Chapter 5

RESULTS AND DISCUSSIONS
Chapter 5: RESULTS AND DISCUSSIONS

5. Results and Discussion 110

5.1 Authentication Phase 111

5.2 Authorization Phase 117

5.3 Number of Keys Used for Computation Analysis 122

5.4 Comparison of PKINIT and PKLK system 124
Chapter 5

RESULTS AND DISCUSSIONS

Standard PKINIT relies on client certificate, server certificate, digital signature, and message digest. Addition of digital certificates in a request requires additional parameters like certificate authority, certificate lifetime, certificate directory, and revocation list. Both the client and the KDC have a public-private key pair in order to prove their identities to each other over the open network. The intention of PKI infrastructure is to address issues of managing secret keys of a large number of clients and a secure authentication procedure by building a new trust model where KDC is not the first entity to identify the user. However, for scenarios where revocation or the suspension of certificates happens, the legislation of digital signatures and related PKIs becomes a difficult and challenging task.

We have proposed the PKLK model to address some of the issues related to Standard PKI. We have reduced the number of pre-authentication data elements used during AS_REQ. Thus, it keeps the message size between request and response as minimum as possible. PKINIT expects a dataset consisting of Auth-Pack, and trusted certificates. We have not used these parameters as our framework assumes that the client and the server are carrying a copy of each other’s certificates, which can be referred on demand from their database.
5.1 Authentication Phase

We have captured data that is being transmitted between the client and KDC using the WireShark packet analyzer tool for 3 scenarios.

Scenario 1:

This scenario belongs to basic Kerberos as mentioned in Table 5.1. A Login request was sent to this server using password based symmetric encryption. Figure 5.1 shows a request and response captured using the Wireshark tool during the authentication phase.

![Wireshark packet capture](image)

**Fig. 5.1: Basic Kerberos – Network Data Captured by the Wireshark Tool.**

Scenario 2:

This scenario belongs to PKINIT flow as mentioned in Table 5.1. A Login request was sent to the server user’s public key in request. The server encrypts a response using the user’s public key. Figure 5.2 shows the request and response captured using the WireShark tool during the authentication phase. Note that scenario 2 is not the exact simulation of PKINIT and we are only passing the public key and we will have to ensure that the certificate size is added to the request and response before we do a comparative analysis.
Scenario 3:

This scenario belongs to PKLK flow as mentioned in Table 5.1. A Login request was sent to the server using an encrypted timestamp by the user’s device key and a server encrypted response using the user’s public key. Figure 5.3, which shows the request and response, captured using the WireShark tool during the authentication phase.

An authentication request carries less data than response. This is because a response consists of the TGT and TGS session key, which is used in the second phase in all scenarios. PKINIT wraps the request and responses with a client certificate and a server certificate respectively. Hence, it carries the maximum size of bytes in both request and response. PKLK request is based on a similar theory as used in Kerberos, and it encrypts a timestamp.
using a device key instead of a user password, and hence, a PKLK request carries almost the same byte size as that of Kerberos. Response of PKLK eliminates the KDC certificate and thus, we see a reduction in size than PKINIT as shown in table 5.1.

**Table 5.1: Comparison of Authentication phase message Exchange Bytes.**

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Scenario</th>
<th>Server</th>
<th>AS_REQ (bytes)</th>
<th>AS_REP (bytes)</th>
<th>Total (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Basic Kerberos</td>
<td>Apache DS / standard KDC Server</td>
<td>277</td>
<td>557</td>
<td>834</td>
</tr>
<tr>
<td>2</td>
<td>PKINIT * (Pub Key only)</td>
<td>Apache DS / modified KDC for Public Key</td>
<td>602+699</td>
<td>846+1024</td>
<td>3171</td>
</tr>
<tr>
<td>3</td>
<td>PKLK</td>
<td>Apache DS Server / PKLK KDC Server</td>
<td>277</td>
<td>846</td>
<td>1123</td>
</tr>
</tbody>
</table>

**Overall Communication**

Figure 5.4 shows the communication summary graph. Each column shows the total number of bytes transferred in the authentication phase. This graph shows that the PKINIT protocol has the heaviest load of bytes being transferred on the network. However, basic Kerberos is carrying the least number of bytes.
Communication overheads using PKLK are much lesser than standard PKINIT.

Figure 5.5 illustrates payload distribution considering ten requests. It captures payload in Request and Response named as AS_REQ and AS_REP respectively. Results in the graph shows that the proposed PKLK works more efficiently in terms of payload transmission. The performance results also have been compared with existing PKINIT and basic Kerberos. The purposed scheme shows that there is significant improvement over PKINIT
Fig. 5.5: Payload Comparison for Ten Request and Response

Assuming basic Kerberos’s payload as baseline, the first line shows the representation for basic Kerberos, the second line about PKLK Kerberos, and the third line about PKINIT. The reason PKINIT is heavy on payload is because it relies on client certificate, server certificate, digital signature, and message digest. Having digital certificates in a request requires additional parameters like certificate authority, certificate lifetime, certificate directory, and revocation list.
Payload during Authentication in PKLK and basic Kerberos is the same. Public key components are not involved in the Authentication phase of PKLK and basic Kerberos. The device key, which works on the principal of a symmetric key system takes part in an authentication request, and hence payload for basic Kerberos and PKLK is the same while the public key component and digital signature data is added to payload, and hence payload in PKINIT is large and it goes on increasing when the number of requests are more as shown in Figure 5.6.

