ENHANCING CYBER SECURITY OF ONLINE ACCOUNTS VIA A NOVEL PROTOCOL AND NEW TECHNIQUES

Mehrdad Nourai and Haim Levkowitz

Computer Science Department, University of Massachusetts Lowell, Lowell, MA, USA

ABSTRACT

The financial world has gotten more sophisticated. People need to make informed financial decisions, so they seek out efficient tools to help them manage their finances. Traditionally, money management software has been available for individuals to use in their homes on their personal computers. These tools were a local install, often expensive, and required a learning curve to use them effectively. With a paradigm shift to cloud computing and storage, users are looking for inexpensive alternatives that are accessible at home or on their mobile devices. As a result, third-party companies have been forming over the last few years to meet this need. However, to access the functionality of these online resources, users are required to divulge their personal financial account login credentials. While third-party companies claim that subscribers' private information is safely stored on their servers, one cannot ignore the fact that hackers may be able to break into their system to steal users' information. Once hackers manage to compromise users' login credentials, they have complete control over their accounts. Therefore, there is a need to have a holistic approach that incorporates security elements to protect users' accounts from hackers.

We present a novel, holistic model with a new handshake protocol and online account access control, which authenticate account access and form a sandbox around third-party access to users' accounts. When utilizing these novel techniques, users' login credentials can remain private, providing safeguards against unauthorized transactions on their accounts.

KEYWORDS

Cyber Security, Network Protocols, SSL, Cryptography, PKI

1. INTRODUCTION

In today's fast-paced world, personal financial accounts are accessible online, and often they tend to be with a different institution. When one needs to figure out their net worth or track what is happening with their money, they start by logging into each of their accounts individually. The user must be able to remember their login credentials for each account. Ideally, for good security practices, each account should have unique login credentials. However, as a convenience, users' may use one set of credentials for most (if not all) of their accounts. Once the user logs into their account, they need to download account information in the proper format and import it to local install financial software (e.g., Intuit Quicken). Although the use of these tools is an improvement over tracking financial details by hand, this process can be tedious, time-consuming, and may become overwhelming for users that are not familiar with the world of finances or computers. There are usability issues and inconveniences with locally installed applications. For instance, software localized to one computer requires maintenance and updating to avoid security vulnerabilities. The lack of mobile accessibility is also prohibitive for today's world where everyone is in a state of perpetual motion. Also, frequent redesign of the software creates confusion and frustration over new interfaces and functionality for new editions of the software. This model has a steep learning curve, and although it may have been a sufficient form of financial aggregation years ago, it is no longer the case. Thus, it is not surprising that users are migrating to online tools for managing their personal finances.

DOI: 10.5121/ijnsa.2017.9302

New independent online companies are emerging that have no relationship with financial institutions to offer their financial aggregation tools to users. In this paper, we refer to such enterprises as "third-party companies." The idea behind these businesses is to provide a set of budgeting features that were previously offered by the locally installed financial software, but with the added advantage of the third-party doing all the work for little to no cost. For third-party companies to provide these services, they need to login to users' online accounts to read and collect information. These third-party companies utilize algorithms when they critically examine account transactions, net worth, and other details. As a result, they create an aggregate report that presents an analysis of accounts' information in a graphical or a textual manner. Users prefer the method of automatically gathering and aggregating account information rather than the old practices, which required them to perform all the work themselves.

Although an online budgeting tool is a convenient and affordable tool, some users may shy away from using it. Users have security concerns or feel vulnerable when they sign up for such a service. The vulnerability begins when users give their private accounts' login credentials to third-party companies. If an attacker manages to compromise a third-party's network, they have the entire financial lives of all users in their hands. This security vulnerability is due to the fact that when someone logs into an account, they are able to do everything that the owner of the account can do there. That is, the breach extends past viewing personal information to the ability to perform transactions on the account as well. The main idea of this paper is to introduce a novel and holistic login architecture and a new sandboxing technique for securing online accounts by providing an entirely new separate set of login credentials with lower permissions for third-party utilizations. We explain precisely the proposed techniques, which can protect online accounts from potential fraudulent activity when users utilize services offered by third-party companies.

The main contributions of this paper are:

- To introduce a new handshake protocol for use by third-party companies to authenticate access to users' online accounts (discussed in Section 4.2).
- To introduce a new granular access control layer sandbox techniques for fine-grained access to users' online accounts (discussed in Section 7.4).

2. EXISTING PRACTICES AND INFRASTRUCTURE SHORTCOMINGS

We now discuss the present practices and their weaknesses, which create cyber security vulnerability for users.

2.1. USER AUTHENTICATION VIA USERNAME AND PASSWORD

For financial institutions to provide a secure online mechanism for their customers to access their accounts, financial institutions utilize the HTTPS (HTTP over SSL) protocol, which can be used via a web browser to access their account. The current process is as follows: the user opens a web browser on their computer, types in "https://" followed by the website address of their financial institution. Once the user hits the Enter key, the communication between the user's computer (the client) and the financial institution's computer (the server) starts. The communication first establishes a secure channel between the client and the server by utilizing encryption techniques and the SSL handshake protocol, and then HTTP commands can flow in the secure channel. The server can now send the home page to the client, which may contain an area within the page to obtain the customer's login credentials or may have a sign-in link to open a login page. The mechanism used for inputting the account credentials utilizes HTML FORM INPUT tags via the POST method.

When the home or login page opens in the browser, it provides two textboxes, one for the username and one for the password plus a sign-in button (e.g., Figure 1). Once the user inputs the necessary fields and clicks the sign-in button, the server starts the user authentication process.

