SSL BEAST ATTACK
Time to Review SSL Security

Exploring Digital Signature certificate

Possible Fuzzy Logic implementation in Governance

A .Net library to code Intelligent Program
When the Chinese intrusion along the McMahon line was hitting the national attention, Indian intrusion into the security perimeter of Facebook attracted the international focus. Mr Arul Kumar from India presented the proof of concept to delete any photograph from the server of Facebook proved that system need to be developed with much more security and robustness before taking the usability to the next level. Security is always the main concern with the invention of new technologies. When cloud computing came into the picture related vulnerabilities also became the research topic. SSL become the synonym of security and developer trusted on SSL completely for encryption during data transmission. But SSL also do come with new security holes which were primarily ignored but now it became exposed to the world. In this issue of FocusIT we explained the SSL BEAST attack which is one of the most recent type security threats that 85% of the current web portals are suffering from this and presented a small analytical as well as comparative study of web portals against the SSL BEAST attack.

Fuzzy logic the most heard word for any engineering graduates has rarely been used in e-governance. People use fuzzy logic mainly for research simulation using MATLAB, but in this issue we explained fuzzy logic in more simple and practical way with e-governance related examples to get a breakthrough in implementing this magical theory in e-governance area.

Writing an Artificial Intelligence program need very high level understanding about the AI algorithms as well as the concepts behind them but implementing those already developed algorithm does not need to be build from scratch instead AI library is available for all popular programming platform. We have presented information about few library files for Dot Net as well as JAVA developers.

I like to thanks all the contributors and supporter for their selfless service for this research forum and wish all reader to get at least a bit useful information from this FocusIT issue which will justify our labour.

Readers your feedback is most valuable for us. We have started our journey only with a hope to disseminate knowledge and create a culture of research in cyber world. Please send your feedback or become our contributing author.

E-Mail: nirmali@mlab.co.in
1. SSL BEAST Attack: Ultimate Time to Review your SSL Security ..................................................

2. Exploring Digital Signature Certificate ......................................................................................

3. FUZZY Logic and its possible scopes in e-Governance ..............................................................


5. Implementing Fuzzy Logic in e-governance using Aforge.net Framework ............................

6. News & Events in Cyber Security Area .....................................................................................

7. News & Events in Artificial Intelligence Area ............................................................................
Ever since Netscape Communications invented the Secure Sockets Layer Protocol (SSL) in 1994, it has become an industry standard for securing communication sessions over an unsecured network, such as the Internet, over years. Transport Layer Security (TLS) Protocol came up as a new edition to its predecessor, the Secure Sockets Layer Protocol (SSL). However, off late a new attack called BEAST, Short for Browser Exploit Against SSL/TLS, has taken the whole world by storm. BEAST is an exploit first, revealed in late September 2011 that exposes the weaknesses in cipher block chaining (CBC) to exploit the Secure Sockets Layer (SSL) protocol. The CBC vulnerability can enable man-in-the-middle (MITM) attacks against SSL in order to silently decrypt and obtain authentication tokens, providing hackers with access to the data passed between a Web server and the Web browser accessing the server.

In order that a BEAST attack can take place, there are three conditions that have to be met with:

1. JavaScript or applet injection into the same origin of the web site
2. Network sniffing of the connection must be possible
3. A vulnerable version of SSL must be used which is using a block cipher

The BEAST attack that was demonstrated for the first time used a flaw in Java’s Same Origin Policy (SOP). According to the same origin policy two web pages are considered to have the same origin if the protocol, port (if one is specified), and host are the same for both pages. And if this policy holds in case of any two pages those are allowed to access each other's methods and properties with no specific restrictions. However this is not the only exploit that allows BEAST; there may be other weaknesses within the website that allows including malicious content within the Same Origin, for instance weaknesses within a file upload feature.

1.0 Some of the possible exploits

If it is possible to inject content within the same origin policy, whether through a browser bug or otherwise, there is already a large array of attacks that can be performed; and once page content can be accessed, any action can be performed that the victim user could do and the cookies can be stolen as well as sessions can be hijacked. Even a new login prompt can be displayed that tricks the user into sending their credentials to the hacker. And all of this can be done without needing to sniff the network traffic.

1.1 HTTPOnly Cookies

So if all this can already be done, where does actually BEAST attack comes into picture. Well there are a number of ways in which a developer can increase the adding an additional layer of protection. One example could be the HTTP Only property that can be set on cookies. With this property, a browser will not allow applets or JavaScript to gain access to the cookie value. This can prevent session hijacking. If the plaintext of the HTTPS session can be extracted, then one can gain access to the HTTP Only. Therefore in terms of risk, the BEAST attack is akin to not setting the HTTPOnly property on cookies.

1.2 Rich Internet Applications

Rich Internet Applications (RIA) are much more similar to desktop applications, providing a richer array of functionality which makes many more complex tasks much easier and possible to develop and deliver from a web site. RIAs include Java applets, Flash, and Silverlight.

In order for an RIA such as a Java Applet to be injected into the page or site, the RIA must appear as if from the same origin as the website. This is
SSL BEAST Attack: Ultimate Time to Review your SSL Security

September, 2013

typically performed by using a browser flaw in the Same Origin Policy or vulnerability within the website that allows it to be uploaded and hosted from within the same domain. This can then either be injected into a pre-existing page, for example using Cross Site Scripting, or a custom page to load it can be used.

It also states that a flaw in the SOP will still prevent access to existing cookies. Existing cookies (with the exception of ones with the HTTPOnly flag) are accessible by all web content within the SOP, whether by an SOP flaw or not. All cookies are of course accessible once the channel’s plaintext has been accessed without the need for a network sniffer.

1.3 JavaScript

It appears to be possible to perform a BEAST attack using HTML5 WebSockets (v.76 and lower) in Safari. It is not known whether there are other JavaScript mechanisms, as a pure binary socket is needed, however changes to the specification or new features may in the future introduce more mechanisms. Using a feature such as WebSockets to perform a BEAST attack could result in the ability to use a Cross-Site Scripting (XSS) vulnerability to inject the BEAST agent. This is especially concerning given the prevalence of XSS.

2.0 Safety over TLS

Though initially BEAST was termed as a theoretic attack only, after the practical demonstration of BEAST which exposed the weakness of TLS 1.0 and the earlier SSL versions, immediately TLS 1.1 and subsequently TLS 1.2 got introduced. However, the issue still remains vital as many browsers do not support TLS 1.1 or 1.2. This includes all versions of IE on Windows XP. Those browsers that do support TLS 1.1/1.2 have it turned off by default. The reason it is off is because older web servers, which account for a large portion of the internet, are unable to handle a TLS 1.1/1.2 connection, and due to the error condition that takes place, the connection won’t downgrade to an older protocol.

Therefore not only will most of the users be unable to access a web site using TLS 1.1 or 1.2, but those that can, will not have it enabled. Those that enable it will probably be stuck in a position of needing to turn it on and off depending on which website they want to access. The following table shows the browsers that support TLSv1.1, and the percentage of users

<table>
<thead>
<tr>
<th>Browser</th>
<th>Percentage</th>
<th>TLSv1.1 Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Explorer</td>
<td>56.49%</td>
<td>Uses underlying OS</td>
</tr>
<tr>
<td>Firefox</td>
<td>20.45%</td>
<td>No</td>
</tr>
<tr>
<td>Chrome</td>
<td>11.87%</td>
<td>No</td>
</tr>
<tr>
<td>Safari</td>
<td>10.13%</td>
<td>Uses underlying OS</td>
</tr>
<tr>
<td>Other</td>
<td>0.64%</td>
<td>N/A</td>
</tr>
<tr>
<td>Opera</td>
<td>0.43%</td>
<td>Yes - Since Presto v2.2</td>
</tr>
</tbody>
</table>

3.0 Fixes for TLS 1.0

It looks like it. The major vendors of both browsers and server-side technologies have all announced that they are working on a patch. In some cases, patches have already been released. This appears to be mainly a trade-off between finding a solution quickly, and minimising compatibility issues.

For example, OpenSSL based servers (e.g. Apache) can make use of an option to prevent the attack, however it is incompatible with IE6 which still accounts for around 2.7% of users.

Microsoft as of yet have not released a patch for their servers, and have not provided details on which versions and products will be patched.

4.0 Ensuring the security

Due to the compatibility issues mentioned above, you are going to have difficulties in enforcing a secure HTTPS connection. A trade-off between usability and security is needed.

4.1 Upgrade

Within a controlled environment, for example an internal network, although a potentially large task, it may be possible to upgrade all users and servers to products that support TLS 1.1 or 1.2 only. If already upgraded, it should be possible to roll out a policy change. However, be aware that users may have difficulties accessing some external sites, and these may be sites needed for their work. A secondary browser could be supplied for this purpose.

It is however, a worthy exercise to upgrade your servers to provide support of TLS 1.1 and 1.2, in
addition to the older versions. This allows one to provide those users who are security aware and proactive, with a secure environment and is likely to protect you against other attacks that might come to light.

If one can throw usability to the wind and limit the server to TLS 1.1 and 1.2 only, then it can be done in such a way as to allow for a friendly user error message to be presented to the user. This can be used to educate the user.

### 4.2 Getting Tested & Fixing Security Flaws

Although the demonstrated attack utilises a browser bug, there is the potential for other methods to be employed to inject the BEAST agent into the browser. Some of these may utilise flaws in the application, and thus it is important to ensure a high security standard is being adhered to.

