Hindawi
Security and Communication Networks
Volume 2017, Article ID 4286903, 20 pages
https://doi.org/10.1155/2017/4286903

Research Article

CLAS: A Novel Communications Latency Based Authentication Scheme

Zuochao Dou,[1] Issa Khalil,[2] and Abdallah Khreishah[1]

[1] *Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA*
[2] *Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar*

Correspondence should be addressed to Zuochao Dou; zd36@njit.edu

Received 2 January 2017; Revised 3 April 2017; Accepted 7 May 2017; Published 12 June 2017

Academic Editor: Emanuele Maiorana

We design and implement a novel communications latency based authentication scheme, dubbed CLAS, that strengthens the security of state-of-the-art web authentication approaches by leveraging the round trip network communications latency (RTL) between clients and authenticators. In addition to the traditional credentials, CLAS profiles RTL values of clients and uses them to defend against password compromise. The key challenges are (i) to prevent RTL manipulation, (ii) to alleviate network instabilities, and (iii) to address mobile clients. CLAS addresses the first challenge by introducing a novel network architecture, which makes it extremely difficult for attackers to simulate legitimate RTL values. The second challenge is addressed by outlier removal and multiple temporal profiling, while the last challenge is addressed by augmenting CLAS with out-of-band-channels or other authentication schemes. CLAS restricts login to profiled locations while demanding additional information for nonprofiled ones, which highly reduces the attack surface even when the legitimate credentials are compromised. Additionally, unlike many state-of-the-art authentication mechanisms, CLAS is resilient to phishing, pharming, man-in-the-middle, and social engineering attacks. Furthermore, CLAS is transparent to users and incurs negligible overhead. The experimental results show that CLAS can achieve very low false positive and false negative rates.

1. Introduction

During April 2015 breaking news from RSA, Idan Tendler pointed out that 80% of successful attacks exploit authentication credentials [1]. Passwords have always been the most notorious but dominant authentication credential for web services. However, attackers continuously innovate ways to compromise passwords: phishing, pharming, guessing, shoulder surfing, brute force, social engineering, and eavesdropping, just to name a few [2].

To overcome password evolving weaknesses, multifactor authentication schemes have been implemented as one of the most appealing solutions [3, 4]. Multifactor authentication schemes require, in addition to the regular password, other authentication codes to log in. The additional credentials (passwords, biometrics, PINs, hardware tokens, etc.) are either preagreed upon or, in most cases, delivered to users in real-time through a different channel such as email and SMS. Multifactor authentication schemes considerably enhance the security of password-based systems; however, they suffer from many limitations and they introduce new vulnerabilities.

(i) Many multifactor authentication mechanisms do not protect against man-in-the-middle attacks (through phishing or pharming, for example, [5, 6]). A phishing attacker may get the additional authentication codes the same way as he/she gets the traditional credentials. The attacker can simply present the user with a forged fake screen to input his/her additional codes, which the attacker uses in real-time to impersonate the legitimate user.

(ii) Many of the channels used to provide additional credentials are likely easy to be compromised in many cases. One of the most commonly used second-factor channels is smartphone SMS. However, smartphones are vulnerable to theft and loss and hence may expose authentication codes passed through them. According to customer report, 3.1

Table 1: A survey that compares single-factor and two-factor authentication of an e-banking service.

Ranked best	Overall preference	Convenience	Security
Single-factor	32 (52.5%)	42 (68.9%)	4 (6.6%)
Two-factor	26 (42.6%)	9 (14.8%)	46 (75.4%)
Rate equally	3 (4.9%)	10 (16.4%)	11 (18.0%)

million smartphones were stolen and 1.6 million smartphones were lost during 2013 in the USA alone [7]. Most importantly, smartphones become more and more susceptible to mobile malware and spyware. In the 2015 threat report [8], Symantec revealed an active Android malware that can intercept SMS messages with second-factor authentication codes and forward them to attackers.

(iii) Many multifactor authentication mechanisms have poor user experience due to scanning or typing of extra bits of information. The survey presented in Table 1 [14] reveals that only 14.8% of the users of a two-factor authentication e-banking service think it is convenient. This clearly indicates that any security solution has to accommodate the trade-off between security guarantees and usability desires.