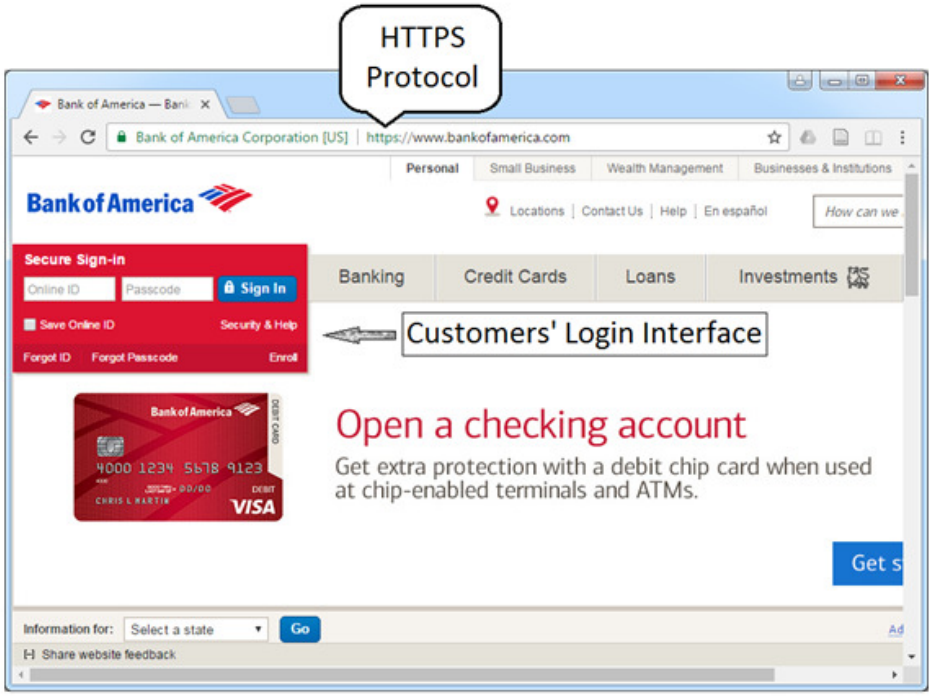


Figure 1. An example of a financial institution's home page

The financial institutions' server converts the customer's password into a custom hash value using the same algorithms they used at sign-up to create the original hash value. The server then checks the hash values for a match. If the hash values are matched, the server allows access to the customers' account; otherwise, the server is likely to display an error message after a failed login attempt. Based on the financial institution's policy, the server may allow a few more attempts before locking the account to protect it from attackers.

2.2. DEVICE VERIFICATION

In addition to the process of checking users' password as a means of authentication, financial intuitions often ask for other forms of verification. Especially when the user is attempting to log into their account from a device that they have not used before with their account. The extra authentication step involves one or more security questions that the user had already provided answers to when they first set up their account's credentials. The extra verification is an added security measure to ensure that even if an attacker has obtained the password, the attacker has to overcome another layer of security before being able to gain access to the account.

The technology often used for the user's device verification is via the browser's cookies (e.g., Figure 2). Cookies are small text files that are stored on the user's computer to save specific information related to the use of a particular website. This is because the web is a stateless medium; hence, cookies are the most popular way to keep track of information while users

browse a website. Financial institutions utilize cookies and use them to save specific information on users' devices to verify that the access is coming from the account's real owner rather than an attacker trying to gain access to someone's account.

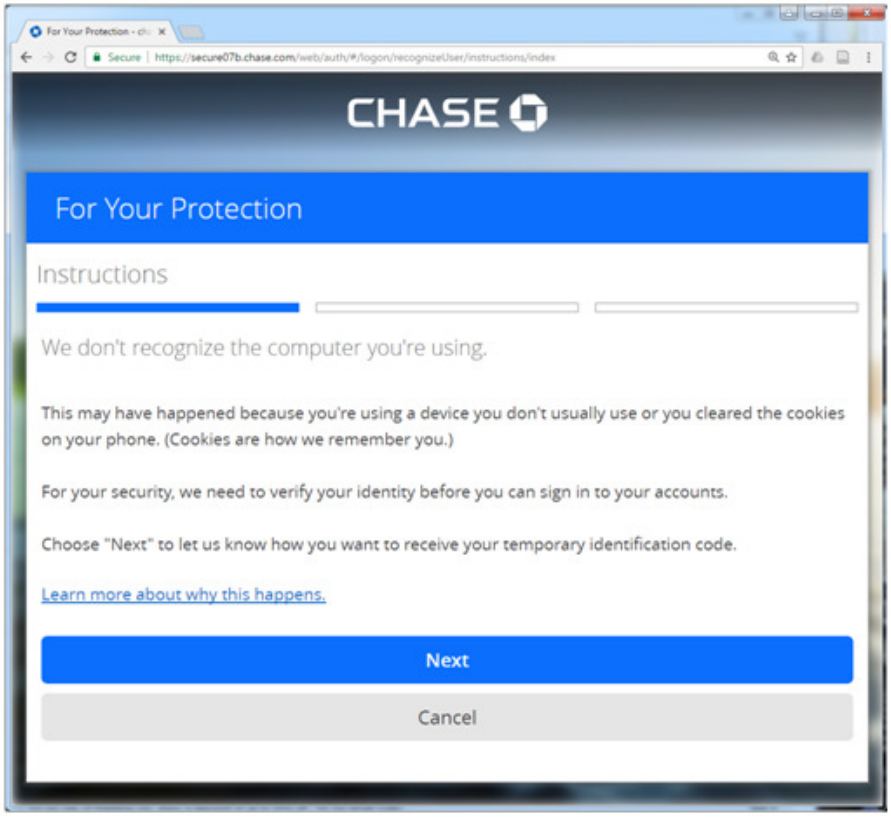


Figure 2. An example of a new device verification

2.3. VARIATIONS IN AUTHENTICATION

With the current practices, there are variations to the authentication steps, and not all financial institutions follow a standard mechanism for authenticating their users' access. For instance, some may display a predefined image for identification of the legitimate versus a fake website that was made to look like their financial intuition's website. Others may ask for the username on the home page but not password at first, until the user clicks the Sign-in, Continue, or Next button. There is also a second authentication process utilizing a code that is delivered via email, text message, or a mobile app. In general, the extra authentication steps may consist of some other information that the owner of the account knows in addition to their password, which enables financial institutions to perform additional security measures before allowing access to an account. Therefore, a hacker would have to break down at least two barriers to gain access to the account. While this process works well in practice, developers designed it for humans' capabilities, not for machines. Hence, financial institutions had to make their interfaces easy enough for human use, while appealing to the masses that use online services. That is, current login credentials are safe enough but not too cumbersome to become an inconvenience for users to circumvent the intended security.

2.4. CURRENT INFRASTRUCTURE SHORTCOMINGS

The present infrastructure of online accounts lacks the mechanisms to support a different form of account authentication with restrictive access. As a result, users are giving their personal access credentials to third-party companies to utilize the financial services they provide. Ever-increasing cyber security threats, coupled with existing industry practices make users vulnerable to cyber criminals. Current practices have created a challenging and engaging problem that needs to be solved to keep users safe from potential cyber-attacks.

