SecurePass: Guarding sensitive information from un-trusted machines

Justin Martineau, Palanivel Kodeswaran
{jm1,palanik1}@umbc.edu

Abstract

We propose a proxy based solution to secure web access from un-trusted machines.
Traditional proxy based solutions require constant out of band communication with the proxy. However, our system does not require constant communication with the user. Instead, the user carries a mobile device capable of performing cryptographic operations to generate onetime passwords. These passwords are then used to authenticate to the proxy. We have developed a prototype system and tested its performance in terms of security and response time.

1. Introduction

The Internet is emerging in the mainstream media as a medium for performing commercial transactions. Many people make purchases through online e-commerce sites such as amazon.com, buy.com etc. In addition, we also perform sensitive transactions such as paying bills, transferring money across accounts etc online. Most of these websites make use of a username and password mechanism to authenticate the users to the system. However, traditional methods of authentication often assume that the end systems are secure and trustworthy. Given the prevalence of computers, these assumptions are longer valid. People use computers at cyber cafés, libraries etc to conduct their business transactions. However, there is no guarantee that these machines are secure. A simple key-logger program could record the username and password, and later masquerade as the user. Havoc ensues. Traditional security techniques don't thwart this class of attacks. In this paper, we propose a proxy based mechanism to secure sensitive information from un-trusted machines. Typically, proxy based solutions require that there be a continuous communication channel between the proxy and the user for input validation. However, our system does not require such a communication channel. Instead we require users to carry a mobile device capable of performing certain cryptographic operations.

2. Related Work

Wu et al. [1] proposes a generic framework for web authentication from un-trusted machines using mobile phones as hand held authentication mechanisms. They use a trusted proxy to mediate connections to websites. When the user connects to the proxy, the proxy sends back a randomly chosen word to the kiosk as well as the mobile. The user makes use of a Wireless Access Protocol (WAP)-enabled mobile phone to access a page on the proxy that contains the randomly chosen word. If they match, the user sends back a confirmation to the proxy, else the proxy terminates the connection. To prevent attackers masquerading as the user, the proxy sends a nonce to the user's mobile phone, which is used to uniquely identify the user.

Delegate [2] proposes using a proxy based solution to secure web access from un-trusted machines. Delegate prevents an attacker from stealing passwords, destroying user's personal information, and hijacking user sessions. In this model, the user is assumed to be carrying a trusted mobile device such as a cell phone. The cell phone is used in the initial authentication phase to establish connection with the proxy. All accesses to websites from the un-trusted machine are mediated through the proxy. In other words, the un-trusted machine opens a connection to the proxy, and the proxy opens a connection to the website, thereby effectively

playing a man-in-the middle role. All sensitive information to the website is entered by the proxy by intercepting HTTP POST requests. Similarly, when the website returns sensitive information, the proxy removes the sensitive portion from the response, and forwards the rest to the un-trusted machine. Thus, no sensitive information is ever leaked to the un-trusted machine. The proxy uses the cell phone to communicate with the user in cases when it needs explicit user confirmation. This is done when the proxy thinks that certain actions could be dangerous or when the proxy suspects a potential session hijacking. Delegate reduces the number of times user confirmation is required by implementing rules at the proxy. These rules reflect the user's perception of dangerous situations.

Our system differs from delegate in that we do not need to have constant cellular communication with the proxy.

Clarke et al. [3] propose a camera based authentication mechanism. The main goal of their system is to provide authenticated bidirectional communication between the trusted machine and the user through the un-trusted machine. In this context, authenticated means that messages received are unmodified copies of data sent by the source. They provide a formal model for the problem and a simple proof of how a unidirectional secure channel can be used to obtain a bi-directional secure channel. They compare use of the mobile device as a relaying device and monitoring device. Using the mobile device in monitoring mode is advantageous as the user can leverage the display and computational capabilities of Internet café kiosks which are generally more powerful than handhelds. However, authentication mechanisms in this case need to be more robust. They use the mobile as a monitoring device and their system consists of a camera equipped device that can capture screen shots of the un-trusted machine's screen and perform authentication and integrity checks on the data. In another method, the mobile device takes a screen shot and relays it to the proxy through the un-trusted machine. The proxy then performs an OCR on the received image to validate that the information was properly transmitted.