A key question that motivated this work is if it is possible for an authenticator to distinguish whether an ongoing login attempt is initiated by the legitimate user or by a perpetrator in possession of the legitimate credentials. Profiling the behavior of the user is one of the first solutions that jump straight ahead as it is proved to be effective in flagging some of the most dangerous attacks. A plethora of mechanisms have been introduced to profile users' behavior in the context of anomaly based intrusion detection systems. Many of such mechanisms profile users' network traffic information, such as entropy of IP addresses and port numbers, percentage of network bandwidth utilization, packet loss rate, and percentage of unencrypted traffic flow. However, it has been shown that many of the features used are not robust [15]; that is, attackers could easily change some of the features that characterize their behavior to match those of the victims and, hence, successfully impersonate them. Moreover, these network-based profiling mechanisms incur prohibitively high overhead because they require a relatively large amount of real-time traffic to achieve acceptable profiling accuracy [16, 17]. This may also lead to poor user experience due to the longer time it takes to log in.

In this work, we *design* and *implement* a novel scheme, dubbed CLAS, that strengthens the security of web authentication by profiling the round trip network communications latency (RTL) between clients and authenticators. CLAS complements the state-of-the-art authentication schemes by overcoming many of their limitations such as susceptibility to phishing and pharming. In fact, CLAS is capable of protecting legitimate users even when their credentials are compromised. More importantly, CLAS has lightweight real-time overhead and utilizes features that are *extremely hard to be manipulated*. Additionally, CLAS is *transparent to end users* as they do not provide any additional authentication information beyond that of username and password.

CLAS uses round trip network communications latency (RTL) to uniquely profile users. In this context, RTL is defined as the time elapsed between sending out a packet and receiving its acknowledgment. It has been observed that RTL between two communicating parties (e.g., client and server) approximately follows a Gaussian distribution regardless of the Internet access technology used (e.g., WiFi, wireline, and 3G/4G) [18]. The RTL changes when any of the communicating parties changes the location of its Internet access.

When a user registers for a service, the responsible server initiates a process to establish his/her profile. For every user in CLAS, the profile is the mean and standard deviation of the round trip network communications latency (RTL) between the user and the server through a special network device (i.e., *Stealthy Relay* (SR) as discussed in Section 3.3) that resides in the middle of them. Profile parameters are then stored on the server along with the traditional credentials (username and password) for future login verification. Later login attempts are profiled in real-time and compared against the stored profile parameters. Access is granted only when the ongoing real-time profile parameters fall within predetermined boundaries of the stored profile parameters.

In the context of this work, the same geographical location does not necessarily imply the same login location. Two users using different types of Internet access techniques (e.g., one uses WiFi while the other uses 4G) most likely have different login locations even if they have the exact same geographical location. Throughout the rest of this paper, location refers to the login location, not the geographical location.

The *first research challenge* that CLAS faces is the secure measurement of RTL. CLAS has been carefully designed to make it highly unlikely for attackers, and even for the legitimate users themselves, to learn profile parameters. This is mainly achieved, as detailed in Section 3.3, by inserting a special one-way forwarding device, dubbed *Stealthy Relay* (SR), in the round trip route between the clients and the server. This novel design makes RTL values *extremely hard to be manipulated*. It is true that the RTL is different from RTT (round trip time) by our design, mainly to prevent potential manipulation. RTT can be easily estimated and hence manipulated by any perpetrator to impersonate legitimate users, simply by judiciously delaying packets. In CLAS, the profile is the round trip network communications latency (RTL) between the user and the authenticator through the SR as discussed in Section 3.3 and illustrated in Figure 2. It is directly measured, stored, and used by the authenticator. As long as the reference profile and the real-time profile are measured through the same SR, there is no need to do any adjustments or modifications to user profiles.