The following is a list of possible problems with the current infrastructure:

- Users are making themselves vulnerable by giving their complete credentials in the form of username/password plus security questions and answers to third-party companies.
- Users' credentials are designed to be private and not shared with others.
- Once users' credentials are given to a third-party company, they are stored on their server, which may be exposed to an attacker.
- Users may use the same username/password for other accounts and in other places and, as a result, hacked or stolen credentials can be used to access multiple accounts.
- Current bank accounts are full access accounts and, as a result, once a third-party company has access to these accounts, they can perform transactions on that account.
- Financial institutions are not the only companies that always allow full access to online account once users enter their security credentials. Therefore, other online accounts that users share with third-party companies may be at risk to cyber-attacks.

3. NETWORKING INFRASTRUCTURE

In this section, we discuss the foundation of the networking infrastructure that our new protocol utilizes to deliver the needed security.

3.1. SECURE CHANNEL

Secure channels between two parties are required when the transmitted information is sensitive and private while traveling over an insecure medium (e.g., the Internet). In the next section, we discuss the current practice referred to as SSL, which secure a channel for private communications.

3.2. SECURE SOCKETS LAYER

The SSL (Secure Sockets Layer) protocol is a secure connection technology used on the Internet for securing communications between a client and a server. SSL has been updated and renamed TLS (Transport Layer Security). Although TLS is the next generation of its predecessor, SSL, the term SSL is prevalent, and therefore it is used throughout this document.

The SSL protocol was originally developed by Netscape Communications in 1994 to address security issues of communicating via the Internet [1]. The protocol was a revolutionary technology for securing Internet traffic that carried personal or sensitive information. SSL is over two decades old and has evolved over time to be more secure. The SSL architecture continues to change over time as new SSL vulnerabilities are found, computers become faster, and demands for tighter security increases. The workings of SSL depend on trust models provided by Certificate Authorities (CA) and public key infrastructure that is based on cryptography. The SSL underlying mechanisms protect the transmitted information's integrity and security by providing

authentication of the server to the client, and optional authentication of the client to the server. Although SSL has several security features built-in, it is not immune to cyber-attacks (discussed in Section 8.2). The new proposed protocol leverages SSL and its ability to secure the communications between the client and the server without any changes to this layer.

3.3. TRANSPORT LAYER

The Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are located in the transport layer. TCP is a connection-oriented protocol and has three-way handshakes (Figure 3) to establish a connection between the client (Initiator) and the server (Receiver). TCP is the most reliable and prevalent protocol on the Internet because it guarantees packet delivery, ordering, and congestion control. UDP is a connection-less protocol that does not guarantee packet delivery or ordering; it is typically used for streaming applications. TCP, along with the IP layer, make up the TCP/IP protocol, which we use as our transport layer protocol. The only change in this layer is a minor one, where TCP flags might be set to flush out the packets.

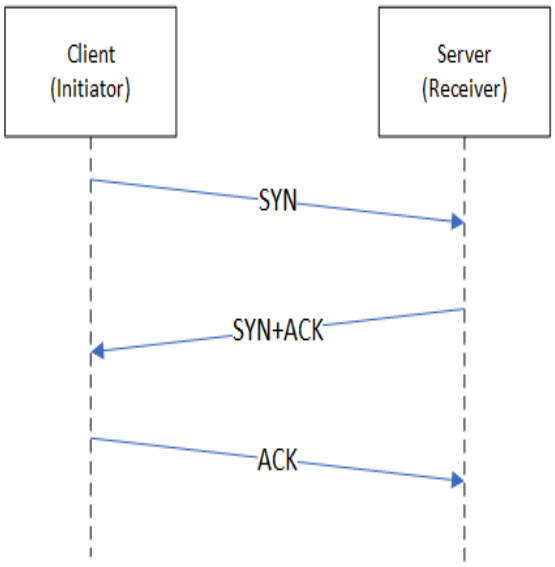


Figure 3. TCP/IP Three-Way Handshake

3.4. SECURE VERSUS INSECURE NETWORK PORTS

The TCP and UDP transport layer protocols have 65535 ports each. The Internet Assigned Numbers Authority (IANA) assigns the ports for specific protocols and general use [2]. Although some ports are used for secure protocols, there are no “secure” or “insecure” port numbers. The traffic that flows through any network port can be encrypted or in clear text. Hence, it is up to the underlying protocol to use them as “secure” or “insecure” ports. Nevertheless, there are benefits in standardizing assignments of default ports for “secure” or “insecure” traffic. The assignment can reduce the confusion or errors in using certain common ports for secure communications while setting up firewall rules.

4. APPLICATION LAYER PROTOCOLS

In this section, we discuss the current HTTPS and its issues and then present our new HTTPAS protocol, which addresses security concerns of current practices when used with third-party companies.

4.1. HTTPS PROTOCOL

The Internet is an open and insecure medium for communication. When a connection between a client machine and a server is established, all the information transferred over the Internet is traveling over an insecure connection. This information can be intercepted and be seen by others. Therefore, sensitive information (e.g., username/password, bank account information, medical records) must be protected while the information is in transit. The Hypertext Transfer Protocol (HTTP) that is primarily used for accessing information via the web moves data in clear text, which makes it vulnerable to an attacker. To protect data going between a client and a web server, a protocol called HTTPS is used. HTTPS consists of HTTP over the SSL protocol, which encrypts and secures the communication to protect it while in transit [3]. The HTTPS protocol works flawlessly behind the scene, and it performs its tasks without user intervention unless something goes wrong with the connection. When HTTPS finds an issue with the connection, it alerts the user and allows the user to decide on the next step. When a user accesses a website using HTTPS with their browser, and if it supports the HTTPS protocol, a green lock icon is shown as part of the link in the browser. Once the lock is clicked, information about the security and connection are displayed. For instance, details can show the connection information as TLS 1.2 AES_256_GCM RSA (2048), which is the version and cryptography specifications of the connection. The HTTPS protocol was designed for human-to-computer interactions to secure sensitive online web activity while users are accessing their online account. The HTTPS protocol can be kept as is, and still be used when humans need to access their accounts. In the next section, we introduce a new, novel protocol, which is suited for computer-to-computer communications.

4.2. INTRODUCING THE NEW HTTP AS PROTOCOL

While HTTPS was designed for human-to-computer interactions, for greater cyber security another protocol is needed for computer-to-computer interactions. We have designed a new, novel Application Layer Protocol called HTTPAS for computer-to-computer interactions. HTTPAS's architecture consists of HTTP with a new Authentication Handshake Protocol (AHP) over SSL. This new protocol utilizes SSL to secure the communication channel between a third-party's computer and a financial institution's web server. The novel protocol uses the new AHP for the two computers to negotiate and authenticate secure access to users' accounts. The motivations for a new protocol are flexibility, extra security, and custom enhancements that the current HTTPS does not offer. The HTTPS protocol was designed to be a general multipurpose protocol for providing secure communication channel on the web. This protocol is often used for human-to-computer communications, which require more conveniences for a human user. Therefore, security versus convenience for the human user was the compromising factor.