There are also a number of areas in which the use of session hijacking can be reduced or made more complex, for example:

- Transferrable session prevention – prevent the ability to use a session token from a different IP and a different user agent, to increase complexity (this won’t prevent an attack)
- The existence of a logout button, which functions correctly
- Session timeout set to a sensible time and functioning correctly
- Regenerating a new and unique cookie value per session
- One-time passwords
- A secure HTTPS configuration

### 5.0 Analysis of some web-sites using SSL/TLS

There are some popular online and offline tools available which can analyse the strength of any website in public domain over https. QUALYS SSL Labs (https://www.ssllabs.com) is one such online tool which provides a comprehensive report on the strength of the TLS/SSL certificate and possible vulnerabilities.

#### 5.1 Reports from QUALYS SSL Labs

The images below depict a comparative analysis that has been made on a few well known https sites.

Exploring the e-procurement site of Karnataka it has been observed that it is still in SSL 2.0 only and hence remains most
As we can see from the image above the mail.gov.in site has been still found to be running on TLS version below 1.2 and remain on threat.

Due to outdated TLS 1.0 and CBC based TLS_RSA_WITH_AES_128_CBC_SHA this site is SSL BEAST vulnerable and rate “F” in SSL Lab.
It is found that although IRCTC is getting “A” rating from the SSL Lab but still it is vulnerable to SSL BEAST Attack due to the use of TLS_RSA_WITH_AES_128_CBC_SHA which is CBC based encryption algorithm. This tool says the same thing in Protocol details tab of the report.

The online SBI site though in version below TLS 1.2 and using TLS 1.0 but due to its strong configurations and encryption algorithm remains secure as per SSL Labs.
The most popular site Facebook has already moved to TLS 1.2 and remains secure.

Although it is tested on 100+ web portals both from government domain as well as commercial industry leaders but it is practically impossible to make those available on limited space of this article. But still for the curiosity of the reader it is better to mention that SSL certificate used in gmail, incometaxindiaefiling.gov.in are secure from SSLBeast Attack, whereas according to the SSLLAb report hotmail,

5.2 Reports from TestSSLServer.exe

TestSSLServer.exe is another such testing tool from http://www.bolet.org which can be used in command line for analysis of SSL based website to check the security level. There is also TestSSLServer.jar file available to test SSL certificates from Linux bases machine. Some of the web sites tested using SSLab are also got tested using this TestSSLServer.exe tool to project the correctness of each other. This tool can be downloaded and used from command prompt. The images below show some of the reports generated by the tool.

onlinesbi.com
SSL BEAST Attack: Ultimate Time to Review your SSL Security

September, 2013

---

gmail.com

---

facebook.com

---
A detail study has been done by Mr. Arun Verghese on both Government as well as Commercial categories of web portals using SSL. This study presents facts and figures which supposed to make security provider be more concern. The report prepared is unbiased and used two different popular third party SSL Analysis tools (SSL Lab online Tool and TestSSLServer.exe) to carry out the study so that cross checking can be performed at any point of doubt.
To understand the penetration capability of the SSL level security vulnerability a detail study has been done and an comparative figure has been extracted to project that commercial service provider are more conscious on SSL level security vulnerability. The study has been done by taking many parameters of SSL certificate to understand the vulnerability but due to the space constrain only the important parameter for this context have only been presented in this report.

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Name</th>
<th>URL</th>
<th>SSL BEAST Vulnerable or not</th>
<th>CRIME vulnerable or not</th>
<th>SSL LAB Rating</th>
<th>Expired</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aadhar Portal (UID)</td>
<td><a href="https://portal.uidai.gov.in">https://portal.uidai.gov.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>E-Procurement, Govt Of Karnataka</td>
<td><a href="https://eproc.karnataka.gov.in">https://eproc.karnataka.gov.in</a></td>
<td>No</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>e-Samarth</td>
<td><a href="https://esamarth.gov.in">https://esamarth.gov.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>IRCTC</td>
<td><a href="https://www.irctc.co.in">https://www.irctc.co.in</a></td>
<td>Yes</td>
<td>No</td>
<td>A</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Traffic Police - Hyderabad</td>
<td><a href="http://www.htp.gov.in">http://www.htp.gov.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>NIC Email</td>
<td><a href="https://mail.gov.in">https://mail.gov.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Andhra Pradesh Online</td>
<td><a href="https://secure.aponline.gov.in">https://secure.aponline.gov.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>Indian Visa Online</td>
<td><a href="https://indianvisaonline.gov.in">https://indianvisaonline.gov.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>Data Portal of India</td>
<td><a href="https://data.gov.in">https://data.gov.in</a></td>
<td>Yes</td>
<td>Yes</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>Tenders India</td>
<td><a href="https://tenders.gov.in">https://tenders.gov.in</a></td>
<td>No</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>11</td>
<td>Vahan Portal</td>
<td><a href="https://vahan.nic.in">https://vahan.nic.in</a></td>
<td>Yes</td>
<td>Yes</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>NIC Certifying Authority</td>
<td><a href="https://nicca.nic.in">https://nicca.nic.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>13</td>
<td>Epayment Gateway (India)</td>
<td><a href="https://gepg.nic.in">https://gepg.nic.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>14</td>
<td>Commercial Taxes, Govt Of Karnataka</td>
<td><a href="https://vat.kar.nic.in/">https://vat.kar.nic.in/</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>15</td>
<td>Income Tax efiling</td>
<td><a href="https://incometaxindiaefiling.gov.in/">https://incometaxindiaefiling.gov.in/</a></td>
<td>No</td>
<td>No</td>
<td>B</td>
<td>No</td>
</tr>
<tr>
<td>16</td>
<td>Tax Information Network</td>
<td><a href="https://tin.tin.nsdl.com">https://tin.tin.nsdl.com</a></td>
<td>Yes</td>
<td>No</td>
<td>A</td>
<td>No</td>
</tr>
<tr>
<td>17</td>
<td>National Securities Depository Ltd</td>
<td><a href="https://nsdl.co.in/">https://nsdl.co.in/</a></td>
<td>Yes</td>
<td>No</td>
<td>A</td>
<td>No</td>
</tr>
<tr>
<td>18</td>
<td>Open Technology Center</td>
<td><a href="https://portal.otc.nic.in">https://portal.otc.nic.in</a></td>
<td>Yes</td>
<td>Yes</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>19</td>
<td>quickSMS</td>
<td><a href="https://msgapp.nic.in/">https://msgapp.nic.in/</a></td>
<td>Yes</td>
<td>Yes</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>20</td>
<td>LIC India</td>
<td><a href="https://www.licindia.in/">https://www.licindia.in/</a></td>
<td>No</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>21</td>
<td>Oversean Citizenship of India</td>
<td><a href="https://passport.gov.in/">https://passport.gov.in/</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
<tr>
<td>22</td>
<td>ePOST Office</td>
<td><a href="http://www.epostoffice.gov.in">www.epostoffice.gov.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>Yes</td>
</tr>
<tr>
<td>23</td>
<td>Online Filing of Patents</td>
<td><a href="https://ipindiaservices.gov.in">https://ipindiaservices.gov.in</a></td>
<td>Yes</td>
<td>No</td>
<td>F</td>
<td>No</td>
</tr>
</tbody>
</table>

Report Prepared by: Arun Varghese as on 19/09/2013

**Category of Government sites:**

- Number SSL certificates in portals tested: 23
- Number of SSL certificate in portal found to be SSL BEAST vulnerable: 19
- Number of Vulnerable sites in percentage: \((19/23)*100 = 82.6\%\)
This phase of testing focused on various popular commercial and financial institutions for which security should be utmost priority and the report has been presented below after truncating the extra testing parameters.

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>Google India</td>
<td><a href="https://www.google.co.in">https://www.google.co.in</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>25</td>
<td>Facebook</td>
<td><a href="https://www.facebook.com">https://www.facebook.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>26</td>
<td>Twitter</td>
<td><a href="https://twitter.com">https://twitter.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>27</td>
<td>Vodafone</td>
<td><a href="https://www.vodafone.in">https://www.vodafone.in</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>28</td>
<td>Canara Bank</td>
<td><a href="https://www.canarabank.in">https://www.canarabank.in</a></td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>29</td>
<td>OnlineSBI</td>
<td><a href="https://www.onlinesbi.com/">https://www.onlinesbi.com/</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>30</td>
<td>Union Bank</td>
<td><a href="https://www.unionbankonline.co.in/">https://www.unionbankonline.co.in/</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>31</td>
<td>Bank of India</td>
<td><a href="https://starconnectcbs.bankofindia.com">https://starconnectcbs.bankofindia.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>32</td>
<td>ICICI Bank</td>
<td><a href="https://infinity.icicibank.co.in">https://infinity.icicibank.co.in</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>33</td>
<td>Axis Bank</td>
<td><a href="https://www.axisbank.co.in/">https://www.axisbank.co.in/</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>34</td>
<td>Indian Bank</td>
<td><a href="https://www.indianbank.net.in/">https://www.indianbank.net.in/</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>35</td>
<td>BillDesk</td>
<td><a href="https://payments.billdesk.com">https://payments.billdesk.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>36</td>
<td>Ebay</td>
<td><a href="https://signin.ebay.com">https://signin.ebay.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>37</td>
<td>Amazon</td>
<td><a href="https://www.amazon.com">https://www.amazon.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>38</td>
<td>Flipkart</td>
<td><a href="https://www.flipkart.com">https://www.flipkart.com</a></td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>39</td>
<td>Infibeam</td>
<td><a href="https://www.infibeam.com">https://www.infibeam.com</a></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>40</td>
<td>Makemytrip</td>
<td><a href="https://www.makemytrip.com">https://www.makemytrip.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>41</td>
<td>Redbus</td>
<td><a href="https://www.redbus.in">https://www.redbus.in</a></td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>42</td>
<td>itunes</td>
<td><a href="https://itunes.apple.com">https://itunes.apple.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>43</td>
<td>Gmail</td>
<td><a href="https://mail.google.com">https://mail.google.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>44</td>
<td>Yahoo Login</td>
<td><a href="https://login.yahoo.com">https://login.yahoo.com</a></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>45</td>
<td>Paypal</td>
<td><a href="https://www.paypal.com">https://www.paypal.com</a></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>46</td>
<td>LinkedIn</td>
<td><a href="https://www.linkedin.com">https://www.linkedin.com</a></td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Report Prepared by: Arun Varghese as on 19/09/2013

**Category of Commercial sites:**

**Number SSL certificates in portals tested:** 23  
**Number of SSL certificate in portal found to be SSL BEAST vulnerable:** 5  
**Number of Vulnerable sites in percentage:** \((5/23) \times 100 = 21.7\%\)

**Concluding thoughts**

As the study done by Arun clearly represents that vital commercial and financial organization are patching up their certificate problem either migrating to TLS 1.2 or by using strong configuration so it is clearly understandable to need of the review of our SSL certificate. Government web portals using SSL certificate to avoid Man-In-The-Middle sniffing of data should now be in mode to check the strength of their certificate in post BEAST era.

