A Remote Password Authentication Scheme for Multiserver Architecture Using Neural Networks

Li-Hua Li, Iuon-Chang Lin, and Min-Shiang Hwang

Presented By:
Derek J Corbett

Traditional Password Security

- Security Issues
- Doesn’t scale for multiserver environments

<table>
<thead>
<tr>
<th>ID</th>
<th>PW</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID₁</td>
<td>PW₁</td>
</tr>
<tr>
<td>ID₂</td>
<td>PW₂</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>IDₙ</td>
<td>PWₙ</td>
</tr>
</tbody>
</table>

ID and password stored in a table

<table>
<thead>
<tr>
<th>ID</th>
<th>F(PWᵢ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID₁</td>
<td></td>
</tr>
<tr>
<td>ID₂</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>IDₙ</td>
<td></td>
</tr>
</tbody>
</table>

Password encoded with a one-way hash
A New Authentication Scheme Using Neural Networks

- Multiple Servers
- System Administrator
- User

All Servers have the Same NN weights

Registration (PW)
Registration (ID)
Login (W,ID,v)

W = g^{PW^T} \mod p
T = Time stamp
ID = E_k(PW)

Registration Phase

- User freely chooses a password PW
- ID = E_k(PW)
- Train the Neural Network with the input of the user’s password and a value v which is the binary number for the expected output (e.g. 17 from 010001)
Network Specification

- Input: the password characters and the value v.
- Hidden layer neurons
- Output: If the system has \( m \) servers then the number of output units is \( m \).

<table>
<thead>
<tr>
<th>The training pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
</tr>
<tr>
<td>password</td>
</tr>
<tr>
<td>Kg</td>
</tr>
</tbody>
</table>

Authentication Phase

- Server Checks the Timestamp (This avoids replay attacks)
- Decrypts ID to get the PW checks this corresponds to the one encoded in W
- Pass PW and v through the Neural Network
- Check output meets the threshold and matches v
Network Architecture

- Backpropagation network (BPN)
- Sum-of-product network
- Hybrid sum-of-product network
- Supervised Learning models

![Diagram of a hybrid sum-of-product network]

Training

- System with 100 users and 6 Servers
- Training set has a total of 100 patterns
- The weights are randomly initialized
- Transfer function is the Sigmoid Function
- Learning rate of 0.5
- Trained till SSE reaches its minimum
Comparison of Architecture Performance

<table>
<thead>
<tr>
<th></th>
<th>BPN</th>
<th>Hybrid</th>
<th>S-O-P</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8.19</td>
<td>7.84</td>
<td>4.478</td>
</tr>
<tr>
<td></td>
<td>4.89</td>
<td>5.19</td>
<td>4.33</td>
</tr>
<tr>
<td>training times</td>
<td>1324 sec</td>
<td>1652 sec</td>
<td>599 sec</td>
</tr>
<tr>
<td># of units in hidden layer</td>
<td>24</td>
<td>48</td>
<td>24</td>
</tr>
<tr>
<td># of the weights</td>
<td>384</td>
<td>768</td>
<td>1224</td>
</tr>
</tbody>
</table>

Security

- Secrecy (The secret key k of the system and the users PW need to be kept secret)
- Nonforgability (discrete logarithms)
- Replay Resistance
**Strength**

- Essentially a complex hashing function
- Neuron output equates to which server you can access
- Very Secure
- All users are forced to have different passwords

**Weakness**

- The network needs to be retrained every time a new user joins the system or any time someone wants to change their password
- If the number of servers increases, the number of interconnect weights grows rapidly
Future

- Trial using smaller less interconnected networks to reduce space complexity see if they give similar results
- A centralized Neural Network based authentication server that just holds NN weights.

References

- A Remote Password Authentication Scheme for Multiserver Architecture Using Neural Networks; Li-Hua Li, Iuon-Chang Lin, and Min-Shiang Hwang; IEEE Transactions Neural Networks, Vol.12, No.6, Nov. 2001