The new, novel HTTPAS protocol closes this gap by offering a feature set that is well suited for computer-to-computer communications. This protocol can also be adapted for human-to-computer communication, however, due to extra security steps, it would require more efforts on the human side. Our new approach increases the security to another dimension, which utilizes a public key infrastructure (PKI) framework to enhance and improve the security of the protocol.

This method can eliminate the need for third-party companies to request and use a user's username and password combination plus other login credentials while offering extra and better security not found in current practices. We explain components of this new protocol (discussed in Section 5) in details later in the paper.

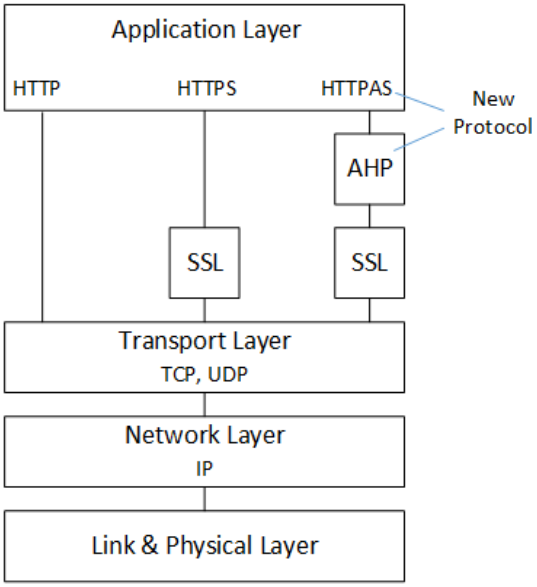


Figure 4. The TCP/IP stack with the new HTTPAS and the new authentication handshake protocol

The diagram in Figure 4 shows HTTPAS within the realm of the TCP/IP network stack. The new protocol consists of new Authentication Handshake Protocol (AHP) (discussed in detail in Section 5). The AHP protocol authorizes the client's computer and authenticates access to users' accounts. Like HTTPS, the new HTTPAS protocol uses SSL as its foundation for a secure communication channel. Using existing SSL protocol reduces or minimizes the risk of introducing new vulnerabilities. Utilizing existing proven technology as the foundation of the new protocol is consistent with security practices and philosophy of some researchers that may argue that creating new cryptography may introduce new vulnerabilities.

The following are the benefits of using HTTPAS for third-party access:

- The solution we are envisioning would result in better security practices that address concerns of existing and potential new users. The existing users get the better security. The new users that were hesitant to sign-up with a third-party due to security issues of such services can now be sure that the third-party cannot perform any transactions on their accounts. Having better cyber security can potentially increase the size of the user base utilizing third-party services, which can benefit all parties involved, i.e., users, banks, and third-party companies.
- Users' don't have to give their banking credentials, which can be in the form of username/password plus security questions and answers to a third-party site, i.e., their credentials which are meant to be private, can stay private and not shared. Instead, the

third-party can utilize the new handshake protocol and access control for accessing users' accounts.

- Often users have the same username/password for more than one online account. As a result, once an attacker steals their credentials from a third-party's server, then the attacker can get access to those accounts in other places, which can become a major security issue for the users.
- The solution is not limited to banking websites, as it can be adapted for online accounts, such as email, which utilizes usernames/passwords for their authentication. A third-party company can access these accounts on a read-only basis.
- If an attacker compromises a third-party's server and access information is stolen, the attacker cannot perform transactions on the account. Also, the bank and the owner of the account can easily revoke the access and generate a new access protocol, which can be safer and more convenient than with current practices.

4.3. PORT NUMBERS

To place the separation of traffic between the current and the new protocol, as well as minimize or eliminate any changes to the existing protocols, HTTPAS uses a different TCP/IP port number than HTTPS. The current application layer TCP/IP protocol port assignments for existing application protocols have been designated by IANA, which are as follows: port 80 for HTTP and port 443 for HTTPS. The new HTTPAS protocol does not currently have a designated port assignment. Hence, in the interim, we use default port 10443 which according to the IANA online registry search tool, has not officially been assigned to any protocol.

5. NEW AUTHENTICATION HANDSHAKE PROTOCOL

The new Authentication Handshake Protocol (AHP) utilizes public key infrastructure (PKI) framework, TCP/IP as its transport layer and SSL as its transport layer security for its underlying mechanisms. AHP is the main component and workhorse behind the new HTTPAS protocol and is responsible for authorizing and authenticating the access to users' accounts. In this section, we now describe the precise details of the new protocol.

5.1. CIPHER SPECIFICATION

As mentioned in the last section, we use the secure communication channel that SSL provides as the foundation of a secure connection. SSL negotiates Cipher Spec between client and server. Therefore, for performance reasons, we leverage the negotiated Cipher Spec to minimize unnecessary handshakes between the client and the server.

5.2. SEQUENCE OF EVENTS

AHP is responsible for the following sequence of Events:

- Negotiate AHP version to use for the handshake
- Obtain user account id that the client is requesting access for
- Obtain the client's computer certificate and public key
- Verify the client's computer certificate with certificate authorities
- Verify the user account has been authorized for access from the client's computer
- Authenticate the access utilizing challenge/response and pass-phrase
- Grant or deny access to user's account

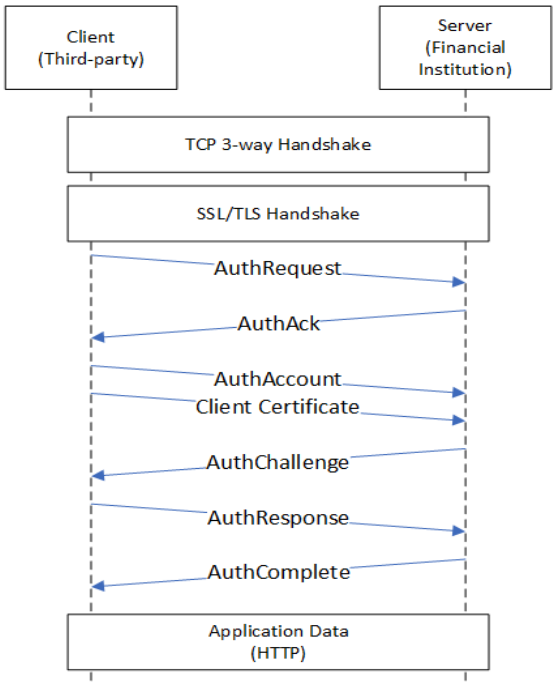


Figure 5. Authentication Handshake Protocol (AHP) Sequence Diagram

5.3. SEQUENCE DIAGRAM

The diagram in Figure 5 shows the new AHP handshake protocol between the client and the server. It also shows its placement within the timeline of the TCP and SSL handshake protocols which are needed for healthy and secure connection. The new handshake consists of several command components as follows:

- AuthRequest
- AuthAck
- AuthAccount
- AuthChallenge
- AuthResponse
- AuthComplete

We now describe each of the new handshake's elements in detail.