The BEAST attack utilises the flaw in TLS/1.0. It is observed that the tests performed using both the tools are yielding same results and as suggested by those it can be inferred that the need of the hour is definitely a shift to TLS 1.2 version.
In the last 3 weeks, download hits of TOR (The Onion Routing project uses for anonymous browsing) and Open PGP (Pretty Good Privacy – Cryptography algorithm for personal communication) increasing tremendously in higher rate. Countries like India planning to ban US/ Other Country based internet services like email, video calling, messenger and etc., for any official communication. Why this kind of sudden aversion created in the cyber world; not even happened in the last decade. All this because of Snowdream whistle blowing and the special project of US government intelligence agency PRISM to access the IT Giant’s service data like Google, Yahoo, Microsoft, Oracle & etc., Now, everyone keeps their hope on the cryptography to keep our privacy and safe our data from others interests.

As security credentials like user name and passwords, we used in the conventional programming left us naked in the cyber warfare. Security is a major concern for Web users as well as service providers. We need to be equipped with bunch of latest gadgets like OTP hardware tokens, Mobile based OTP (One Time Password), MAC based Challenge & Response system etc. In this series, Digital Signature Certificates also plays very crucial role in the security of the system. This digital certificate solely sits on the top of the Public cryptography; it gives more privacy compared to the private cryptography, because of their mathematically bound cryptographic key pair in their crypto process. After commencement of IT Act 2000, Digital Certificates and Signatures are authenticated entities for the electronic documents and legally valid. It has returned with few changes to fulfil other vital industry requirements, another IT (Amended) Act 2008 introduced. The digital signing is a mechanism for certifying the origin and the integrity of electronically transmitted information. In the process of digitally signing, additional information; sequence of numbers so called a digital signature is added to the given document and calculated using the contents of the document using a private key. At a later stage, this information can be used to check the origin of the signed document and also assures CIA. Before, we will be diving into the pool of Digital Certificates usage and application in real time projects; we may see some of the basics constitute this system as follows.

The Public Key Infrastructure (PKI) provides the architecture, organization, techniques, best practices and procedures that support by means of digital certificates. The application of public key cryptography is for the purposes of secure information exchange over insecure transmission media.

The digital certificates bind a particular public key with a particular person. They are issued by special kind of authorities called as certification authorities (CA) in strict security precautions, which guarantee their authenticity. We can think of the digital certificates as electronic documents, certifying that a given public key is property of a given person. In practice for the purposes of the digital signature the most widely used certificates are X.509 certificates. X.509 is a widely accepted standard for digital certificates. An X.509 digital certificate contains the public key of a given person along with the private data about this person (name, organization, PAN and so on), information about the certification authority that has issued the certificate, validity period, information about the used cryptographic algorithms and other various details.

The certification authority (CA) is an institution entitled to issue digital certificates and to sign them with its own private key. The purpose of the certificates is to confirm that a given public key
is property of a given person, and the purpose of the certification authorities is to confirm that the given certificate is valid and can be trusted. Every certification authority possesses a certificate and a corresponding private key, with which it signs the certificates it issues to its clients. A certification authority can be at the top level (top-level certification authority; root CA) or at some subsequent level. Top-level certification authorities issue themselves a certificate at the beginning of their activity and sign it with the same certificate; those certificates also known as self-signed certificates. These certificates are called Root certificates. To cater the services for DC, top-level certification authorities opened local certification authority/ sub CA. As example CCA is Indian Government controlled certifying authority established as Root Certifying Authority for India to issue Certificate as per IT Act 2000. NICCA, Safescrypt, TCS are few of the local certification authorities ensured by CCA. To cater the need of DS at each regional level throughout the India, NICCA setup NICCA Sub CA.

A **certification chain** is a sequence of certificates in which each certificate is signed by the one after it. At the beginning of the chain usually stands some certificate issued to an end-client, and at the end of the chain is the Root certificate of some certification authority (CA). In the middle of the chain stand the certificates of some intermediate certification authorities. A certificate at the beginning of the certificate chain is only trusted, if the certificates signed by its next level certificates are successfully verified. At the end of the chain is always trusted root certificate, it always unconditionally trusted root certificate. Every software deals with digital certificates maintain the list of trusted root certificates. Mozilla, Internet Explorer, Chrome like browsers always keep the bunch of trusted certificates in their certificate store.

In the process of certificate verification, certificates also verified for revocation due to loss of certificate/some reasons, through the **Certificate revocation List** (CRL) mentioned along with the certificates. Those revoked certificates and their signatures after the revoked time becomes invalid. CA published periodically the list of revoked/temporary disabled certificates before expiry date and signed the list.

**PKCS** is a group of public-key cryptography standards devised and published by RSA Security Inc under the name PKCS #1 to #15, each one promotes the use of the cryptography techniques for making possible secure information exchange on the Internet using a public key infrastructure (PKI). The following are few of the very important standards.

- PKCS #1-RSA Cryptography Standard
- PKCS #7-Cryptographic Message Syntax Standard for sign and/or encrypt messages
- PKCS #11-Cryptographic Token Interface also known as "Cryptoki". An API defining a generic interface to cryptographic tokens (see also Hardware Security Module). Often used in single sign-on, Public-key cryptography and disk encryption systems
- PKCS #12-Personal Information Exchange Syntax Standard defines a file format commonly used to store private keys with accompanying public key certificates, protected with a password-based symmetric key. PFX is a predecessor to PKCS #12.
- PKCS #15-Cryptographic Token Information Format Standard defines a standard allowing users of cryptographic tokens to identify themselves to applications, independent of the application's Cryptoki implementation (PKCS #11) or other API. RSA has relinquished IC-card-related parts of this standard to ISO/IEC 7816-15.

**Protected keystore** can contain three kinds of elements: certificates, certification chains, and private keys. As the information stored in protected stores is confidential due to security considerations, it is accessed using password: known as PIN. As per PKCS and IT ACT, Indian CA provides two certificates. The needs of separate certificate for encryption and signing, due to underlying mathematics are same and reverse in nature.

If an attacker can convince a key holder to sign an unformatted encrypted message using the same key then she/he gets the original (The Chinese Remainder Theorem). Due to the confidentiality of signing...
Explore the Digital Signature Implementation in a generic way

CAPICOM and other ActiveX controls are used to address the solution only for IE browser and windows based, whereas DOMCrypt is the Mozilla extension for accessing the keystores entirely through Javascript and is not open to other browsers. Currently, only feasible solution for this is Java Applets, due to the pervasive use of JVM made it as common requirement in every computing devices ranging from smart phones to computers. Signed Java Applets have special permission to access the local resources that made possible to generate the signatures at the client end, send it for storing and verifying at the server end.

There are several standards for storing X.509 digital certificates. Most frequently, **ASN.1 DER** encoding is used, in which the certificates are stored in files with a **.CER** extension (or more rarely with **.CRT** or **.CA** extensions).

A CER file contains a public key, information about its owner and a digital signature of some certification authority certifying the public key really belongs to a person. The certification authorities distribute their Root certificate in CER files. A CER file can be stored in binary format or text format, encoded with Base64.

**Digital Signing of the Message**

1. Hashing: Apply a hashing algorithm (MD5, SHA1, ...) to convert the message into a hash value.
2. Signing: Use the private key to create a digital signature.
3. Verification: Compare the hash value with the received hash value and the digital signature to verify the message.

**Verifying Digital Signatures**

1. Decryption: Use the public key to decrypt the digital signature to obtain the hash value.
2. Comparison: Compare the decrypted hash value with the original hash value to verify the signature.

Cryptography is available as JCA (Java Cryptography Architecture) specification that gives the programmers a standard way to access cryptographic services, digital signatures, and digital certificates. From an architectural point of view, JCA is designed to allow different implementations of various services from different software vendors. Such implementations of cryptographic services are called cryptographic service providers. If we need advanced cryptographic algorithms and features, use Bounty Castle (BC) as service provider. BC is open source project consists of all sort of latest cryptographic algorithms and features.

