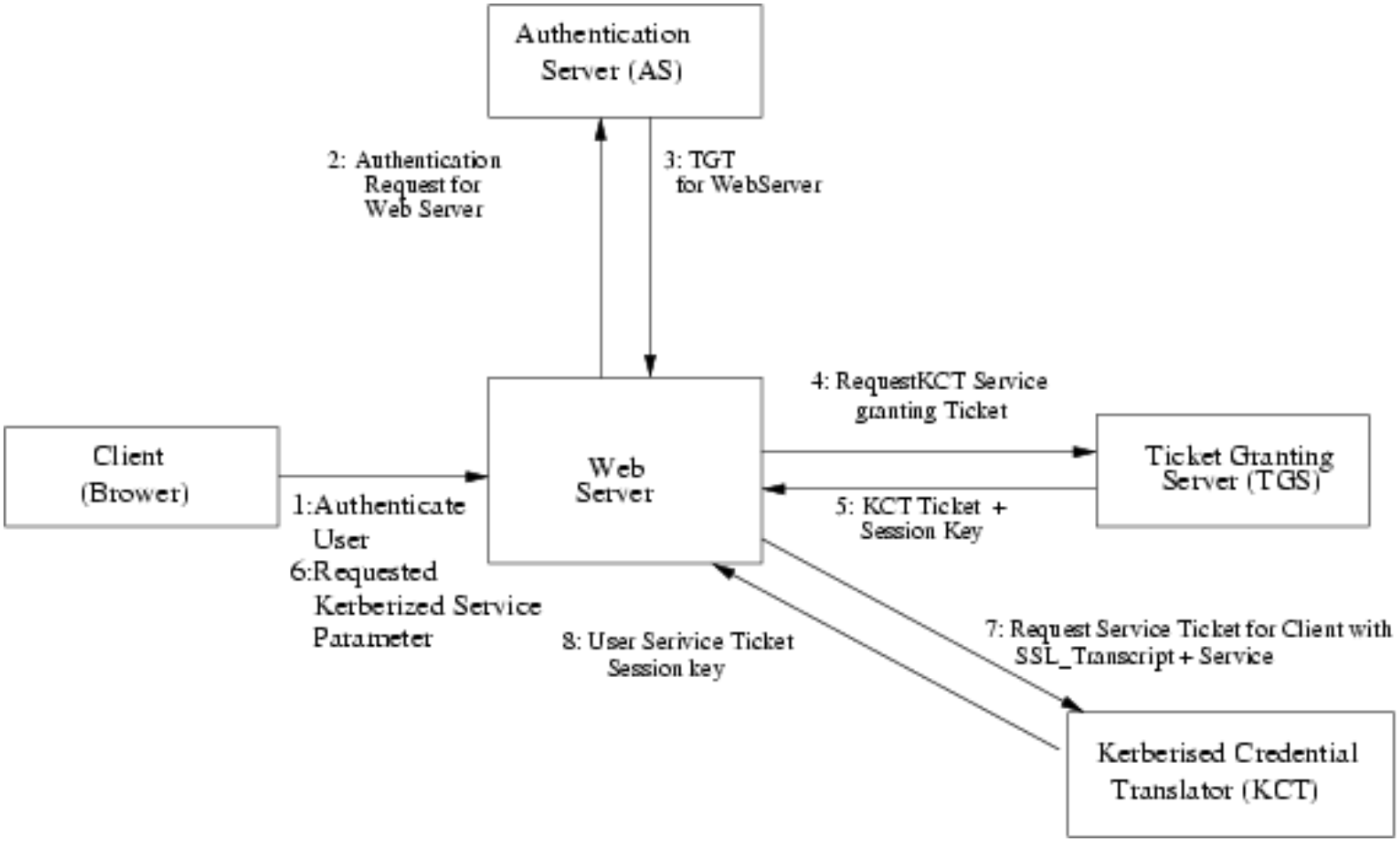
Kerberos Credential Translator

Backend services use Kerberos for authentication, while the Web Server uses SSL with Public key cryptography for the same.
We use a Kerberos Credential Translator (KCT), that translates PK credentials of the user into a Kerberos service ticket.