5.3.1. AUTHREQUEST

The first step of the new AHP handshake protocol is AuthRequest command. This action occurs right after completion of SSL handshake protocol, which secures the channel for communication. The client starts by sending a request for authentication by providing a list of AHP protocol version numbers it supports in a string (listed in order of preferred version numbers). The line must contain a comma separated displayable characters, with two zeros as terminating character. The purpose of exchanging version numbers between the client and the server is that in case

either of the party does not support the latest version of the protocol, they can negotiate on which version to use for a successful handshake.

5.3.2. AUTHACK

The server checks to make sure the version numbers are acceptable and respond to the client with a version number string of its own (similar string format as client's string). The client adjusts the protocol to the server required version number. Note that, for security reasons, only the server can decide on what AHP protocol version number to use. If the server does not support or agree on the client's desired version number, the server terminates the connection.

5.3.3. AUTHACCOUNT

The client provides user account id along with its CA certified digital certificate and public key. Once the server receives the necessary information, it checks the client's certificate with certificate authorities for authenticity and extracts domain information. Now, the server must check to ensure that the user account id exists and that the client has the prior authorization to access the account. The server terminates the connection if any of these steps fail to complete successfully. Note that although SSL provides an optional client verification, we perform the client verification as a mandatory step here. This extra step ensures that even if the user account's private key has been compromised, the key would not be accepted from an unauthorized client.

5.3.4. AUTHCHALLENGE

The server creates a challenge string made up of an arbitrary length of random numbers, encrypts it with the user id's public key and sends the challenge string to the client. The random number can prevent a forward attack, as the challenge string cannot be reused for future authentication.

5.3.5. AUTHRESPONSE

The client must first decrypt the user account's private key utilizing a pass-phrase, then decrypt the challenge string and send it to the server.

5.3.6. AUTHCOMPLETE

This AuthComplete is the final step in the AHP handshake protocol. The server checks the decrypted string for a match. When there is a match, the server returns AHP_SUCCESS code. Otherwise, returns AHP_FAILED code to the client and terminates the connection. Once the authentication is successful, the HTTP request/response commands can be sent between the client and the server.

6. TRUST MODEL

To identify a server or a client machine that we want to communicate with, a trusted source is needed to verify the identity of either of the party. We now discuss that next.

6.1. CERTIFICATE AUTHORITY

Certificate Authority (CA) is a trusted entity (e.g., VeriSign, GeoTrust, Comodo, DigiCert) that issues digital certificates for authentication of computer systems on the Internet. CA validates the originator of the request for a digital certificate before issuing the certificate. The concept of using a CA is similar to trusting government institutions that issue driver's licenses that are often used in verifying the identity of individuals.

6.2. CERTIFICATE

Certificates are based on X.509 standard [4] and are digital documents which are generated and issued by a CA. Certificates are made utilizing the public-key cryptography (PKI) which enables parties communicating over the Internet/network to validate the identity for establishing a secure connection [5]. Digital certificates can be purchased from CAs for an annual fee or through other purchase plans offered by a particular CA. Certificates can also be locally generated (i.e., self-signed certificate) which are typically used for development and testing purposes. We require that third-party companies purchase digital certificates from their desired CA and deploy them on their computers for using the AHP handshake protocol.

6.3. CERTIFICATE REVOCATION LISTS

To identify the validity and status of a digital certificate (e.g., revoked or expired), CA keeps a record of those certificates in a list called the Certificate Revocation Lists (CRL) [6]. The AHP protocol checks the CRL list to ensure the validity of the digital certificate. This is a critical step in the AHP protocol authentication process to ensure that the client or server meets the identity requirements of the CA.

7. ACCESS CONTROL

The term access control defines the system protection mechanism for granting or denying access to the available resources. The access control ensures that persons or services that request access to a resource have prior authorization for using the resource. The resource can have (but not limited to) the protection attributes such as read, write or modify, and delete. The design of protection mechanism needs to accommodate each type of system that requires protection, and it can vary widely from system to system. [7]

7.1. PRINCIPLES OF LEAST PRIVILEGE

The best approach to giving authorization to a resource is to use the concept of giving the least privilege, i.e., giving what is needed to use a resource and restricting everything else [8]. Utilizing this principle can limit what action can take place on a resource, which can minimize data leaks or misuse.

7.2. ACCESS MODELS

Access control security can be designed for infrastructure protection via one or more of the following security models [9] (listed in alphabetical order):

- Access Matrix Model
- Bell-LaPadula Confidentiality Model
- Clark-Wilson Integrity Model
- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Role-Based Access Control (RBAC)

These security models provide an array of protection and access control for a variety of real-world applications, such as operating systems, databases, and web systems. This paper is interested in the Role-Based Access Control (RBAC) model since it provides the necessary

granular access for securing online accounts. The RBAC has been used for protecting a variety of domains for government and commercial infrastructure [10]. We utilize the RBAC model to define new roles and types of access for particular web objects to provide added security for online accounts.

7.3. CURRENT ACCESS STRUCTURE

With the current account access paradigm, when a third-party company logs into users' online account at their financial institutions, they get the same privileges as the account's owner, i.e., full privilege access to users' accounts. The third-party having the same access level as the owner of the account is the by-product of using the same authentication mechanism for logging into a user account, which makes it difficult to distinguish access made by a third-party company versus the account's owner. As a result, financial institutions may not be able to offer any account access control for their users, unless they can differentiate between accesses made by each of the parties. We believe that no one other than the designated account's owner should have full privileges to users' accounts, even after successful login to the account. Therefore, the third-party access must be limited and sandboxed to allow the display of account information or authenticate the account access, without any other privileges afforded to the account owner.