First, we see the basic java program to generate the sign for given message with the USB based security tokens. We are getting DSC security tokens/smart cards through NICCA. The basic requirement for the program is jdk & jre 1.6 32bit (64bit is not supported, because of sunpkcs11.jar not available) and drivers for tokens properly installed as per the instruction given at NICCA site. In windows, accessing the keystore is very easy when compared to Linux machines. Windows-MY is a common keystore maintained in windows platform for self use.

```java
SunMSCAPI providerMSCAPI = new SunMSCAPI();
Security.addProvider(providerMSCAPI);
KeyStore ks = KeyStore.getInstance("Windows-MY");
ks.load(null, null);
String alias = (String) ks.aliases().nextElement();
PrivateKey obPotKey = (PrivateKey) ks.getKey(alias, null);
Signature obSign=Signature.getInstance("SHA1withRSA");
obSign.initSign(obPotKey);
byte[] arrDataBytes=“Hello!Welcome to Mlab,FocusIT”.getBytes();
obSign.update(arrDataBytes);
byte[] signedDataBytes = obSign.sign();
```

The usage of Digital Signature in web programming is very essential for enhanced security and privacy of data in open space. But the problem of accessing client end protected keystore is very cumbersome and there is no open standard available.
The signing process requires the private key; we may get the reference to the private key as mentioned in the code. The real encryption is handled by the drivers at the hardware token itself. The signature for the given message “Hello! Welcome to Mlab, FocusIT” is available in the signedDataBytes. The keystore is accessed by using PIN that can be captured by the password dialog prompted by the native driver of the token. The signing algorithm is SHA1withRSA implies that, SHA1 is used for generation of message digest and RSA encryption used for signing the hash/digest of the message. The signed data is in raw format, is not suitable to send to server through http because of the characters presents in the format may interpret as commands over stream like end of message etc., So, it always advisable to encode the signed data into Base64, before sending it to server and vice-versa. Base64 encoding may be done by our own encoder if we used the verification process in the same platform in the server end, if it's different system use apache codes for encoding and decoding. For example, we suppose use the digital signature in PHP, in client end we use applets to sign the message and send it back to server to validate the signature. In that case, we need to use apache Base64 codes to encode the signature in base64 at client end, PHP base64_decode() in the server end. Because, encoder and decoders are reversible in nature. If, we wrote our own base64 codes in java, same codes we need to use in other end of the system. So, it’s always advisable to use common standards like apache common codes api for encoding and decoding.

As I mentioned in the previous paragraph, in MS Windows, MS Cryptographic API takes care of all sort of process of signature generation through the native drivers for the tokens. Whereas, in linux, we need to register the SunPKCS11 provider along with the driver files as available in shared library object (.so) file. As the same mechanism, we can access the token in windows also by using DLL file. But, In the case of Windows-MY keystore, MSCAPI do the necessary actions in behalf of us.

```java
Security.addProvider(providerPKCS11);
KeyStore ks = KeyStore.getInstance("PKCS11");
ks.load(null, null);
```

The rest of the signing process is common on the both platforms. If we scrutinize the code, we got to know that the signature generation process entirely done inside the token and the resultant signature only comes out. The entire process is handled by the token native driver; Security Providers just forwards the calls to the native drivers. The process of verifying the signature is same as above with the conditional statement.

```java
PublicKey obPubKey = (PublicKey) ks.getPublicKey(alias, null);
Signature obVerify = Signature.getInstance("SHA1withRSA");
obVerify.initVerify(obPrtKey);
byte[] arrDataBytes = “Hello! Welcome to Mlab, FocusIT".getBytes();
obVerify.update(arrDataBytes);
boolean verStatus = obSign.verify(signedDataBytes);
```

The boolean variable verStatus contains the verification of the signature. if it's true, signature is valid and verified and False implies invalid signature. Invalid signature may be due to the following reasons.

- If the digital signature is adulterated (it is not real) and is decrypted with the public key, the obtained original value will not be the original hash-value of the original message but some other value.
- If the message was changed (adulterated) after its signing, the current hash-value calculated from this adulterated message will differ from the original hash-value because the two different messages correspond to different hash-values.
- If the public key does not correspond to the private key used for signing, the original hash-value obtained by decrypting the signature with an incorrect key will not be the correct one.

To access the keystores through web programming, we need the signed applets with special permission. This can be done by creating the java.applet.policy file with the following content.

```java
grant {
    permission java.security.AllPermission;
};
```
Jar files can be created by our favorite IDE like Netbeans, Eclipse etc., and sign the same by our security token, where our signature certificate presents. The command to do the signing as follows.

```
jarsigner -keystore NONE -storetype PKCS11 -providerClass
sun.security.pkcs11.SunPKCS11 -providerArg java-pkcs11.cfg <jar
file> "D MADAN PRABHU-Sign"
```

jarsigner command is available in the jdk bin folder, change our path in your command prompt and run the above command. Create the java-pkcs11.cfg file with the same content, we used in the last program such as native driver details and name with slotIndex.

```
name=eToken
library=c:\windows\system32\aetpkss1.dll
```

While signing your jar with your certificate, jarsigner may throw the error “Certificate Chain is not verified”. To avoid this problem, we need to import the certificate into the java keystore available in the cacerts file present in the jre path of security folder. This java keystore contains all ca certificates for certification verification by running the following command. Before, running this command, you need to take the signing certificate in .cer format through backup option of driver

```
keytool -list -keystore ../../jre7/lib/security/cacerts -import -
file <File Name>
```

Now, Your Applet is ready for use. First, upload the signed applet along with the necessary libraries in the server and use it by applet tag in your web programming. Because of calling javascript function inside signed applet, other ways of deployment of applet by using deployJava.js and Java Network Launch Protocol (JNLP) don't work. Communication and access the DOM objects between applets by using plugin.jar available in jre/lib folder. In addition to the PARAM tags, don't forget to set the MAYSCRIPT attribute on your applet tag in order to explicitly enable java-js communication which normally is disabled by default for security reasons. The following one is applet tag for enrolment of certificate by KeyRegister.jar file.

```
<object width="100%" height="150" type="application/x-java-applet;version=1.6" archive="KeyRegister.jar"
classid="java.gov.nic.keyupload.gui.KeyRegister.class">
   <param value="true" name="MAYSCRIPT">
   <param id="Any_2" name="FEED_1" value=""/>
   <param id="Any_3" name="FEED_2" value="1"/>
   <param id="Any_4" name="FEED_3" value="ahR0cDovLzEwLjE4NC4xMjguNy9EU0MvZW5yb2xsQ2VyGlmaWNhdGUucGhw">
   <param id="Any_5" name="FEED_4" value="08-24-2013 11:41:43">
   <param id="Any_6" name="FEED_5" value="1595">
   <param id="Any_7" name="FEED_6" value="0">
</object>
```

```
<object width="100%" height="150" codebase="http://java.sun.com/update/1.6.0/jinstall-6-windows-i586.cab#Version=6,0,0,0"
classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93">
   <param value="gov.nic.keyupload.gui.KeyRegister.class" name="CODE">
   <param value="KeyRegister.jar" name="ARCHIVE">
   <param value="application/x-java-applet;version=1.6" name="type">
   <param value="true" name="MAYSCRIPT">
   <param id="Any_8" name="FEED_1" value=""/>
   <param id="Any_9" name="FEED_2" value="1"/>
   <param id="Any_10" name="FEED_3" value="ahR0cDovLzEwLjE4NC4xMjguNy9EU0MvZW5yb2xsQ2VyGlmaWNhdGUucGhw">
   <param id="Any_11" name="FEED_4" value="08-24-2013 11:41:43">
   <param id="Any_12" name="FEED_5" value="1595">
   <param id="Any_13" name="FEED_6" value="0">
</object>
```

<![if !IE]><!-->
<!--![if !IE]-->
Procedure to integrate DSC into Applications:

The certificate registration is the first part of the integration. After the successful logon into the system, we show the option for enrolment for their security tokens. This can be achieved by using KeyRegister.jar file. This applet will list all the certificates present in your protected keystore, after which we choose which certificate we are going to enrol for our account. The applet will check for the validity, trustworthiness of the certificate and finally map the certificates into the user account.

While designing the tables for DS in our project, don't map directly into the user table. Create the new table that consists of all DSC in our application and user table mapped into the DSC table through the same reference. It's to cater the need of verification of previously signed data; we need to keep the old/revoked certificates for future reference in our application. Keep the certificates and signing/encryption parameters in Base64 encoded format.

Now, we have the certificates in X.509 format at server for verification. This process is accomplished by the VerifyDSC.jar file. After successfully logon with their password, we prompt the Challenge & Response by sending random values to the client, the user will signing it with their private key and sends that back to the server for verification. We will verify the signature by their public key extract from the certificate. This process depends entirely on the server-side language used in our application.

In PHP, this can be achieved by openssl extension. For verification use this function,

```
openssl_verify(<Plain text of signing data>, <cyber text of signed data>, <certificate used for signing in PEM>)
```

PEM means (Privacy-enhanced Electronic Mail) Base64 encoded DER certificate, enclosed between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----"

Suppose we need to sign some user input data or server generated data for security. We can use the VerifyDSC.jar file with the user input/server-generated data instead of random values used at the time of secure logon. This process also known as Form Signing. The resultant signed data along with the original data sent back to the server for verification. If the signed message is then it will be stored along with the message in plain text, the certificate used for signing and etc.

Integration of DSC into projects is simple task with minimum code change at the application. But in the long run, handling of certificates and signed messages without proper understanding is really challenging job. So, before jumping on the DSC bandwagon make sure we have the strong foothold on the fundamentals of digital signatures and their implications on real-time application.