The *second research challenge* for CLAS is to cope with potential network instabilities. The work in [18] and our experiments show that network instabilities (e.g., network congestion due to server workload increment [19], intermediate node dynamic wake-up time scheduling, and/or reprogramming [20, 21]) may cause network communications latency to vary over time. CLAS addresses this issue through preprocessing of RTL measurements and using multiple temporal profiling. The detailed discussion and analysis of the impact of network instabilities on CLAS are presented in Section 3.7. The third challenge is the ability of CLAS to support mobile clients. CLAS is designed to only allow login from profiled locations, which creates an inherent challenge for applications that support anywhere authentication. CLAS can support anywhere authentication by incorporating out-of-band-channels such as SMS messages or by integrating additional profiling features such as browser fingerprints and typing patterns as explained in Section 5.

The *final research challenge* that CLAS faces is the potential successful authentication of perpetrators who both possess login credentials and have access to the corresponding profiled location. CLAS uses two factors to authenticate, password and RTL, and it can tolerate the compromise of either one but not both; that is, CLAS can defend against attackers who capture legitimate passwords but have no access to the corresponding profiling location. This limits the attack surface of a compromised account from anywhere in the world to only the profiled location, which is a significant security improvement especially for nontargeted attacks. Most of the password compromise attacks are nontargeted as in the case of phishing through massive email spam, malware spreading through worldwide bonnets, and server database compromise.

In this work, we address the first two challenges in detail and outline the proposed mitigation of the last two challenges. The detailed design and analysis of the last two challenges are planned for a future work.

We provide mathematical analysis and conduct extensive experiments to evaluate the security guarantees and performance overhead of CLAS. We mainly use two security metrics, false positive rate (FP) and false negative rate (FN). False positive occurs when the attacker succeeds to log in using compromised credentials, while false negative occurs when the legitimate user is denied access from his/her profiled location. The results show that CLAS can achieve FP as low as 0.0017 while the FN is below 0.007. In other words, out of 1000 legitimate login trials less than 7 fails on average. Additionally, perpetrators who possess the credentials of a legitimate user have only 1.7 in 1000 chance to authenticate on average. We also evaluate the login latency overhead, the bandwidth overhead, and the storage requirements of CLAS and show that they are negligible.

Our contributions in this work are summarized as follows:

(i) To design and implement a novel scheme, dubbed CLAS, that strengthens the security of web authentication by leveraging the round trip network communications latency (RTL) between clients and authenticators

(ii) To design and implement the novel network architecture of CLAS which ensures its resiliency to manipulation attacks

(iii) To design and develop algorithms to mitigate the impact of network instabilities on CLAS

(iv) To outline the proposed solutions to support mobile clients and defend against the compromise of both password and RTL (e.g., access to the profile location)

(v) To perform security analysis about defense methodologies of CLAS for many types of attacks that cannot be addressed by many state-of-the-art authentication schemes

(vi) To perform mathematical analysis to evaluate the security properties and performance overhead of CLAS

Build a prototype of CLAS and conduct extensive experiments to practically evaluate its security guarantees and performance overhead.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 defines the attack model and presents the design and the implementation details of CLAS, including authentication process, secure measurement of RTL, and network instabilities. Section 4 discusses the defense methodologies of CLAS for many types of attacks that cannot be addressed by many state-of-the-art authentication schemes. Section 5 discusses the challenges of CLAS and presents extensions that address them, including mobile user authentication, prevention of same location attacks, and RTL sample space analysis. In Section 6, we present the experimental setup and results. In Section 7, we conclude this study and discuss potential future research extensions. Appendix A presents the theoretical analysis of the tradeoff between FP and FN based on the Gaussian approximation.

2. Related Work

In [18], the authors perform extensive experiments to measure network communications latency and conclude that it approximately follows a Gaussian distribution with the mean and standard deviation used to characterize different cloud servers. We partially leverage this observation to develop our authentication scheme.