7.4. GRANULAR ACCESS CONTROL VIA RBAC MODEL

The new online account security model we are envisioning here would make the granular access control for online accounts feasible. Hence, due to using a separate protocol, it would be possible to distinguish third-party accesses from the account's owner access. Utilizing a granular access control with the role-based scheme of the RBAC model enables fine-grained access to form sandbox around third-party companies' login. When a third-party uses the new protocol to access the users' accounts, the source login by definition becomes distinguishable (e.g., based on which protocol used), and account sandbox becomes feasible. That is, financial institutions can detect the alternative form of account access, when it is initiated using the new protocol versus current protocol, and then allow the access according to the initiator's role. We define the architecture of the access control structure with the user role and access pair as access sequence Role (Object, Attribute), and discuss that next.

7.4.1. ROLES

We define the following three roles to address the security of the accounts and enable each role to perform their authorized access:

- ROLE_ADMIN - This role performs account administration tasks. It offers full privileges that can allow an account manager at a financial institution to make changes to users' accounts.
- ROLE_OWNER - This role gives privileges to the owner of the account, which includes reading the contents of the account and performing transactions on the account.
- ROLE_THIRDPARTY - The third-party companies that are authorized by users to read the contents of their accounts use this role. No transactions can be performed on the account via this role.

7.4.2. OBJECT

The Object is the entity that needs to have granular access protection, which for instance can apply to the whole page, individual accounts, or any components within a page. For example, objects can be a Checking Account, a Savings Account, or a Certificate of Deposit.

7.4.3. ATTRIBUTE

The Attribute is a word length, which consists of fine-grained privileges allowed for a particular object. We define four Attribute flags in a hexadecimal nibble format and present it as binary numbers as shown in Table 1:

Table 1. List of Attribute flags and their values.

Attribute flag	Value (binary)
ATTRB_READ	1000
ATTRB_MODIFY	0100
ATTRB_TRANSACT	0010
ATTRB_CUSTOM	0001

The binary numbers can be used with the logical OR operator to build a hexadecimal nibble representing an access configuration for a particular role. There is a dedicated hexadecimal nibble for each role to specify the privileges for that role. Note that, for the majority of objects, the first three privileges should cover most of the permission cases. However, a "custom" attribute has been added in case there are particular circumstances where an extra permission is needed.

7.4.4. ROLE MASK

The least significant hexadecimal nibble of an Attribute word is assigned to ROLE_ADMIN, then moving toward to the most significant bit. The next nibble is assigned to ROLE_OWNER, and the next is assigned to ROLE_THIRDPARTY. We define role mask for the Attribute word in hexadecimal number as shown in Table 2:

Table 2. List of Role masks and their values.

Role mask	Value (hexadecimal)
ROLE_ADMIN_MASK	F
ROLE_OWNER_MASK	F0
ROLE_THIRDPARTY_MASK	F00

When new roles are needed, it can be directly added to the Attribute word after the last role moving toward the most significant digit. For example, access pair (Checking Account, 0x8EC) has the following meaning:

- The third-party can read, but cannot perform transactions or modify the account.
- The owner of the account can read, modify, and transact on the account.
- An administrator (e.g., manager) can read and modify, but cannot perform transactions on the account.

8. SHORTCOMINGS

In this section, we discuss security and performance shortcomings of a novel protocol and new techniques that we have introduced in this paper.

8.1. PERFORMANCE SHORTCOMINGS

The new HTTPAS protocol was designed to be more secure when used for computer-to-computer communications than what is being employed in current practices. However, this new security enhancement comes at the cost of performance. The loss of performance is due to the primary security steps of verifying client's certificate and public key infrastructure using asymmetric cryptography. The asymmetric encryption and decryption utilize larger bits to avoid brute-force or other forms of attacks. As a result, the runtime may be slower than a simpler method that utilizes lower security for its cryptography.

8.2. SECURITY SHORTCOMINGS

We now discuss security considerations and vulnerability that can affect the protection and integrity of secure communications over the Internet and networks.

8.2.1. CYBER-ATTACKS

In today's modern connected world, with many types of devices connecting us all together, security has never been more important. During our research and writing of this paper, cyber-attacks often made news in the media and on the web. For instance, there were two major cyber-attacks involving two big companies:

- It was announced that there was cyber-attack on Yahoo email in which data from one billion accounts were stolen [11] and Yahoo data being sold on the "dark web" [12].
- There was a cyber-attack on Dyn, a major provider of domain name service. The attackers used Distributed-Denial-Of-Service (DDOS) causing interruption of services to the popular websites [13].

The attackers often use any vulnerabilities they can exploit to achieve their objectives. In the case of Dyn, the attackers were exploiting the IoT (Internet of Things) vulnerabilities. The IoT devices such as webcams and digital video recorders (DVRs) are by nature less secure than mainstream devices. In the new era of computing, more and more IoT devices are becoming accessible online via the Internet. Therefore, creating an opportunity for attackers to utilize them to create and operate bots (i.e., applications that perform automated instructions), to overwhelm specific system or network of their choosing. [14] The concept of bots is a known technique, which was used in the past on systems that were compromised by malware to take down victim's website using heavy traffic.

The number and the types of devices (Figure 6) being accessible on the Internet is on the rise [15]. Therefore, creating even a bigger number of systems that attackers can exploit and take advantage of if those devices have security flaws or weaknesses. Manufacturers need to create more secure IoT devices with automatic updates built-in, which can update the device's firmware when security vulnerabilities are found. The login security for IoT devices should have double layer security to create another barrier for the attackers to overcome. If attackers manage to

compromise the first layer of security, they would still have one more security layer to overcome before gaining access to the device.