Other than the secure login and form signing, DSC also extensively used on file signing. Due to maintain the authenticity of server generated reports/files and user uploaded some files, we need to sign those files with the specified certificates to prove their CIA. We can use the SignFiles.jar file to sign any PDF files using security token at the client end. These files will be downloaded directly from the server and message digest created & signed at the client end through the SignFiles applet. This jar file had extensive dependency due to PDF generation. The dependent jar files as follows.

- itextpdf-5.4.3.jar : PDF Generation and Signing
- bcprov-jdk15on-149.jar : Bounty Castle as Security provider
- bcpkix-jdk15on-149.jar : BC library with PKI specification like OCSP (Online certificate Status Protocol)

All the applets mentioned here consists of parameters like Certificate Alias, Server Time (Used as Time Stamping), User ID, Certificate in PEM format, Response URL.

We can also write web service to cater signing and verification of DSC at central level for all application instead of keeping various DSC deployments at each project level. In this way, it eliminates writing the DSC related codes like certificate verification, signature generation etc., & maintains of DSC codes in each project deployments.

While enrolling, most of our application (e-File) checking only certificate validity without checking the user information. But we need to verify, whether the enrolling user name is same as specified in the certificate. Otherwise, we need to reject the DSC registration process or at least give warning.
Security Flaw in DSC at Driver:

While using DSC based web applications, hardware security module (HSM)/security tokens doesn't prompt for PIN to subsequent access. For example, after we successfully logon into e-office using security token by giving PIN, next subsequent access like document approval or note sheet signed automatically done without giving PIN number. This leaves some security flaw like without PIN also hacker can gain access to secured system once he/she got the token and password credentials within few elapsed times after successful access to the token in same browser. This problem is due to the session handling done at the token driver level for minimizing the number of times PIN dialog prompted. This problem can be eliminated by instantiating new keystore through Keystore builder with SunPKCS11 security provider & native driver details instead of Windows-MY keystore. Once the successful logout done, we need to make sure to set keystore object to null and call logout function on the sunpkcs11 provider. In this way, we can successfully clear any cache of PIN at the security token’s driver. If we rely only on Windows-MY keystore, this security hole can’t be patched without the feature request for updated driver from vendor.
What is Fuzzy Logic?

Before illustrating the mechanisms which make fuzzy logic machines to work, it is important to realize what fuzzy logic actually is. Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth—truth values between "completely true" and "completely false". As its name suggests, it is the logic underlying modes of reasoning which are approximate rather than exact. The importance of fuzzy logic derives from the fact that most modes of human reasoning and especially common sense reasoning are approximate in nature. Lotfi A. Zadeh (1965) introduced the term fuzzy logic and described the mathematics of fuzzy set theory.

An assertion can be more or less true in fuzzy logic. In classical logic an assertion is either true or false—not something in between—and fuzzy logic extends classical logic by allowing intermediate truth values between zero and one. It enables a computer to interpret a linguistic statement such as if the washing machine is half full, then use less water. It also adds intelligence to the washing machine since the computer infers an action from a set of such if-then rules. Fuzzy logic means 'computing with words' quoted by the creator "Lotfi A. Zadeh"

The essential characteristics of fuzzy logic are as follows.
- In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
- In fuzzy logic everything is a matter of degree.
- Any logical system can be fuzzified.
- In fuzzy logic, knowledge is interpreted as a collection of elastic or, equivalently, fuzzy constraint on a collection of variables.
- Inference is viewed as a process of propagation of elastic constraints.

Fuzzy Set:

Fuzzy set theory is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. The aim of Zadeh was to give us a language, with syntax and local semantics, in which we can translate our qualitative knowledge about the problem to be solved. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions.

Fuzzy-set theory has been used over the past 10 years in numerous scientific and engineering applications, primarily in control systems and pattern recognition. Application of fuzzy-set theory in the social sciences, however, has been a much slower process. Fuzzy-set theory represents a generalization of classical (crisp) set theory. A fuzzy set is defined by a function that ranges between 0 and 1, which assigns the degrees of membership to each element in a set. Intuitively, the degree of membership represents the extent to which an expert's judgment places an element in a set. An element can belong to more than one set with different degrees of membership. This allows a gradual transition among adjacent sets. Thus, it allows us to view concepts of possibility and vagueness separate from probabilistic or random uncertainty. That's a lot of mathematical baggage, so here's an example. Let's talk about people and "tallness". In this case the set S (the universe of discourse) is the set of people. Let's fine a fuzzy subset TALL, which will answer the question "to what degree is person x tall?" To each person in the universe of discourse, we have to assign a degree of membership in the fuzzy subset TALL. The easiest way to do this is with a membership function based on the person's height.
tall(x) = 0, if height(x) < 5 ft.,
       (height (x)-5ft.)/2ft., if 5 ft. <= height (x) <= 7 ft.,
       1, if height(x) > 7 ft. }

A graph of this looks like:

Warning: put on your peril-sensitive sunglasses. Bad ASCII graphics follow!

1.0 +                   +-------------------
|                  /                |
|                 /                |
|               /                |
|              /                |
|             /                |
|            /                |
|           /                |
|          /                |
|         /                |
|        /                |
|       /                |
|      /                |
|     /                |
|    /                |
|   /                |
|  /                |
| /                |
|+-------------------+-----+-------------------
0.0                    5.0  7.0

Given this definition, here are some example values:
Person   Height   degree of tallness
-----------------------------
Billy    3' 2"    0.00 [I think]
Yoke     5' 5"    0.21
Drew     5' 9"    0.38
Erik     5' 10"   0.42
Mark     6' 1"    0.54
Kareem   7' 2"    1.00 [depends on who you ask]

So given this definition, we'd say that the degree of truth of the statement "Drew is TALL" is 0.38.
What does it offer?

The first applications of fuzzy theory were primly industrial, such as process control for cement kilns. However, as the technology was further embraced, fuzzy logic was used in more useful applications. In 1987, the first fuzzy logic-controlled subway was opened in Sendai in northern Japan. Here, fuzzy-logic controllers make subway journeys more comfortable with smooth braking and acceleration. Best of all, all the driver has to do is push the start button! Fuzzy logic was also put to work in elevators to reduce waiting time. Since then, the applications of Fuzzy Logic technology have virtually exploded, affecting things we use every day. Take for example, the fuzzy washing machine. A load of clothes in it and press start, and the machine begins to churn, automatically choosing the best cycle. The fuzzy microwave, Place chili, potatoes, or etc in a fuzzy microwave and push single button, and it cooks for the right time at the proper temperature. The fuzzy car, maneuvers itself by following simple verbal instructions from its driver. It can even stop itself when there is an obstacle immediately ahead using sensors. But, practically the most exciting thing about it, is the simplicity involved in operating it.

Fuzzy Components:

A fuzzy inference system (FIS) maps crisp inputs to crisp outputs. An FIS consists of four components: the fuzzifier, inference engine, rule base, and defuzzifier. The fuzzifier maps input numbers into corresponding fuzzy membership values. The inference engine defines mapping from input fuzzy sets to output fuzzy sets. It determines the degree to which the antecedent part is satisfied for each rule. If the antecedent part of the rule has more than one clause, fuzzy operators are applied to obtain a number that represents the result of the antecedent part for that rule. Outputs of all rules are then aggregated. The defuzzifier maps the output fuzzy sets into a crisp number. In designing a FIS we need fuzzy rules. Often, these rules are obtained from expert’s knowledge. However, it is also possible to generate fuzzy rules from sample data points.

Fuzzy Logic Application:

Fuzzy Firewall

In addition, “crisp” (the term “crisp” is used as the opposite of “fuzzy” ) rules are defined and used to “tell” the firewall what is allowed to pass or not through the network interface. This way, we can block all incoming echo-request ICMP packets of a given set of hosts and admit GET SNMP packets of another one. Note that, in this scheme, the rules force the firewall to behave in a static (all or nothing) manner, not leaving room for adaptive dynamics, based on parameters collected from the network traffic. A very known kind of network threat is the “denial of service” attack, that consists of bombarding network servers with enormous amounts of well-formed requests, driving them to their knees. That could occur using several protocols like ICMP (echo-request), UDP (SNMP GET request), TCP (SMTP transactions), etc.. The key idea is that there is a certain threshold after which the computational resources will be exhausted and weird things could happen.

But what to do to ensure maximum usability by keeping the site at good levels of security? The answer seems to be in the use of Soft Computing technique to build a model of an “intelligent” and adaptive firewall that changes its behavior depending on the dynamical conditions of networks and computational resources. For the sake of simplicity, we’ll focus the discussion on the protection against denial of service attacks,
to port 80, over tcp, to a local machine we would do:

```bash
iptables -A INPUT -p tcp --dport 80 -j DROP
```

-p sets the IP protocol used, be it TCP, ICMP, UDP or one of the other more unusual protocols, and --dport specifies the destination port of the packet. We can, of course, use --sport to specify a source port, but that is rarely used as connections normally use a high source port, unless they are from a specific service, which has packets coming from a specific port. As with every firewall we have notice, the rules are Aristotelian (they don’t allow a rule to be partially satisfied), that is, one of two conditions is met: a given packet flow is totally blocked or totally freed to pass through the filters. But, in many cases, DoS (Denial of Service) attacks consist of using perfectly legal services with enormous packet rates, exhausting server and communication resources and, frequently, causing interrupts in normal service. **The question is: How to maintain services publicly available and to block DoS attacks at the same time?**

The answer seems to be related to the degree at which the traffic hits the server infrastructure – if packet rate is low or moderate then everything is ok. If traffic is excessive, we may be facing DoS attack. So, it is necessary to design mechanisms that can “understand” and take evasive actions in adverse conditions and remove barriers when conditions are favorable. To this end, the Netfilter/iptables system was endowed with a extension (the fuzzy match) that allowed the user to translate his own degree perception. The fuzzy match allows you to match packets according to a dynamic profile implemented by means of a simple Fuzzy Logic Controller (FLC).