Ross et al [4] propose a composable framework for secure web access from un-trusted machines using a combination of trusted and un-trusted devices working in tandem. The system architecture isolates device capabilities from service requirements. The system uses a collection of components that act as building blocks to map service requirements to device capabilities using trusted proxy architecture. Some of the components for e.g. perform format encodings to match the display capabilities of the end device. Other components perform semantic transformations so as to ensure that no sensitive information is leaked to the un-trusted machine. All sensitive data are stored at the trusted proxy which injects them at the time of requesting an internet service that requires authentication. By isolating device capabilities from functionalities, we can exploit the rich display of an internet kiosk while using a trusted mobile device to confirm requests to the proxy. Also, the system uses a content filtering mechanism that can be used to obfuscate sensitive data sent to the un-trusted machine as well as verify whether the un-trusted machine displays unmodified data. These operations are performed on the basis of rules located at the proxy.

Vipul Gupta et al [5] describe their experience with implementing standard web-security protocols on mobile handhelds. Traditional wireless clients do not provide end-to-end security protocols such as SSL due to the perceived resource constraints of handhelds in computing cryptographic operations as required by SSL. Instead they use a proxy/gateway to perform end-to-end security. The wireless clients use an incompatible protocol called WTLS to communicate with the WAP gateway. The WAP gateway on the other hand decrypts the WTLS encrypted data from the mobile and re-encrypts using SSL before forwarding to the destination website. Thus, the proxy has complete access to the communication of the mobile clients. Typically the proxy is under the control of an ISP and hence cannot always be trusted. As a result, many of the websites

are reluctant to extend their services to wireless clients due to the lack of end-to-end security. The authors provide empirical evidence to show that standard end-to-end security protocols such as SSL can still be implemented on wireless handhelds without much performance losses in terms of latency and battery life. The judicious use of features in SSL can be leveraged to reduce the amount of computation required to guarantee end-to-end security. For e.g. the shared master-secret can be re-used among multiple sessions when a client communicates with the same server. The authors show that current web security protocols can seamlessly be adapted to the wireless world eliminating the need to develop separate protocols for mobile handhelds.

3. Motivation:

Delegate [2] provides users with a mechanism to authenticate from an un-trusted machine to an Internet service without leaking private user information. However, Delegate assumes that constant wireless or cellular communication from the user to the proxy is possible. This assumption is not always true. Users may often enter regions where a cellular reception and wireless signals lack sufficient signal strength to establish long-term communications with the proxy. Without such a communications channel Delegate [2] fails to function, and thus provides no protection. Instead of requiring users to authenticate online through a mobile device, our system requires users to lookup a onetime password from their mobile device to be used for authentication with the proxy. This allows users to authenticate without using out of band communication with the proxy.

4. System Architecture and Protocol

In this section we provide an overview of our system architecture and how it is used to solve the problem of securing web access from un-trusted machines. In the following we assume that a secret key is shared between the proxy and the mobile device and that the proxy is secure. We also assume that data can be reliably relayed between the un-trusted machine and the mobile device, for e.g. using a USB stick, infrared, or even being entered by hand. To this end, encryption always results in base 64 encoded output of a reasonable length so that it may be entered at the untrusted machine's keyboard if necessary.

Our design is fundamentally similar to Delegate [2]. The user's home/office computer acts as a trusted proxy that authenticates the user to the service he/she is trying to access from the un-trusted machine. The user's authentication details, such as usernames and passwords, for sites and services are stored and protected on the proxy. When a user requests access to a secure website from the un-trusted machine, the proxy will fill in the authentication details on behalf of the user, provided the un-trusted machine provides a valid response to the proxy's one time password challenge. The correct response to this challenge can be computed with the user's mobile device. The user's mobile device is initialized with a onetime password program in conjunction with the proxy. Both the proxy and the mobile device are in sync with respect to the one time password generation. If the response to the challenge matches the value computed by the proxy, then the proxy supplies the appropriate authentication details to the trusted web site. Thus users can authenticate to internet services without revealing any long time passwords to the un-trusted machine.