Authentication reply of basic Kerberos consists of Identification of Client, random key, and TGT while in PKLK authentication reply consists of all these things along with a device key and a random key encrypted with a private key of KDC. Hence, payload is slightly increased in comparison with basic
Kerberos. Authentication reply of PKINIT contains more data like certificate and information regarding a certificate. Hence, it carries more data as shown Figure 5.7.

Fig. 5.7: Payload Comparison in Authentication Reply

5.2 Authorization Phase Result Analysis

Authorization is not bound strongly with the Kerberos authentication protocol. If the user sends a valid SGT_REQ then the Key Distribution Centre generates a service ticket SGT and gives it back to the user. The Application Server is supposed to take authorization related validations.

In the situation where an organization wants to tightly control access for its critical services using the standard Kerberos system, the Application Servers do key authorization. But in standard Kerberos, the Application Server
comes into action in the last phase only. If an organization has a number of critical services then all of these services need to validate authorization checks at their end. In the scenarios where multiple services and servers are involved, it becomes difficult to maintain an authorization policy for critical services.

We overcome the above problem by enhancing the second phase of the Kerberos Authentication Protocol. Here, TGS and the Context Manager works collectively to validate the user’s location and performs additional checks. If the user’s location is within a defined zone, then only it is authorized to avail the services. This makes the second phase stronger, thus, addresses the organization’s critical need of centralizing authorization decisions for accessing services. Upon failing to validate the location parameter, TGS will not give back a service ticket SGT. If the SGT is not given then the requester is denied the intended service. In Standard Kerberos TGS returns the SGT to all legitimate users. Due to this, a ‘Replay’ attack is possible in phase III. In our proposed framework SGT is not returned to the user unless the user passes through an authorization validation. The SGT will not be available with the hacker and this makes it difficult to do a replay attack in phase-III of Kerberos.

**Comparison between proposed PKLK-Kerberos and PKINIT based authorization systems.**

Table 5.2 presents a detailed Comparison between proposed PKLK-Kerberos and PKINIT based authorization systems against various parameters.

**Table 5.2: Comparison of Authorization Phase between PKINIT and PKLK-**
Kerberos

<table>
<thead>
<tr>
<th>Parameters</th>
<th>PKINIT Authorization Phase</th>
<th>PKLK-Kerberos Authorization Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authorization Methods</td>
<td>Application Servers take part in the procedure.</td>
<td>Centralized in to Key distribution servers Domain.</td>
</tr>
<tr>
<td>Functions supported with TGS Server</td>
<td>TGS grants SGT to all the valid users. Only basic support is given.</td>
<td>TGS performs detailed authorization checks.</td>
</tr>
<tr>
<td>Dynamicity of authorization capabilities</td>
<td>No – No dynamic authorization.</td>
<td>Yes – System captures the user’s run time Location data securely. Thus, it provides dynamic authorization.</td>
</tr>
<tr>
<td>Application server dependency during Authorization</td>
<td>Dependent on AS. The Application Server takes a decision for validation of authorization checks in the third phase.</td>
<td>Does not depend on the Application Server. TGS and Location Manager collectively, can prevent users from granting SGT for critical services in phase II.</td>
</tr>
</tbody>
</table>

We have done a case study for an organization having three services TELNET, FTP, and HTTP as shown in Table 5.3.

**Table 5.3: Users and Access Grant Permission**

<table>
<thead>
<tr>
<th>Group</th>
<th>Number of Users</th>
<th>Access Grants for services</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP1</td>
<td>30</td>
<td>TELNET only</td>
</tr>
<tr>
<td>GROUP2</td>
<td>30</td>
<td>FTP only</td>
</tr>
</tbody>
</table>
These services are grouped under critical services. The organization wants to provide access for their remote users to avoid business impact. It also wants to restrict the access from a specific location boundary due to their company working policy related regulations.

**Table 5.4: Number of Tickets generated with the Kerberos based system.**

<table>
<thead>
<tr>
<th>Group</th>
<th>User Requests</th>
<th>SGT Tickets</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP 1</td>
<td>[GROUP1, TELNET] + [GROUP1, FTP]</td>
<td>30+30+30 = 90</td>
</tr>
<tr>
<td></td>
<td>+ [GROUP1, HTTP]</td>
<td></td>
</tr>
<tr>
<td>GROUP 2</td>
<td>[GROUP2, TELNET] + [GROUP2, FTP]</td>
<td>30+30+30 = 90</td>
</tr>
<tr>
<td></td>
<td>+ [GROUP2, HTTP]</td>
<td></td>
</tr>
<tr>
<td>GROUP 3</td>
<td>[GROUP3, TELNET] + [GROUP3, FTP]</td>
<td>30+30+30 = 90</td>
</tr>
<tr>
<td></td>
<td>+ [GROUP3, HTTP]</td>
<td></td>
</tr>
</tbody>
</table>

If all the users of GROUP3 in standard Kerberos V5 try to get a service ticket for the TELNET service, the KDC will return the SGT for the entire group because the system expects the Application Server to perform an authorization check and take decisions. This means for every request whether it is for TELNET, FTP or HTTP, the service ticket SGT is generated. The system doesn’t care that the requester belongs to GROUP1, GROUP2, or GROUP3. In all 270 tickets are generated in standard Kerberos V5 as shown in Table 5.4.