- When the packet rate is below `lower-limit`, the rule will never match.
- Between `lower-limit` and `upper-limit`, matching will occur according an increasing (mean) rate.
- Finally, when the packet rate comes to `upper-limit`, (mean) matching rate attains its maximum value, 100%.

For example, if you wish to avoid generic Denial Of Service attacks, you could use the following rule:

```bash
iptables -A INPUT -m fuzzy --lower-limit 100 --upper-limit 1000 -j REJECT
```

- Below the 100 pps (packets per second) rate, the filter is inactive.
- Between 100 and 1000 pps the mean acceptance rate drops from 100% (when we are at 100 pps) to 0%.
- Above 1000 pps the acceptance rate keeps constant at 0%.

---

**Evaluation of Government Performance**

Generally Technology is the invention of tools and techniques that enable man to create new ways of living and new ways of governance. An important area in which fuzzy-set theory can be applied is performance evaluation. Fuzzy logic models provide a reasonable solution to these common situations, which may easily be converted into human linguistic forms and subjective constructs. Fuzzy logic is a problem solving methodology that provides a simple way of drawing definite conclusions from vague and imprecise information. For instance, fuzzy linguistic variable performance can be categorized as low, average, outstanding. Each category is called a linguistic modifier. Each of these linguistic modifiers is linked to a numeric value on a particular scale. Example on this is shown in figure 3. In this figure, three fuzzy sets are used to characterize government performance (low, average and outstanding) on a scale of 0 to 6 with each linguistic modifier having membership value from 0 to 1. The fuzzy linguistic variable "performance" in this example represents government performance. Scale with number 2 represents the highest level of low performance with a membership value of 0.3, while scale number 3 defines low performance with a membership value of 0.1 or average performance with membership value of 0.3. This implies that scale number 3 describes government performance that is 30% average and 10% low. The three fuzzy sets are as follow:
FUZZY Logic and its possible scopes in e-Governance

Low performance = {0.1|0.0, 0.5|0.9, 1.0|0.5, 1.5|0.4, 2.0|0.3, 2.5|0.2, 3.0|0.1, 3.5|0.0, 4.0|0.0, 4.5|0.0, 5.0|0.0, 6.0|0.0}

Average performance = {0.0|0.0, 0.5|0.0, 1.0|0.0, 1.5|0.0, 2.0|0.0, 2.5|0.0, 3.0|0.0, 3.5|0.0, 4.0|0.0, 4.5|0.0, 5.0|0.0, 5.5|0.0, 6.0|0.0}

Outstanding performance = {0.0|0.0, 0.5|0.0, 1.0|0.0, 1.5|0.0, 2.0|0.0, 2.5|0.0, 3.0|0.0, 3.5|0.0, 4.0|0.0, 4.5|0.0, 5.0|0.0, 5.5|0.0, 6.0|0.0}

Electronic Evaluation (e-evaluation) involves assessing the strengths and weaknesses of programs, policies, and public agencies to improve their effectiveness through the use of computer on network of networks (Internet). Governments do make several promises prior to elections, it is therefore necessary to evaluate the level of fulfillment of these promises and make recommendations that could lead to its sustenance. An important area in which fuzzy-set theory can be applied is performance evaluation. Fuzzy logic models provide a reasonable solution to these common situations, which may easily be converted into human linguistic forms and subjective constructs. Fuzzy logic is a problem solving methodology that provides a simple way of drawing definite conclusions from vague and imprecise information. Fuzzy set theory is applied to complement the framework in order to capture fuzziness in the form of inconsistencies and vagueness coming from subjective judgments by the evaluator or decision makers.

jFuzzyLogic: A Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming

jFuzzyLogic is a fuzzy logic package written in java (as you might have guessed). It implements Fuzzy control language (FCL) specification (IEC 61131 part 7) Fuzzy Control is emerging as a technology that can enhance the capabilities of industrial automation, and is suitable for control level tasks generally performed in Programmable Controllers (PC) Fuzzy Logic Controllers are a specific model of Fuzzy Rule Based Systems suitable for engineering applications for which classic control strategies do not achieve good results or for when it is too difficult to obtain a mathematical model. Recently, the International Electro technical Commission has published a standard for fuzzy control programming of the IEC 61131 norm in order to offer a well defined common understanding of the basic means with which to integrate fuzzy control applications in control systems. jFuzzyLogic is open source Java library called jFuzzyLogic which offers a fully functional and complete implementation of a fuzzy inference system according to this standard, providing a programming interface and Eclipse plugin to easily write and test code for fuzzy control applications.

Aforgenet: It is a class library and extention to Microsoft.Net to write various Artificial Intelligence Programming

Writing a computer program implementing the above fuzzy logic using Microsoft.Net and Aforge.Net library is also don’t required any thing to build from scratch. Aforge.Net is a most popular library extension for .Net framework which can be used to develop various Artificial Intelligence programming. This library contents man ready made algorithms so that new research can be done without reinvention the same wheel again and new extension of the existing algorithm can be designed with in very short period of time. Aforge.Fuzzy.dll namespace gives the freedom to the developer from understanding and implementing the Fuzzy logic core engine from scratch instead developer can only focus on the rollout plan.

So, gets warm up with these two popular Fuzzy libraries for both JAVA as well as for .Net and explore the world of Artificial Intelligence and machine learning.
Writing an Intelligent program using most popular AI algorithms doesn’t need to be started from scratch every time. So the role of class library of many popular AI algorithms becomes very important. Although there are many library for .net but still we found the Aforge.net is more robust and error free. Acord.net is also another library available but that is also extended from this Aforge.net only. So to begin with AI programming we suggest first to understand the theory behind the popular AI concepts as well as algorithms then start developing using the class library.

AForge.NET is an open source C# framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence - image processing, neural networks, genetic algorithms, fuzzy logic, machine learning, robotics, etc.

The framework is comprised by the set of libraries and sample applications, which demonstrate their features:

- AForge.Imaging- library with image processing routines and filters;
- AForge.Vision - computer vision library;
- AForge.Video - set of libraries for video processing;
- AForge.Neuro - neural networks computation library;
- AForge.Genetic - evolution programming library;
- AForge.Fuzzy - fuzzy computations library;
- AForge.Robotics - library providing support of some robotics kits;
- AForge.MachineLearning - machine learning library;

The work on the framework's improvement is in constants progress, what means that new feature and namespaces are coming constantly. To get knowledge about its progress you may track source repositories log or visit project discussion group to get the latest information about it. The framework is provided not only with different libraries and their sources, but with many sample applications, which demonstrate the use of this framework, and with documentation help files, which are provided in HTML Help format. The documentation is also available on-line.

Aforge.Net has few very popular extension libraries available and out of them GRATF and IP is very powerful as well as interesting to see.
GRATF stands for Glyph Recognition And Tracking Framework. The project is aimed to provide a library which does localization, recognition and pose estimation of optical glyphs in still images and video files. The library can be used in robotics applications for example, where glyphs may serve as commands or directions to robots. However, most popular application of optical glyph recognition is augmented reality. For detailed description of algorithms implemented in this project, you are welcome to read next series of articles:

- Glyph's Recognition;
- 3D Pose Estimation;
- 3D Augmented Reality;

The project includes:

- Glyph recognition and pose estimation library, which is an extension to AForge.NET framework. The library can be reused in other applications which require glyph recognition functionality.
- Glyph recognition prototype, which is implemented as a plug-in to IPPrototyper tool from AForge.NET framework. Can be used for demonstrating glyph recognition algorithm's idea step by step or for its further improvement, debugging and testing. Simply open a folder containing images with glyphs (sample images are provided) and see steps of glyph recognition algorithm.
- Glyph Recognition Studio - an application which does recognition of glyphs in video streams (video files or local capture devices, like web cameras). Also the application does 2D and 3D augmented reality - puts images into video feed replacing recognized glyphs or puts 3D models on top of recognized glyphs using same pose as real glyphs have - virtual objects in real world. Just define some glyphs in the Glyph Recognition Studio, print them from the application and see how it works with your web camera.
Image Processing Lab is an image processing application written in C#, which includes different filters and tools available in AForge.NET framework to analyze images.