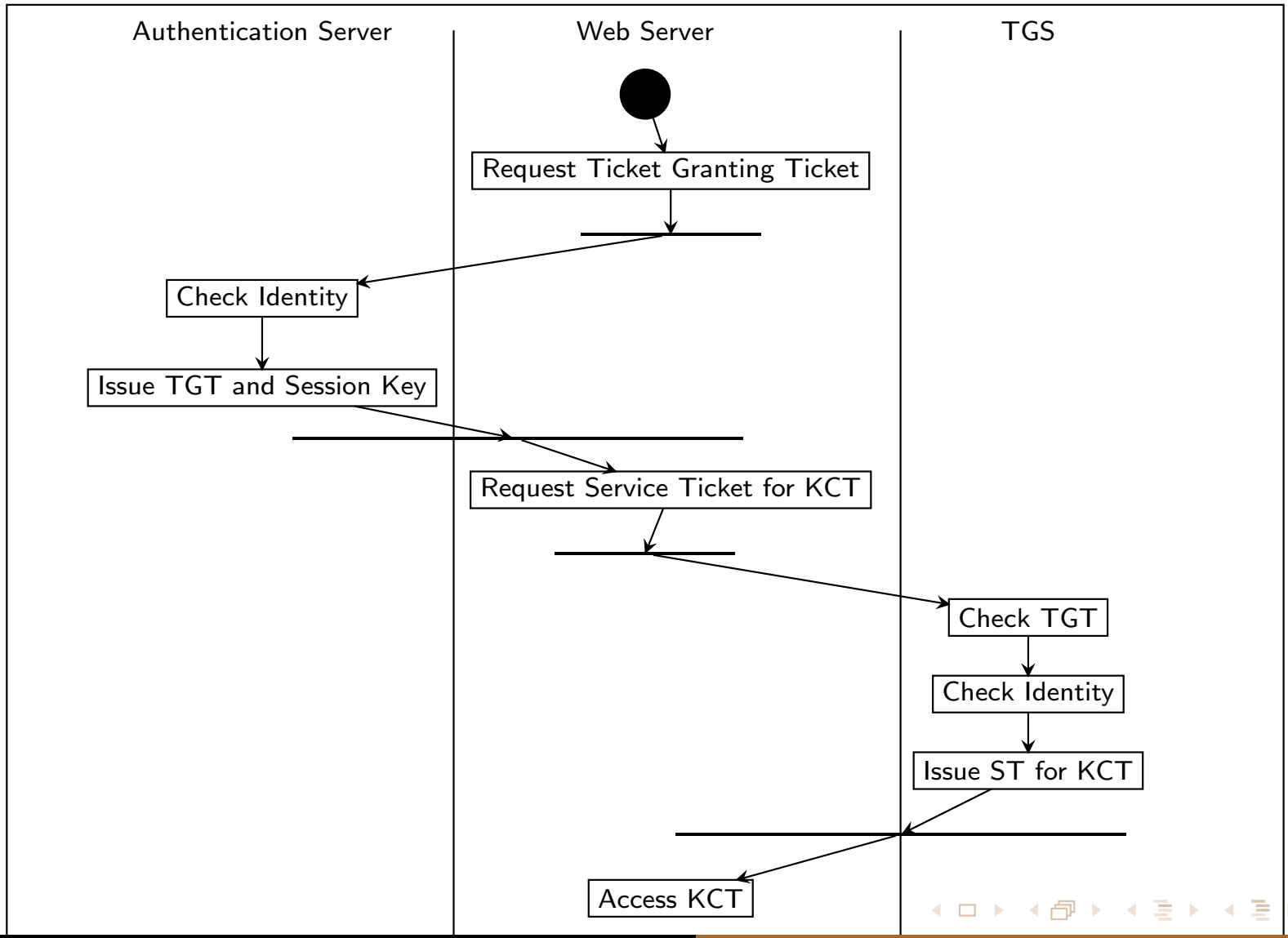
Process

- The user is authenticated by the Web Server using SSL and his public key credentials.
- The Web server gives the proof of authentication to the KCT.
- The KCT authenticates both the Web server and the user.
- The KCT then provides the corresponding Kerberos service ticket for the user to the Web server.
- Using the ticket provided, the Web server can access the Kerberized service, and provide it to the user.