The above scheme prevents the un-trusted machine from capturing and reusing the user's passwords and other authentication data, but does not guarantee that the user's password was used for authenticating to the user specified website. For e.g. the user could have sent the one time password for www.amazon.com, while the un-trusted machine could have modified the

destination address to www.buy.com and still sent the valid one time password. The proxy wouldn't be able to detect this modification, and would supply the authentication details for www.buy.com. In order to thwart this attack, the onetime password used for authentication must be dependant upon both the user's secret key and the site the user is trying to access. This can be accomplished using AES as follows.

Initialization Phase:

The user creates a mapping of web sites to authentication details on the proxy. A shared secret key is then computed from user input. Next, the list of servers and the secret key is copied to the mobile device. A counter is maintained on the proxy for each site representing the number of times authentication has been attempted for a site. This counter is initialized to zero during the initialization phase. The proxy and the mobile device do not require secure communication beyond this stage.

Authentication Phase:

When the user wants to communicate with a secure site, the un-trusted machine will send the address of the site to the proxy. The proxy then retrieves and sends the next counter value for the site to the un-trusted machine. The un-trusted machine must return the correct AES encryption of the site address concatenated with the counter value using the secret key. If the two encryptions match, the proxy will supply the authentication details to the web site. Otherwise the proxy will not service the request. In either case, the counter on the proxy will increment. The encryption of the site address concatenated with the counter acts as a onetime password granting access to the web site.

The only way the un-trusted machine can get the correct encryption of the web site address concatenated with the counter is by asking the user to perform the AES encryption using the key shared between the user and the proxy. The user can use the mobile device to calculate this value. If the un-trusted machine attempts to use this one time password to go to a different site it will not match with what the proxy expects and authentication details will not be provided. (It tried to encrypt different data so the encryption won't match.) Similarly, the un-trusted machine can't replay the one time password since, the counter value associated with the site on the proxy would have incremented, and therefore authentication wouldn't succeed.

Protocol:

Initialization Phase:

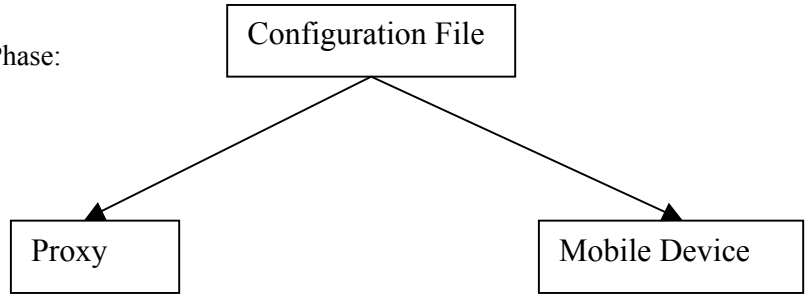

Fig 1. Protocol Initialization Phase

The configuration file contains the user provided pass phrase, which is used to derive the AES key to be shared between the Proxy and the Mobile Device. Also, the configuration file contains the mapping between web sites and user authentication details such as username and passwords.

Authentication Phase:

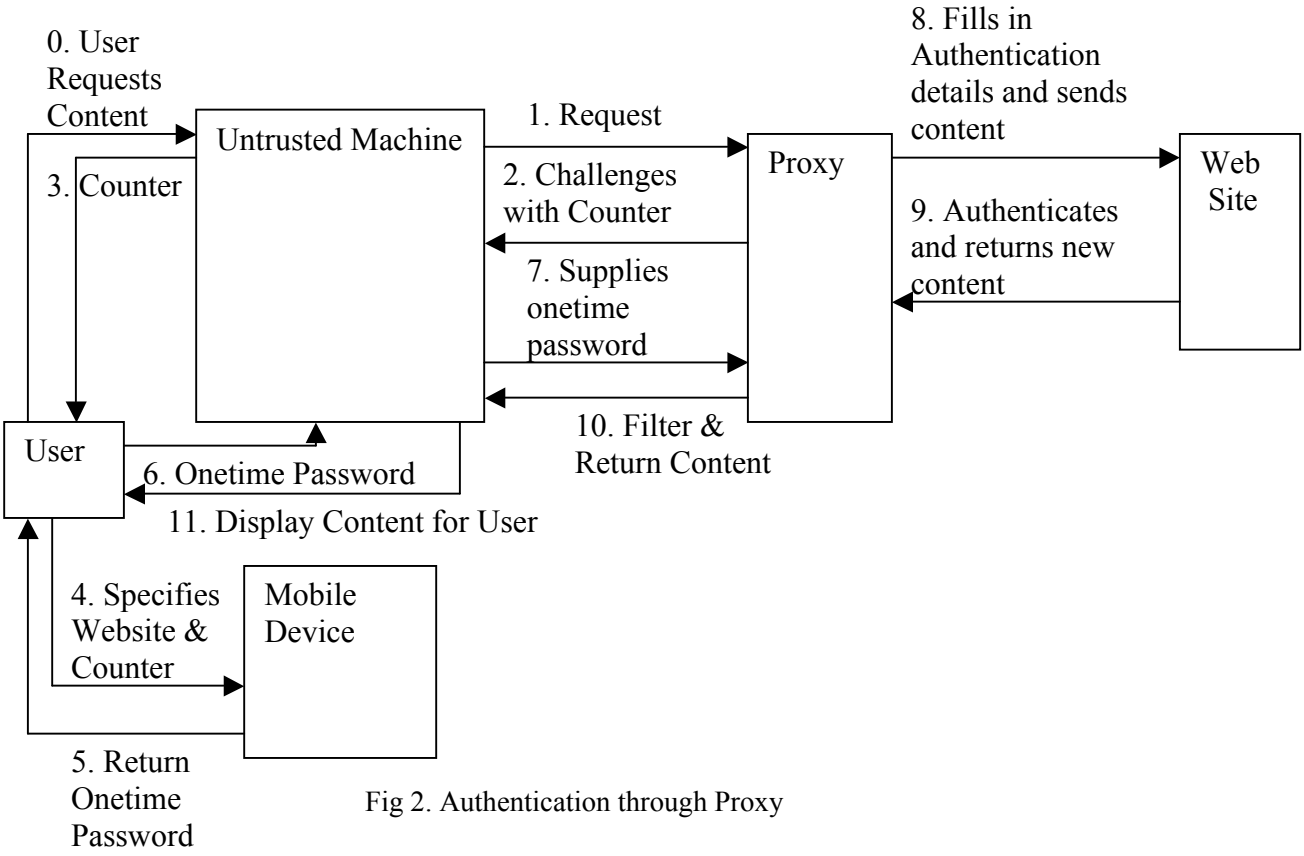

Fig 2. Authentication through Proxy

4.1 Algorithms at the Untrusted Machine, Proxy and Web Site

Notation :

UT – Untrusted Machine
WS – Web Server
REQUEST- Packet sent by UT to Proxy and by Proxy to Web Server to denote request for a site.
CHALLENGE – Packet sent by Proxy to UT in response to a request made by the UT. This packet contains the counter value to be used by the user for AES encryption.
RESPONSE- Packet sent by UT to Proxy containing result of encryption computed by the mobile device.
PROXY_RESPONSE – Packet sent by Proxy to UT containing information to be displayed to the user.
WS_RESPONSE- Packet sent by Web Server to Proxy.

Untrusted Machine:

1) Send user *REQUEST* to Proxy
2) if received *CHALLENGE*
    a. Obtain counter value

b. Provide counter value to user to encrypt
c. Send encryption result to proxy

3) if received *PROXY_RESPONSE*
   a. Display content to user

Proxy:

1) if received *REQUEST* for site X
   a. Check if X is a listed site
   b. If X is listed
      i. send next counter value for X to UT
      ii. update counter value for X
2) if received *RESPONSE* for site X
   a. if request exists for site X
   encrypt site X with sent counter value
   check if received encryption and above encryption match
   if match
      Retrieve sensitive information from database
      Forward request with sensitive information filled in to the web site
3) if received WS_RESPONSE
   a. Strip out sensitive information from received packet
   b. Forward response to UT

Web Server:

1) if received Request
   a. Check authentication for site
   b. If authenticated
      i. Generate response for the request

5. Central Design Principles:

5.1 Least Privilege:

All components in our system are built with the least privilege that needs to be granted for their functioning. The only pieces in the network we added were the proxy and the mobile device. The web servers and untrusted machine already exist, and we grant them no additional privileges. The proxy needs to know the authentication details for sites, because it is the entity to actually send the data to the web server. The mobile device does not need to know this information, and hence is not provided access to user' sensitive information. This prevents long term passwords from being stolen with the physical mobile device. The mobile device can be password protected to provide additional security in case of theft.