The following filters are available in the IPLab application:

- **Color filters** (grayscale, sepia, invert, rotate, channel extraction, channel replacing, channel filtering, color filtering, Euclidean color filtering);
- **HSL filters** (linear correction, brightness, contrast, saturation, hue modifier, HSL filtering);
- **YCbCr filters** (linear correction, YCbCr filtering, channel extraction/replacement);
- **Binarization filters** (threshold, threshold with carry, ordered dithering, Bayer dithering, Floyd-Steinberg, Burkes, Jarvis-Judice-Ninke, Sierra, Stevenson-Arce, Stucki dithering methods);
- **Automatic binarization** (simple image statistics, Otsu);
- **Mathematical morphology filters** (erosion, dilatation, opening, closing, hit-and-miss, thinning, thickening);
- **Convolution filters** (mean, blur, sharpen, edges, Gaussian);
- **2 Source filters** (merge, intersect, add, subtract, difference, move towards, morph, stereo anaglyphs);
- **Edge detectors** (homogeneity, difference, sobel, canny);
- **Corners' detectors** (SUSAN, Moravec);
- **Blobs' filtering**, Largest blob extractor, Blob counter, Connected components labeling;
- **Pixellate, Simple skeletonization, Jitter, Shrink, Oil painting, Simple posterization**;
- **Levels linear filter, gamma correction, contrast stretch, histogram equalization**;
- **Median filter, Adaptive smoothing, Conservative smoothing**;
- **Resize and Rotate**;
- **Texture generators based on Perlin noise**;
- **Texture filters** (texturer, textured filtering, textured merging);
- **Fourier transformation** (low-pass and hi-pass filters);
- **Document skew correction**;

It is possible to create (save and load) your own convolution filters or filters based on standard mathematical morphology operators. Colorized grid makes it very convenient to work with custom convolution filters. A preview window allows viewing results of changing filters' parameters on the fly. It is possible to scroll an image using mouse in preview area. All filters are applied only to the portion of image currently viewed to speed up preview. A Photo Shop like histogram that allows getting information about mean, standard deviation, median, minimum and maximum values. The application allows copying to or paste from clipboard, save and print images.
Fuzzy Logic is one of the most popular words perhaps every computer engineer certainly aware about and is the foundation stone of Natural language processing as well as soft computing. This is the heart of Artificial Intelligence or Machine Intelligence but still it is least popular to be used in e-governance area. Recently planning commission of India was in media controversy after announcing the income per day should be less than Rs. 20/- in rural India to be recognized as below poverty line. This decision was considered by many learned panel and in civil society as completely impractical and joke towards the poor people of India. But the main reason for this problem in fixation of poverty line is due the lack of fuzzification in government policy to properly define the linguistic variables using only two terms BPL or APL.

When we create a poverty line on the basis on income per day per person and we set only two variables to represent the two sets across the line as APL & BPL, then an iterative reduced shadow effect comes to role. Let’s understand the “Iterative reduced shadow effect” in a simple example.

Suppose we say, If a person making income of Rs.500 per day, then is he APL or BPL?

Ans: Of course APL

What about a person making an income of Rs.300 per day?

Ans: Common Sense he is APL !!

Then what about a person making an income of Rs.200 per day?

Ans: Simple APL !!

Sure then what can you say about an income of Rs.100 per day?

Ans: Ahhhh…. APL but not that much (If we don’t consider planning commission recommendation)

No, no.. we have to answer in only APL or BPL no other way to represent.

Ans: then for Rs.100 per day! Take it as APL

Ok what about Rs. 75 per day?

Ans: He should be BPL

Then only for Rs.25 more income in case of Rs.100 per day income he will not eligible for BPL card as well as low cost Rice??

Ans: No, no, you do one thing for income Rs.50 to Rs.75 you keep it as APL but below Rs.50 as BPL
Good but what about the income of Rs.49 and Rs.50 per day people. Only for one rupee extra income the latter will not get the benefit of low cost rice where as with former will enjoy the benefit.

**Ans:** In this way if we will go down then even Rs.10/- per day income will not also come as poor and that how planning commission come down to Rs.20. Then there is no solution other than FUZZY logic.

Every answer can’t be given using two Boolean options as true or false so the poverty status of any person can’t be stated as APL or BPL always. Fuzzy logic has the concept of membership value and can represent any object as the part of multiple sets. Valid range of value forms fuzzy set and multiple fuzzy sets define the state of a “Linguistic Variable” where as database consists of one or many Linguistic Variables. Rule base contains one or more than one rules and rules define the relationship between the different states of the Linguistic Variables.
Instead of creating two sets of citizen considering the poverty line as the bench mark which creates discrepancy in requirement and the supply, fuzzy model can be incorporated which provides adequate amount of subsidy to the needy with much more effective way than the conventional system. In the Fuzzy way to distribute subsidy saves the resources where the need is minimum and supports the maximum needy section of the society by maintaining a perfect justifiable financial balance.

Now we will learn how write a computer program implementing the above logic in Microsoft.Net and Aforge.Net library. Aforge.Net is a most popular library extension for .Net framework which can be used to develop various Artificial Intelligence programming. This library contents many readymade algorithms so that new research can be done without reinvention the same wheel again and new extension of the existing algorithm can be designed with in very short period of time.

**Step1:** Download and install Aforge.net library from the [http://aforgenet.com](http://aforgenet.com) website.

**Step2:** Create a new project in Visual Studio and right click on bin folder then select Add Reference.

**Step3:** Select the Aforge.dll and Aforge.Fuzzy.dll for specifically for this project and click “OK”

**Step4:** Create a web page with one Text Box and one Button control. Let the Text Box will take the income per day as the input and after clicking the Button control the result will show the amount of subsidy to be given in percentage.

**Step5:** Define a function `InitFuzzyEngine()` and Import the name space “AForge.Fuzzy”. Create a global variable `ISys` as InferenceSystem.
The main function of this subroutine is to create two linguistic variables INCOME & SUBSIDY and assign each of them Fuzzy Sets (called Label) {"VERY POOR”, “POOR”, “RICH”) as well as {“MORE”, “STANDARD”, “VERYLITTLE”) respectively. Creates a database (not a physical database but a database object defined in Aforge.dll name space) to store the two linguistic variables. As previously discussed Inference system is takes the database of linguistic variables and rule base to operates on input and to provide crisp output, so rules have been created by relating the linguistic variables and added to the inference system along with the database.

Private Sub InitFuzzyEngine()
    Dim lv_income As LinguisticVariable = New LinguisticVariable("INCOME", 0, 2000)
    Dim fs_verypoor As FuzzySet = New FuzzySet("VERYPOOR", New TrapezoidalFunction(0, 100, 200, TrapezoidalFunction.EdgeType.Right))
    Dim fs_poor As FuzzySet = New FuzzySet("POOR", New TrapezoidalFunction(100, 200, 350, 400))
    Dim fs_rich As FuzzySet = New FuzzySet("RICH", New TrapezoidalFunction(350, 400, 451, TrapezoidalFunction.EdgeType.Left))
    lv_income.AddLabel(fs_rich)
    lv_income.AddLabel(fs_poor)
    lv_income.AddLabel(fs_verypoor)
    Dim lv_subsidy As LinguisticVariable = New LinguisticVariable("SUBSIDY", 0, 100)
    Dim fs_nil As FuzzySet = New FuzzySet("VERYLITTLE", New TrapezoidalFunction(0, 0, 0, TrapezoidalFunction.EdgeType.Right))
    Dim fs_standard As FuzzySet = New FuzzySet("STANDARD", New TrapezoidalFunction(0, 10, 20, 30))
    Dim fs_more As FuzzySet = New FuzzySet("MORE", New TrapezoidalFunction(20, 30, 40, TrapezoidalFunction.EdgeType.Left))
    lv_subsidy.AddLabel(fs_nil)
    lv_subsidy.AddLabel(fs_standard)
    lv_subsidy.AddLabel(fs_more)
    Dim fuzzy_dbase As New Database
    fuzzy_dbase.AddVariable(lv_income)
    fuzzy_dbase.AddVariable(lv_subsidy)
    ISys = New InferenceSystem(fuzzy_dbase, New CentroidDefuzzifier(1000))
    ISys.NewRule("RULE1", "IF INCOME IS POOR THEN SUBSIDY IS STANDARD")
    ISys.NewRule("RULE2", "IF INCOME IS VERYPOOR THEN SUBSIDY IS MORE")
    ISys.NewRule("RULE3", "IF INCOME IS RICH THEN SUBSIDY IS VERYLITTLE")
End Sub

The main function of this subroutine is to create two linguistic variables INCOME & SUBSIDY and assign each of them Fuzzy Sets (called Label) {“VERY POOR”, “POOR”, “RICH”) as well as {“MORE”, “STANDARD”, “VERYLITTLE”) respectively. Creates a database (not a physical database but a database object defined in Aforge.dll name space) to store the two linguistic variables. As previously discussed Inference system is takes the database of linguistic variables and rule base to operates on input and to provide crisp output, so rules have been created by relating the linguistic variables and added to the inference system along with the database.
Step 6: Call the initFuzzyEngine() function from the Page Load event and set the input value of ISys.

Step 7: Run the application and test by giving input in the textbox as income per day in Rupees.
This program is a prototype which takes the “income per day” value as input and produces the “subsidy amount” as crisp output in percentage. Range value set for each Fuzzy Set can be adjusted according to the requirement to justify the distribution and to achieve the most achievable accuracy. Similar type of implementation of fuzzy logic can also be used in case of income tax (A person with income Rs.9999 does not have to pay IT but income with Rs.10001 has to pay IT over entire Rs.10001.), IP Address threat rating (Instead of two rating options like Black Listed or Not fuzzy set can be used) and in many more scenarios which is a very researchable topic in e-governance.