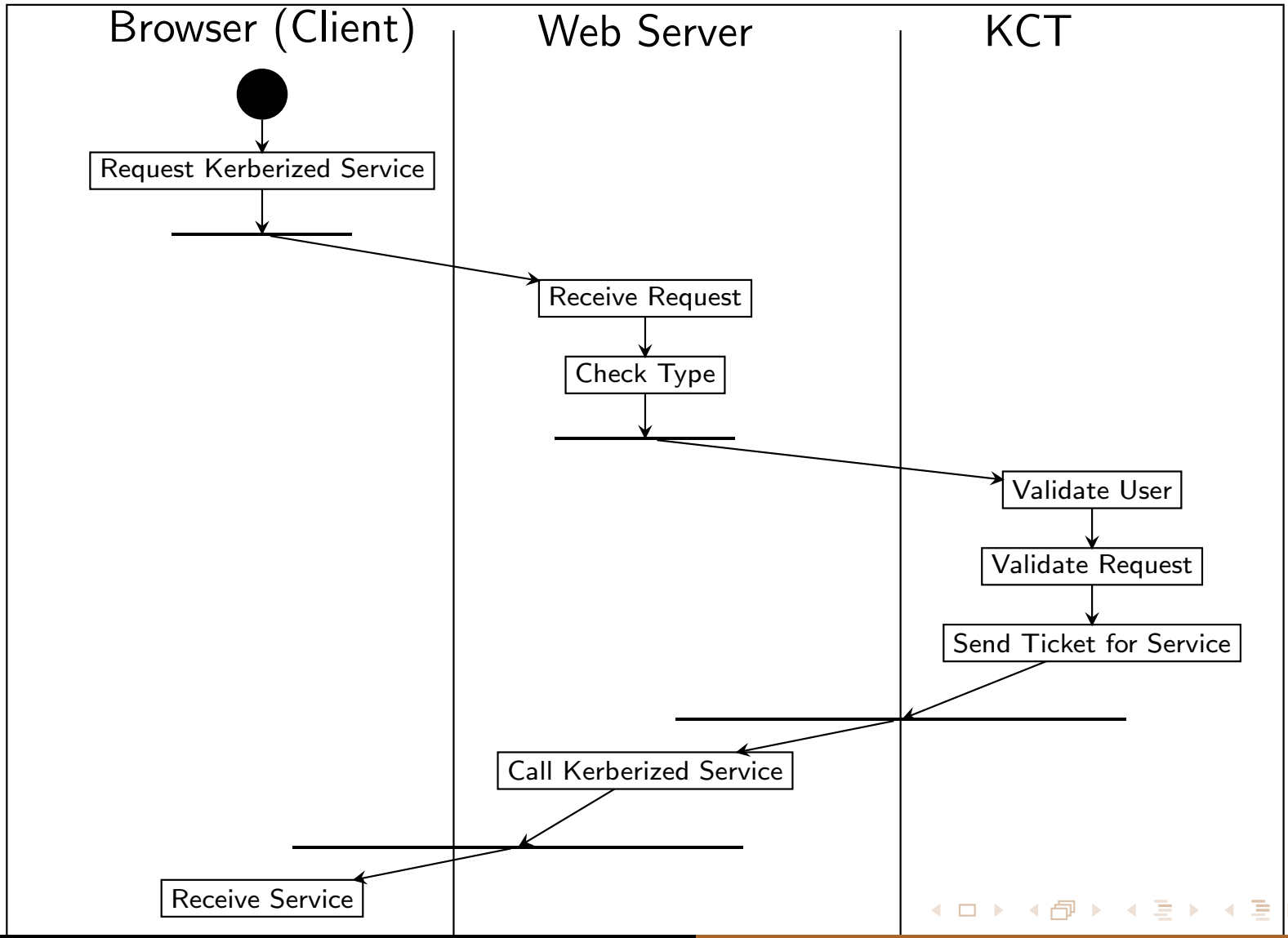
Collaboration Diagram



Activity Diagram 1



Activity Diagram 2



Kerberized Credential Translator(KCT) Design

The KCT accepts the transcript of the SSL handshake from the Web server along with the Web Server's KCT-ticket and the service for which a ticket is required by the client, and performs the foll. steps:

1. Validate user and server certificates.
2. Verify client signature by recomputing the hash of the handshake.
3. Verify that the server certificate identity matches the Kerberos identity.
4. Check the timestamp to ensure the time validity.
5. Generate a service ticket for the user.
6. Encrypt the service ticket using server's session key.
7. Return the encrypted ticket to the webserver.