Untrusted machines do not have the right to know authentication details for web servers, thus they are never provided these details. Web servers only have the right to authentication details for their site, and they are never sent the authentication details for another site from the proxy.

5.2 Economy of Mechanism:

Our design has very few parts. We have the proxy, which mediates access, and the mobile device, which helps users remember their one time passwords.

5.3 Open Design:

There is nothing secret about our design. We have provided diagrams detailing our communications protocol. We have described step by step how one time passwords are generated and checked using a well known respected cryptographic algorithm with an open design.

5.4 Complete Mediation:

Authentication details stored on the proxy are only supplied to their associated web servers when the requestor gives the correct response to the one time password challenge. Authentication details can not be accessed through any other mechanism: Authentication details are private to the proxy.

5.5 Permission Based:

The default response when the proxy receives a request to supply authentication details to a web server is no. Only when a requestor has proven their right to authenticate with a web server, by passing the onetime password challenge, are authentication details supplied to that web site.

5.6 Separation of Privilege:

Authentication to the proxy is based upon two conditions. Instead of the standard user name and password we authenticate to the proxy with onetime password and site URLs. These onetime passwords are dependant upon the site URLs, the counter value, and the secret key, therefore using a onetime password with the wrong site URL will not authenticate the user to the proxy, consequently authentication details will not be supplied to the web server.

5.7 Least Common Mechanism:

Proxies, untrusted machines, web servers, and mobile devices all run on separate hardware, and hence share the least common mechanism to support their operations. The only information channel between web servers, proxies, and untrusted machines is the network. This is the bare minimum required for web browsing. The only communication channel between the untrusted machine and the mobile device is the user: The untrusted machine and the mobile device in general need not have a direct communication channel.

5.8 Ease of Use:

Using our one time password system is very easy for end users. The end user needs only to connect his mobile device to the proxy and run the one time password initialization program. This is trivial. The user then specifies, optionally, a password for the mobile device to protect it against theft. When attempting to authenticate from the

untrusted machine for a web site, the user enters his password to the mobile device (if he chose to protect it) selects his web site from a list, which is strikingly similar to a web browser's book marking system, and inputs the counter value provided to him in the challenge. The user then enters the values displayed on the mobile device into the password field of the web page like he would enter any other password. Basically, all the user does is log in to his device, use a pull down menu, type in the counter and then type the generate password into the untrusted machine. While this might take twice as many keystrokes than a normal login it is actually much easier for users because they don't need to remember their passwords.

6. Cipher Selection Rational:

We choose to use AES to generate the onetime passwords because AES is a well known, and well studied cryptographic algorithm. A fundamental concern with our approach is that, since the attacker knows both the counter value and the site, the attacker knows the plaintext used to produce onetime passwords. If we had chosen a cryptographic algorithm that was weak against know plaintext attacks that would be a serious flaw in our design. However, AES is not vulnerable to known-plaintext attacks. The selection process determining what cipher was selected to be the AES involved extensive cryptanalysis, which would presumably have found any vulnerability based upon known plaintext attacks. Rjindael etal [7] claim that "Obtaining information from given plaintext-ciphertext pairs from other plaintext-ciphertext pairs can not be done more efficiently that by determining the key by exhaustive key search".