COMPLETE CODE OF THE IMPLEMENTATION in ASP.Net (VB Language)

Imports AForge.Fuzzy
Partial Class _Default
    Inherits System.Web.UI.Page
    Dim ISys As InferenceSystem
    Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
        ISys.SetInput("INCOME", Convert.ToDouble(Me.TextBox1.Text))
        Response.Write(ISys.Evaluate("SUBSIDY"))
    End Sub

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        InitFuzzyEngine()
    End Sub

    Private Sub InitFuzzyEngine()
        Dim lv_income As LinguisticVariable = New LinguisticVariable("INCOME", 0, 2000)
        Dim fs_verypoor As FuzzySet = New FuzzySet("VERYPOOR", New TrapezoidalFunction(0, 100, 200, TrapezoidalFunction.EdgeType.Right))
        Dim fs_poor As FuzzySet = New FuzzySet("POOR", New TrapezoidalFunction(100, 200, 350, 400))
        Dim fs_rich As FuzzySet = New FuzzySet("RICH", New TrapezoidalFunction(350, 400, 451, TrapezoidalFunction.EdgeType.Left))
        lv_income.AddLabel(fs_rich)
        lv_income.AddLabel(fs_poor)
        lv_income.AddLabel(fs_verypoor)
        Dim lv_subsidy As LinguisticVariable = New LinguisticVariable("SUBSIDY", 0, 100)
        Dim fs_nil As FuzzySet = New FuzzySet("VERYLITTLE", New TrapezoidalFunction(0, 0, 0, TrapezoidalFunction.EdgeType.Right))
        Dim fs_standard As FuzzySet = New FuzzySet("STANDARD", New TrapezoidalFunction(0, 10, 20, 30))
        Dim fs_more As FuzzySet = New FuzzySet("MORE", New TrapezoidalFunction(20, 30, 40, TrapezoidalFunction.EdgeType.Left))
        lv_subsidy.AddLabel(fs_nil)
        lv_subsidy.AddLabel(fs_standard)
        lv_subsidy.AddLabel(fs_more)

        Dim fuzzy_dbase As New Database
        fuzzy_dbase.AddVariable(lv_income)
        fuzzy_dbase.AddVariable(lv_subsidy)
        ISys = New InferenceSystem(fuzzy_dbase, New CentroidDefuzzifier(1000))
        ISys.NewRule("RULE1", "IF INCOME IS POOR THEN SUBSIDY IS STANDARD")
        ISys.NewRule("RULE2", "IF INCOME IS VERYPOOR THEN SUBSIDY IS MORE")
        ISys.NewRule("RULE3", "IF INCOME IS RICH THEN SUBSIDY IS VERYLITTLE")
    End Sub
End Class
Indian develops tech to make software Unhackable

A unique system has been designed by researchers that will encrypt software in order to make it impervious to reverse-engineering. UCLA computer science professor Amit Sahai and a team of researchers have developed a system which will only allow someone to use a programme as intended, while preventing any deciphering of the code behind it. This is known as software obfuscation in computer science and it is the first time it has been accomplished. Sahai said that the new system puts up an iron wall making it impossible for an adversary to reverse-engineer the software without solving mathematical problems that take hundreds of years to work out on today's computers.

The researchers said their mathematical obfuscation mechanism can be used to protect intellectual property by preventing the theft of new algorithms and by hiding the vulnerability a software patch is designed to repair when the patch is distributed. The key to this successful obfuscation mechanism is a new type of multi linear jigsaw puzzle. This new technique has paved the way for another

US snooping programme had one server located in India, report says

A controversial US surveillance programme that sweeps internet usage data had 700 snooping servers installed at 150 locations around the world, including one in India, according to a report. The XKeyscore programme, run by the National Security Agency (NSA), allowed analysts to search through vast databases containing emails, online chats and browsing histories of millions of individuals, Guardian reported, citing documents provided by whistleblower Edward Snowden. A February 2008 training material presentation for the XKeyscore programme included a map of locations of the surveillance servers, one of which appeared to be near New Delhi, according to the paper's report.

The NSA said XKeyscore is its "widest reaching" system to develop intelligence from computer networks, Guardian said. The presentation claimed the program covers "nearly everything a typical user does on the Internet," including the content of e-mails, websites visited and searches.

US companies IBM, HP and Dell dominate the Indian server market, according to market research and analyst firm IDC. No Indian company makes computer or electronic chips that are key components of servers.

The United States house permanent select committee on intelligence said in a statement the report was misleading and that "the program is simply a tool used by our intelligence analysts to better understand foreign intelligence, including terrorist targets overseas." The statement added that the program does not target American citizens and is not used for indiscriminate monitoring of the internet.
Computer System Automatically Generates TCP Congestion-Control Algorithms

One of TCP's main functions is to prevent network congestion by regulating the rate at which computers send data. In the last 25 years, engineers have made steady improvements to TCP's congestion-control algorithms, resulting in several competing versions of the protocol: Many Windows computers, for instance, run a version called Compound TCP, while Linux machines run a version called TCP Cubic. At the annual conference of the Association for Computing Machinery's Special Interest Group on Data Communication this summer, researchers from MIT's Computer Science and Artificial Intelligence Laboratory and Center for Wireless Networks and Mobile Computing will present a computer system, dubbed Remy, that automatically generates TCP congestion-control algorithms. In the researchers’ simulations, algorithms produced by Remy significantly outperformed algorithms devised by human engineers.

"I think people can think about what happens to one or two connections in a network and design around that," says Hari Balakrishnan, the Fujitsu Professor in Electrical Engineering and Computer Science, who co-authored the new paper with graduate student Keith Winstein. "When you have even a handful of connections, or more, and a slightly more complicated network, where the workload is not a constant -- a single file being sent, or 10 files being sent -- that's very hard for human beings to reason about. And computers seem to be a lot better about navigating that search space."

Lay of the land

Remy is a machine-learning system, meaning that it arrives at its output by trying lots of different possibilities, and exploring further variations on those that seem to work best. Users specify certain characteristics of the network, such as whether the bandwidth across links fluctuates or the number of users changes, and by how much. They also provide a "traffic profile" that might describe, say, the percentage of users who are browsing static web pages or using high-bandwidth applications like videoconferencing. Finally, the user also specifies the metrics to be used to evaluate network performance. Standard metrics include throughput, which indicates the total amount of data that can be moved through the network in a fixed amount of time, and delay, which indicates the average amount of time it takes one packet of information to travel from sender to receiver. The user can also assign metrics different weights -- say, reducing delay is important, but only one-third as important as increasing throughput. Remy needs to test each candidate algorithm's performance under a (continued to page 2…

'Living' Micro-Robot Could Detect Diseases in Humans

Called 'Cyberplasm', it will combine advanced microelectronics with latest research in biomimicry (technology inspired by nature). The aim is for Cyberplasm to have an electronic nervous system, 'eye' and 'nose' sensors derived from mammalian cells, as well as artificial muscles that use glucose as an energy source to propel it. The intention is to engineer and integrate robot components that respond to light and chemicals in the same way as biological systems. This is a completely innovative way of pushing robotics forward. Cyberplasm is being developed over the next few years as part of an international collaboration funded by the Engineering and Physical Sciences Research Council (EPSRC) in the UK and the National Science Foundation (NSF) in the USA. The UK-based work is taking place at Newcastle University. The project originated from a 'sandpit' (idea gathering session) on synthetic biology jointly funded by the two organisations. Cyberplasm will be designed to mimic key functions of the sea lamprey, a creature found mainly in the Atlantic Ocean. It is believed this approach will enable the micro-robot to be extremely sensitive and responsive to the environment it is put into. Future uses could include the ability to swim unobtrusively through the human body to detect a whole range of diseases. The sea lamprey has a very primitive nervous system, which is easier to mimic than more sophisticated nervous systems. This, together with the fact that it swims, made the (Continued to page 2……
a wide range of network conditions, which could have been a prohibitively time-consuming task. But Winstein and Balakrishnan developed a clever algorithm that can concentrate Remy's analyses on cases in which small variations in network conditions produce large variations in performance, while spending much less time on cases where network behavior is more predictable.

They also designed Remy to evaluate possible indicators of network congestion that human engineers have not considered. Typically, TCP congestion-control algorithms look at two main factors: whether individual data packets arrive at their intended destination and, if they do, how long it takes for acknowledgments to arrive. But as it turns out, the ratio between the rates at which packets are sent and received is a rich signal that can dictate a wide range of different behaviors on the sending computer's end.

**Down to cases**

Indeed, where a typical TCP congestion-control algorithm might consist of a handful of rules -- if the percentage of dropped packets crosses some threshold, cut the transmission rate in half -- the algorithms that Remy produces can have more than 150 distinct rules.

"It doesn't resemble anything in the 30-year history of TCP," Winstein says. "Traditionally, TCP has relatively simple endpoint rules but complex behaviour when you actually use it. With Remy, the opposite is true. We think that's better, because computers are good at dealing with complexity. It's the behaviour you want to be simple." Why the algorithms Remy produces work as well as they do is one of the topics the researchers hope to explore going forward.

In the meantime, however, there's little arguing with the results. Balakrishnan and Winstein tested Remy's algorithms on a simulation system called the ns-2, which is standard in the field. In tests that simulated a high-speed, wired network with consistent transmission rates across physical links, Remy's algorithms roughly doubled network throughput when compared to Compound TCP and TCP Cubic, while reducing delay by two thirds. In another set of tests, which simulated Verizon's cellular data network, the gains were smaller but still significant: a 20 to 30 percent improvement in throughput, and a 25 to 40 percent reduction in delay.

sea lamprey the best candidate for the project team to base Cyberplasm on. Once it is developed the Cyberplasm prototype will be less than 1cm long. Future versions could potentially be less than 1mm long or even built on a nanoscale.

"Nothing matches a living creature's natural ability to see and smell its environment and therefore to collect data on what's going on around it," says bioengineer Dr Daniel Frankel of Newcastle University, who is leading the UK-based work.

Cyberplasm's sensors are being developed to respond to external stimuli by converting them into electronic impulses that are sent to an electronic 'brain' equipped with sophisticated microchips. This brain will then send electronic messages to artificial muscles telling them how to contract and relax, enabling the robot to navigate its way safely using an undulating motion. Similarly, data on the chemical make-up of the robot's surroundings can be collected and stored via these systems for later recovery by the robot's operators.

Cyberplasm could also represent the first step on the road to important advances in, for example, advanced prosthetics where living muscle tissue might be engineered to contract and relax in response to stimulation from light waves or electronic signals.