Writing a module for Apache

1. A function called **hook type-checker** is written to check if incoming request should be handled by the module.
2. Define various **hook functions** that are called for the processing of various events of the module.
3. Define a module data structure that defines the interaction between the http_core, and the module.
4. Register all hooks within the module with the http_core so that they are visible and accessible to other modules.
5. Define configuration parameters for the module.
6. Compile the module, and integrate it with Apache using apxs to register the module in httpd.conf, and move the object into the Apache lib directory.

Web Server Design : Modification of Apache Code

Addition of module : kct_mod

Status : Experimental

Modules with which it will interact: http_config.h, http_core.h, http_log.h

Module Data Structure: kct_mod (glue between the httpd core and the module)

```
module AP_MODULE_DECLARE_DATA kct_mod = {
        STD20_MODULE_STUFF,     /*defines config parameters*/
        NULL,                   /*create per-dir config struct*/
        NULL,                   /*merge per-dir config struct*/
        NULL,                   /*create per-server config struct*/
        NULL,                   /*merge per-server config struct*/
        kct_mod_config_cmds,    /*config directives table*/
        kct_mod_register_hooks  /*register hooks*/
        };
```

Hook Functions Defined

kct_service_ticket() :

1. if (!TGT) kct_server_authenticate()
2. if (!KCT-ticket) kct_access()
3. get transcript of user authentication from file
4. send transcript, service request, and KCT-ticket to KCT
5. receive service ticket from KCT and store in a file

kct_server_authenticate() :

1. authenticate self to AS
2. get TGT from AS and store in file

kct_access() :

1. send TGT to TGS, and request for KCT access ticket
2. get KCT-ticket from TGS and store in file

Configuration And Registration

Register Hooks Define a kct_mod_register_hooks() function

Hook type-checker Checks if incoming request is for a kerberized service. Defined as a kct_mod_method_handler

Configuration Parameters Defined in kct_mod_config_cmds

1. kerberized=TRUE
2. address-of-KDC
3. address-of-TGS
4. address-of-KCT

OpenSSL Library

1. The SSL protocol has to maintain a transcript of the Handshake.
2. Hence the OpenSSL library has to be modified to keep track of incoming and outgoing handshake messages, and copy them into a transcript file.
3. This is a minor modification to the library.
4. The transcript contains:
   - client_hello and server_hello messages
   - server certificate and server_hello_done messages
   - client certficate and client_key_exchange
   - certificate_verify message

Implementation Plan

- Implement the Kerberised Credential Translator(KCT) modules as defined in the design.
- Implement the kct_mod module designed for adding KCT support to the Apache Web Server, as explained.
- Modify the OpenSSL Library to be able to record the transcript of the SSL handshake in a file.

Conclusions

- Thus, we designed a system for providing secure Web based access to Kerberized services.
- The Kerberos and SSL protocols were studied, and the protocol was designed for our system.
- A detailed interface has been designed for both the Apache Web server, and the Kerberized Credential Translator.

References


Apache Software Foundation.
Apache web server http://www.apache.org.


P. Dousti.
Project minotaur: Kerberizing the web.
Software at Carnegie Mellon University.


Olga Kornievskaia.
Symmetric and Asymmetric Authentication.
PhD thesis, Computer Science and Engineering at University of Michigan, June 2002.


Olga Kornievskaia, Peter Honeyman, Bill Doster, and Kevin Coffman.
Kerberized credential translation: A solution to web access control.

Analysis of SSL Performance

Workload		Cost Of Operation		
CPU Freq.	Trace	Apache (ops)	Apache+SSL(ops)	RSA (%)
500 MHz	e-com	1370	147	58
500 MHz	data	610	149	23
933 MHz	e-com	2200	261	57
933 MHz	data	885	259	20

Throughput of the Apache Web Server