Since successive calls for authentication are based upon the encryption of the site name which does not change, and the counter, which is incremented by one, it is very important that our system use a cipher with good diffusion. Every other authentication attempt is just one bit different than its predecessor, making the diffusion of this bit critical. This problem is common to any cipher run in counter mode but still merits investigation. AES has both theoretical assurance and empirical evidence of high diffusion. According to [7, page 8] AES has high diffusion since

> "In most ciphers, the round transformation has the Feistel Structure. In this structure typically part of the bits of the intermediate State are simply transposed unchanged to another position. The round transformation of Rijndael does not have the Feistel structure. Instead, the round transformation is composed of three distinct invertible uniform transformations, called layers. By "uniform", we mean that every bit of the State is treated in a similar way.
>
> The specific choices for the different layers are for a large part based on the application of the Wide Trail Strategy [Da95] (see Annex), a design method to provide resistance against linear and differential cryptanalysis (see Section 8.2). In the Wide Trail Strategy, every layer has its own function:
>
> **The linear mixing layer**: guarantees high diffusion over multiple rounds.
> **The non-linear layer**: parallel application of S-boxes that have optimum worst-case nonlinearity properties."

Rijndael follows up these assertions with a proof for "**Theorem 2:** Any trail over four rounds has at least 25 active bytes." [7, page 35] Since AES uses at least 10 rounds, this provides high diffusion.

These theoretical results are backed by Hellekalek's [6] battery of tests showing AES's high diffusion in practice. Hellekalek found that "Although no theoretical equidistribution analysis is available for AES, the empirical findings suggest the use of AES in the Counter Mode as a non-linear high-speed random number generator." [6, page 11] Hellekalek used two test setups. The first test was based upon "an idea introduced in Wegenkittl [2001b] and Wegenkittl and Matsumoto [1999] called dimension reduction (see also L'Ecuyer and Simard [1999] for a related test statistic). We map d-dimensional overlapping tuples to 3 distinct states … The three states here are motivated by a gambling game in Wegenkittl [2001b], the resulting test is, therefore, called gambling test." [6, page 9] In test 2 they " Vary $d \in \{32, 64, 128, 256\}$ and $n \in \{2^{22}, \ldots, 2^{28}\}$. Compute 16 independent repetitions of the gambling test on the bit stream $(y_i)_{i \geq 0}$ with parameters d and $t = d/2$ and the 1-sided Kolmogorov–Smirnov p-value of their empirical distribution." [6, page 10] These statistical tests show that AES should work well as a random number generator when used in counter mode. Since our one time passwords are generated in a similar manner, this leads us to believe that although every other counter varies by only one bit, successive one time passwords are sufficiently diffuse enough to foil attackers.

7. Implementation and Performance Study

We have developed a prototype system in Java using Java's cryptographic libraries. We provide a software emulation of the mobile device. All sensitive data is stored in a MySQL database at the Proxy. The Proxy is assumed to be safe and would in reality be safeguarded against all known database attacks. We also emulate a web server which in our implementation just receives packets from the proxy and checks with its database if the supplied username and password are valid.

The mobile device and proxy are initialized at system boot up. The initialization phase consists of

Shared Key Establishment:

A secret key is shared between the proxy and the mobile device. The user selects a password and the MD5 digest of the password is used as the shared key hence forth.

Loading Sites for future Use:

Since the mobile device will need to provide AES encryption of the web sites the user requests for, the mobile device and proxy are preloaded with possible sites the user would be using on an untrusted machine. Any access to sites not in the initial list would not be serviced by the proxy. All additions to the URL list must be performed at boot up.

7.1 Security Analysis

We tested our system and our system thwarts the following attacks

Replay Attacks:

One of the major issues in current authentication mechanisms is that passwords are re usable. Therefore a malicious user could capture user passwords and replay them at a later time and still gain access to the system. In our system, we use one time passwords which are generated through the AES encryption of the site address concatenated with the

counter value. Every time a request is made, the counter is updated and therefore the AES encryptions for successive requests will be different. This prevents replay attacks in our system. Also, in our protocol we ensure that when the proxy receives a response from the untrusted machine for a web site, a request had previously been made for the site.

Source Authentication:

There is source authentication in our system through the use of one time passwords. When the proxy receives a valid response, it must have come from only the user since only the user is assumed to know the shared key. Also, our system is flexible in that, other ways of source authentication such as SSL could also be applied in addition.

URL Redirection:

Our system ensures that there are no successful URL redirection attacks. This is accomplished through the one time password generation scheme. We use AES in OFB mode for encryption. The shared key is used as the AES key while the IV for AES is chosen as the concatenation of the site address and counter value. Thus, if the user supplied address and the address sent by the UT differ, the password computed by the mobile device would not match that computed by the proxy, and the proxy would not service the request.