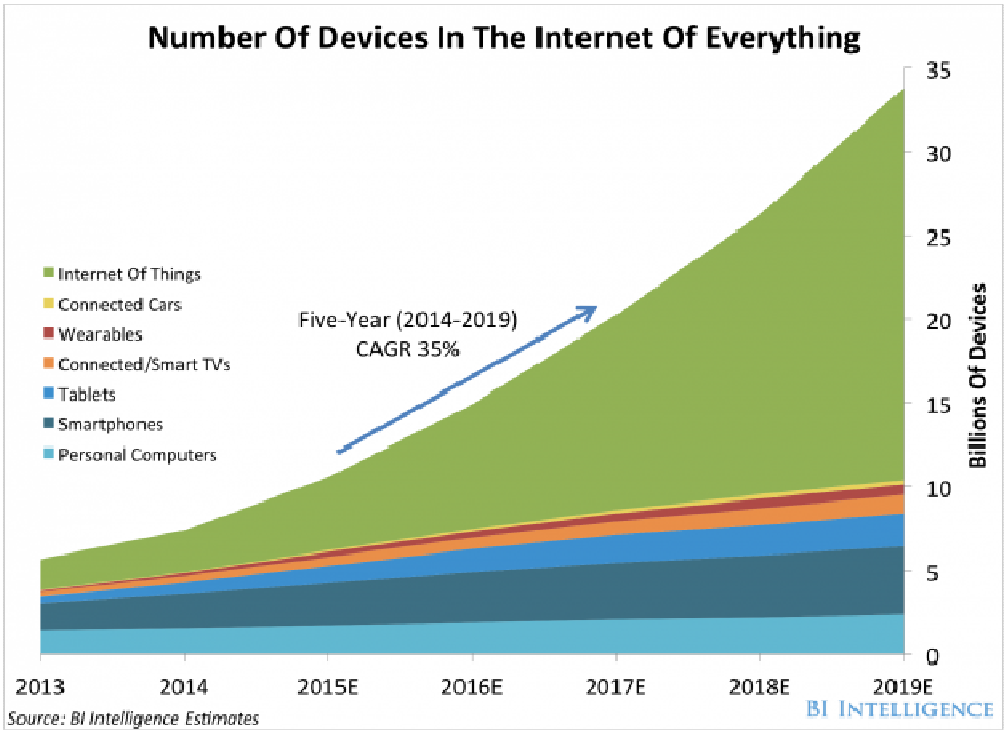


Figure 6. Number of devices on the Internet are on the rise

8.2.2. SSL PROTOCOL VULNERABILITY

Although SSL design and architecture has built-in mechanisms for preventing attacks (e.g., Eavesdropping and Man-In-The-Middle attack prevention features), it is not immune to attackers [16]. The vulnerability may come from flaws in SSL protocol, SSL libraries, hacked digital certificates, and cipher negotiations, to name a few. These flaws are often discussed at security conferences such as Black Hat. Our protocol may be vulnerable to attackers if the infrastructures that we utilize as a foundation for our protocol becomes vulnerable.

8.2.3. CA ENTITIES VULNERABILITY

The CAs are the basis for the trust models of verifying digital certifications on the Internet today. If a vulnerability exists in a CA infrastructure and attackers can exploit it to their advantage, it breaks the trust model that we rely on for identification. The CA security breach may have a profound effect on the reputation of the CA, which may result in the ceasing of the operation of the CA. Nevertheless, if the CA or certificates are hacked, it makes our protocol vulnerable to an attacker. An example of attack is DigiNotar, which was hacked by an attacker in 2011. The attack on DigiNotar had a catastrophic effect, and as a result, they were not able to recover from the attack, and filed for bankruptcy [17], [18].

8.2.4. CRYPTOGRAPHY VULNERABILITY

Mathematicians have proven that encryption algorithms are mathematically sound; however, as computer systems are getting more powerful over time, the number of bits used in calculations of the encryption mechanism must also increase. For example, RFC7935 states that the RSA key pairs for use in public key infrastructure (which is a framework for a one-way function) must be using at least 2048-bit [19]. Therefore, the strength of encryption algorithms for a one-way function relies on the computing power and time required to decrypt the message.

8.3. INITIAL BURDEN FOR ALL PARTIES INVOLVED

All the stakeholders (e.g., financial institutions, third-party companies, end-users) would need to make changes to their current processes, practices, and habits. For instance, financial institutions would need to modify their online accounts and systems to support the new protocol and account access. Third-party companies must make changes to their current informational retrieval processes, as well as follow the specification of the new protocol for user authentication. Account owners would need to request the new access credentials from their financial institutions, and provide the third-party vendor information they want to use for their account. Also, users need to change their full access login credentials if they have already given that to any third-party companies.

9. RELATED WORK

In this section, we discuss related work in the area of authenticating users' access to online services.

9.1. OAUTH

A related work to our research is the OAuth authorization framework. OAuth allows authentication of a user without exposing their account credentials to a third-party provider over HTTP protocol [20]. The concept behind OAuth is that, when a user is trying to login to a third-party website, they can use their accounts' username/password from one of the major companies (e.g., Facebook, Twitter, LinkedIn) to authenticate themselves with a third-party company. With this method, users no longer have to reveal their login credentials to the third-party companies. Currently, the deployments of OAuth framework have had limited exposure. This model has not been widely accepted as the alternative to hiding and protecting account credentials from third-party companies. As a result, OAuth currently suffers from penetration and adoption challenges, as well as privacy and security concerns.

9.2. APPLICATION PROGRAMMING INTERFACE

Authentication via Application Programming Interface (API) is an example of related work, at which companies provide mechanisms for information retrieval and manipulation from their systems. This type of access is performed via application permission model [21] given by their API (e.g., Twitter API) [22]. This method is used to read, write, perform information retrieval and data mining to access and harvest data from companies (e.g., Twitter, Google) which support the APIs. The API may use OAuth technology as its underlying authorization mechanism to authenticate the request. Due to the nature and exposure of information provided with this model, it has its limit where sensitive information, security, and privacy are concerned.

10. CONCLUSIONS

The ubiquity of the Internet has increased the potential exploitation of security weaknesses and vulnerabilities of our financial institutions. Users are sharing their full access accounts' credentials with third-party companies and need to be wary of sharing this sensitive information with others. Consequently, others can use those credentials to execute transactions on their account. Meanwhile, users' need of financial aggregation services provided by the third-party companies are growing. The increase in using these services, coupled with the lack of an alternate mechanism for account authentication with limited access, can make users vulnerable to attackers.

In this research paper, we have introduced and prescribed a novel and holistic model with a new protocol for online account architecture to be used by third-party companies. This model works on the notion of two security components: 1) Authentication mechanism utilizing new handshake protocol, which enables verification of users' credentials that is more secure than a username/password combination, 2) User access sandboxing technique via role-based granular access control that protects the account against unwanted transactions. Utilizing the new architecture, design, and novel techniques we have presented in this paper, users no longer need to give out their full access account credentials to third-party companies. Instead, users can provide a limited and alternate access account credentials for third-party use. Consequently, when attackers compromise the third-party's computer and steal users' access information, they cannot perform transactions on the account. In the case of a security breach, users can revoke the third-party login credentials with no impact on the existing full access account credentials. Furthermore, the novel and holistic new techniques we have prescribed are universal and can be adapted for other domains (e.g., medical records, airline ticket system, online stores, and emails) with little or no modifications to the architecture we have presented in this paper.