Standard Kerberos V5 does not have any provision for handling dynamic data like the location of the user. It is returning tickets SGT to all the successfully authenticated users.
### Table 5.5: Number of Tickets Generated With PKLK-Kerberos System

<table>
<thead>
<tr>
<th>Group</th>
<th>User Requests</th>
<th>SGT Tickets</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP 1</td>
<td>[GROUP1, TELNET] + [GROUP1, FTP] + [GROUP1, HTTP]</td>
<td>30+0+0 = 30</td>
</tr>
<tr>
<td>GROUP 2</td>
<td>[GROUP2, TELNET] + [GROUP2, FTP] + [GROUP2, HTTP]</td>
<td>0+30+0 = 30</td>
</tr>
<tr>
<td>GROUP 3</td>
<td>[GROUP3, TELNET] + [GROUP3, FTP] + [GROUP3, HTTP]</td>
<td>0+0+30 = 30</td>
</tr>
</tbody>
</table>

In the example shown, Table 5.5 is estimated to have a 66% reduction in tickets if we compare it with standard Kerberos V5. Figure 5.8 shows the ticket comparison graph.

![Ticket Comparison Graph](image)

**a)** Number of tickets used in other PKINIT system
b) Number of tickets used in PKLK system

**Fig. 5.8: Shows the Ticket Comparison Graph**

In our proposed PKLK-Kerberos, the user will issue SGT only upon successful authorization and user location verification. The total number of tickets generated for authentication and authorization in this case will be limited to the authorized users for authorized services only.

### 5.3 Number of Keys Used for Computation Analysis

Table 5.6 shows the different type of mandatory computations used in authentication by a server.
We have eliminated computational overheads like certificate verification by both client and server. We propose the maintenance of keys, certificate, and updates in LDAP as an offline activity outside the authentication phase.

The number of computations done in symmetric key encryption is very less compared to asymmetric key cryptography. Table 5.6 shows that the number of public keys used in PKINIT is three while for PKLK only one key is used, which will lead to a less number of computations. It improves the
computational efficiency of the system, which is a crucial factor for mobile devices.

![Number of Key Computations at Authentication Servers](image)

**Fig. 5.9: Number of Keys Used for Computations Graph.**

Figure 5.9 shows a graphical representation of the number of keys used in a different protocol.

### 5.4 Comparison of PKINIT and PKLK system

Thus, Kerberos can become the most preferred authentication and authorization mechanism

Protocol is enhanced for organizations supporting Mobile Users by:

- Integrating PKINIT, which eliminates password
- Enhancing Authorization Capabilities by allowing dynamic (context)
based decisions before sharing service tickets.

Table 5.7 shows the comparison of the PKLK system with Kerberos.

<table>
<thead>
<tr>
<th>ISSUE</th>
<th>Kerberos V5 System</th>
<th>PKLK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependency on User Password</td>
<td>YES – Because Keys are Derived from the Password.</td>
<td>NO – Completely Independent of Password.</td>
</tr>
<tr>
<td>Authorization Support</td>
<td>Minimal (Coarse Grain)</td>
<td>Can be fine grain</td>
</tr>
<tr>
<td>Support for Denial Mechanism During Granting Service Tickets</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Password Guessing, Dictionary Attacks</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Brute Force Attack</td>
<td>Yes</td>
<td>No*</td>
</tr>
</tbody>
</table>

**ADVANTAGES**

- Eliminates the need of a password based authentication system.
- Encrypts all communications to assure confidentiality.
- Gives single Sign on facility. With this property, a user logs in once and gains access to all systems without being asked to log in again at each of them.
- PKINIT implementation (Based on asymmetric cryptography) provides more security and scalability.
• Dynamic Authorization while accessing service request and actual service: This is a distinguishing feature and gives greater advantages to an organization as a mobile user will be roaming while accessing the services.

• Accessibility and availability of important services to mobile users while on the go.

LIMITATIONS

• Location validation accuracy cannot go up to meter level as it’s directly dependent on the accuracy of GPS units.

• Due to Network Latency and a slower authentication process (in some cases), the user may move out of an allowed region and can still get the service.

• Since all authentications are monitored and governed by a centralized KDC, if KDC is compromised then it will allow an attacker to impersonate any user.

• Strict time requirements, which mean the clocks of the involved hosts, must be synchronized within configured limits.

• Single point of failure:
  • Continuous availability of a central server is must.
  • This can be addressed by having multiple KDC servers and fallback authentication mechanisms.