7.2 Location of Proxy

The proxy introduces a delay in the response to a web request and the location of the proxy plays a dominant role in characterizing the delay. We tested our system with the following scenarios

i. proxy and the untrusted machine on the same machine (Co-located)
ii. proxy and the untrusted machine on different machine (separated)

Proxy and Web Server were always collocated.

We measure response time as the time elapsed between a request made and the corresponding response received at the untrusted machine.

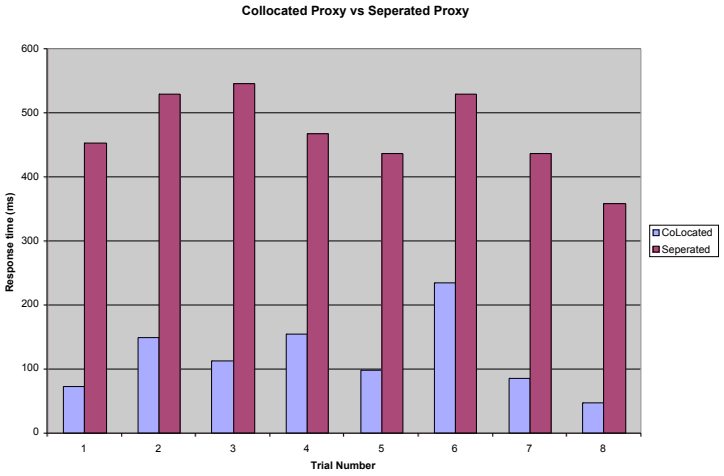

Fig 3. Collocated Proxy vs. Separated Proxy

The average response time with the co-located configuration was about 120 ms, while with the separated configuration; the average response time was about 470ms.

8. Conclusion

Securing web access from untrusted machines is an interesting problem to solve. Traditional techniques propose a proxy based solution. However these approaches require constant out of band communication between the user and the proxy, which may not always be plausible. Our proxy based solution does not require constant communication between the user and the proxy. Instead, the user carries a mobile device capable of performing cryptographic operations to generate one time passwords. These passwords provide authenticate for requests, to the proxy, to supply sensitive data to targeted web sites. We have developed a prototype system and tested it to show that our system thwarts replay attacks and URL redirection attacks, in addition to guarding sensitive data from untrusted machines.

References:

[1] M. Wu, S. Garfinkel, and R. Miller, "Secure Web Authentication with Mobile Phones," Student Oxygen Workshop, MIT Computer Science and Artificial Intelligence Laboratory, 2003.

[2] Delegate:A Proxy Based Architecture for Secure Website Access from an Un-trusted Machine**,** Ravi Chandra Jammalamadaka; Timothy van der Horst; Sharad Mehrotra Kent Seamons; Nalini Venkatasuramanian; 22nd Annual Computer Security Applications Conference (ACSAC), Maimi, FL, December, 2006

[3] Dwaine Clarke, Blaise Gassend, Thomas Kotwal, Matt Burnside, Marten van Dijk, Srinivas Devadas, Ronald Rivest. "The Un-trusted Computer Problem and Camera-Based Authentication" Pervasive Computing : First International Conference, Pervasive 2002, Zürich, Switzerland, August 26-28, 2002.

[4] S. Ross, J. Hill, M. Chen, A. Joseph, D. Culler, E. Brewer. A Composable Framework for Secure Multi-Modal Access to Internet Services from Post-PC Devices, IEEE Workshop on Mobile Computing Systems and Applications, December 2000.

[5] Vipul Gupta, Sumit Gupta. "Securing the Wireless Internet", IEEE Communications Magazine, December 2001.

[6] Empirical Evidence Concerning AES. Peter Hellekalek , Stefan Wegenkittl. University of Salzburg. ACM Transactions on Modeling and Computer Simulation, Vol 13. No. 4, Pages 322-333, October 2003.

[7] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," 2001.

[8] Charles P. Pfleeger, Shari Lawrence Pfleeger, "Security in Computing" , 3rd ed., Prentice Hall, 2003, ISBN: 0130355488.