REFERENCES

[1] I. Jinwoo Hwang, "The Secure Sockets Layer and Transport Layer Security," Jun. 2012. [Online]. Available: http://www.ibm.com/developerworks/library/ws-ssl-security/index.html

[2] "Service Name and Transport Protocol Port Number Registry," 00017. [Online]. Available: http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[3] E. Rescorla, "HTTP Over TLS," Internet Requests for Comments, RFC Editor, RFC 2818, May 2000. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2818.txt

[4] "ITU-T The Directory: Public-key and attribute certificate frameworks." [Online]. Available: http://www.itu.int/itu-t/recommendations/rec.aspx?rec=13031

[5] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," Internet Requests for Comments, RFC Editor, RFC 3647, November 2003.

[6] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) profile," Internet Requests for Comments, RFC Editor, RFC 5280, May 2008, http://www.rfc-editor.org/rfc/rfc5280.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5280.txt

[7] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating system concepts, 9th*. Addison-Wesley Reading, 2013.

[8] G. S. Graham and P. J. Denning, "Protection: Principles and practice," in *Proceedings of the May 16-18, 1972, Spring Joint Computer Conference*, ser. AFIPS '72 (Spring). New York, NY, USA: ACM, 1972, pp. 417–429. [Online]. Available: http://doi.acm.org/10.1145/1478873.1478928

[9] M. G. Solomon and M. Chapple, *Information security illuminated*. Jones & Bartlett Publishers, 2009.

[10] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-based Access Control," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 224–274, Aug. 2001. [Online]. Available: http://doi.acm.org/10.1145/501978.501980

[11] S. Fiegerman, "Yahoo says data stolen from 1 billion accounts," Dec. 2016. [Online]. Available: http://money.cnn.com/2016/12/14/technology/yahoo-breach-billion-users/index.html

[12] S. Larson, "Hackers are selling Yahoo data on the dark web," Dec. 2016. [Online]. Available: http://money.cnn.com/2016/12/16/technology/yahoo-for-sale-data-dark-web/index.html

[13] "Hacked home devices caused massive Internet outage." [Online]. Available: http://www.usatoday.com/story/tech/2016/10/21/cyber-attack-takes-down-east-coast-netflix-spotify-twitter/92507806/

[14] M. Pittenger, "Dyn DDoS Attack: IoT Vulnerabilities," Oct. 2016. [Online]. Available: http: //blog.blackducksoftware.com/ddos-attack-dyn-iot-vulnerabilities

[15] J. Greenough "The Internet of Everything," 2015. [Online]. Available: http://www.businessinsider.com/internet-of-everything-2015-bi-2014-12

[16] "Logjam: the latest TLS vulnerability explained," May 2015. [Online]. Available: http://blog.cloudflare.com/logjam-the-latest-tls-vulnerability-explained/

[17] J. Prins and B. U. Cybercrime, "DigiNotar certificate authority breach Operation Black Tulip," 2011.

[18] K. Zetter, "DigiNotar files for bankruptcy in wake of devastating hack," *Wired magazine, September* 2011.

[19] G. Huston and G. Michaelson, "The profile for algorithms and key sizes for use in the resource public key infrastructure," Internet Requests for Comments, RFC Editor, RFC 7935, August 2016.

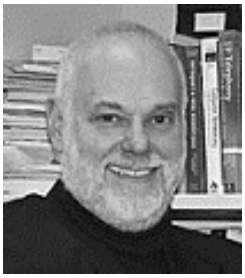
[20] "OAuth Community Site." [Online]. Available: https://oauth.net/

[21] "Application Permission Model - Twitter Developers." [Online]. Available: https://dev.twitter.com/oauth/overview/application-permission-model

[22] "Twitter Developer Documentation - Twitter Developers." [Online]. Available: https://dev.twitter.com/docs

AUTHORS

Mehrdad M. Nourai received a B.S. degree in Electrical Engineering from Northeastern University, Boston Massachusetts in 1982, and an M.S. degree in Computer Science from Boston University, Boston Massachusetts in 2000. He is currently a Ph.D. degree candidate in Computer Science at the University of Massachusetts Lowell, Lowell Massachusetts. He has over three decades of professional experience in computer industry and academics. His industry experience consists of architect, design, and development of software for all kinds of computer systems including embedded systems and systems with standard, proprietary, and real-time operating systems. He has been teaching computer science courses as an adjunct faculty for the MET Computer Science Department at Boston University since 2000. Also, he has been teaching courses as an adjunct faculty for the Computer Science Department at Salem State University since 2008. His research and areas of interests include Computer Networks and Security, Data Communications, Human-Computer-Interaction, Database, and Mobile Apps Development.



Haim Levkowitz is the Chair of the Computer Science Department at the University of Massachusetts Lowell, in Lowell, MA, USA, where he has been a Faculty member since 1989. He was a twice recipient of a US Fulbright Scholar Award to Brazil (August – December 2012 and August 2004 – January 2005). He was a Visiting Professor at ICMC — Instituto de Ciencias Matematicas e de Computacao (The Institute of Mathematics and Computer Sciences)—at the University of Sao Paul, Sao Carlos – SP, Brazil (August 2004 - August 2005; August 2012 to August 2013). He co-founded and was Co-Director of the Institute for Visualization and Perception Research (through 2012), and is now Director of the Human-Information Interaction Research Group. He is a world-renowned authority on visualization, perception, color, and their application in data mining and information retrieval. He is the author of “Color Theory and Modeling for Computer Graphics, Visualization, and Multimedia Applications” (Springer 1997) and co-editor of “Perceptual Issues in Visualization” (Springer 1995), as well as many papers on these subjects. He is also co-author/co-editor of "Writing Scientific Papers in English Successfully: Your Complete Roadmap," (E. Schuster, H. Levkowitz, and O.N. Oliveira Jr., eds., Paperback: ISBN: 978-8588533974; Kindle: ISBN: 8588533979, available now on Amazon.com: http://www.amazon.com/Writing- Scientific-Papers-English- Successfully/dp/8588533979). He has more than 44 years of experience teaching and lecturing and has taught many tutorials and short courses, in addition to regular academic courses. In addition to his academic career, Professor Levkowitz has had an active entrepreneurial career as Founder or Co-Founder, Chief Technology Officer, Scientific and Strategic Advisor, Director, and venture investor at a number of high-tech startups.