In [22], the authors propose a packet delay-based scheme to detect man-in-the-middle (MitM) attacks. They assume that delay increases in the existence of MitM attacker. However, this scheme suffers from serious security issues because the packet delay as presented can be easily manipulated. The way packet delay is computed enables attackers to reduce the delay by simply using a proxy server. More importantly, the scheme uses timestamps of TCP packet headers to calculate the delay, which could be easily manipulated. Additionally, the scheme fails to correctly address mobility issues. CLAS, on the other hand, has a novel network architecture carefully designed to mask network delay from everyone except the profiling server. This design makes CLAS *highly resilient to any possible latency manipulation* and hence highly

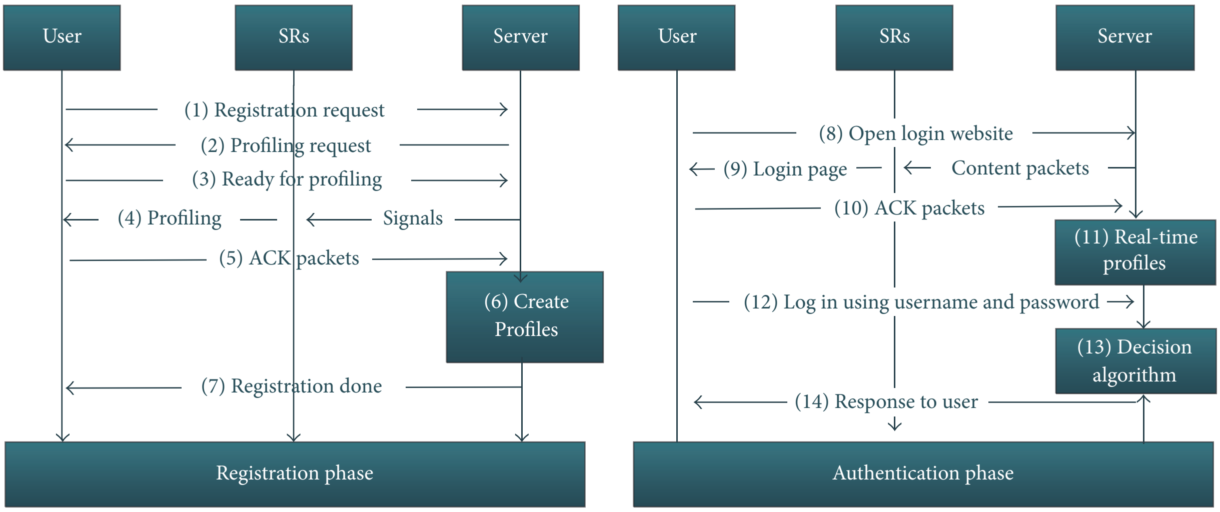


Figure 1: CLAS authentication flowchart.

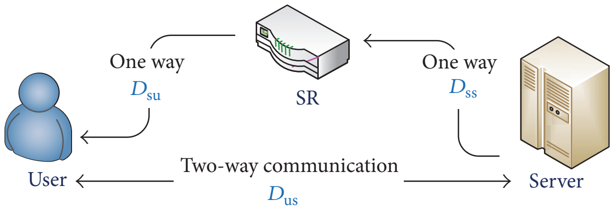


Figure 2: CLAS architecture.

secure. Moreover, CLAS extensions support authentication on mobile scenarios.

Since 2010, Gmail launched a service that enables its users to detect suspicious account login activities based on IP information [23]. The detection is made via the Gmail automated system by matching the relevant IP address to a broad geographical location. Gmail users can see all their login activities within specific locations. This feature enables Gmail users to detect any suspicious login activities and facilitates responses by, for example, changing login password. However, IP-based verification is too weak due to the easiness by which IP-based information can be manipulated. Two easy ways to fake location information are proxy servers (e.g., Tor Project [24]) and VPNs. Alternatively, an attacker can choose to perform IP hijacking attack to impersonate legitimate users. One-way to hijack IPs is by using BGP hijacking attacks [25]. BGP/IP hijacking is much more common than current researchers think and it is hard to be detected in the form of local BGP hijacking [26, 27]. Furthermore, IP-based authentication suffers many other limitations due to current Internet infrastructure including (1) extensive use of NAT, especially the use of Carrier Grade NAT (CGN) or Large Scale NAT (LSN) [28], and (2) complex and inconsistent IP address configuration policies by different ISPs. Therefore, IP address based authentication is more suitable for LAN other than general web service authentication.

Comparing to our previous work in [29], the new paper has achieved significant improvements, such as (1) presenting a theoretical analysis of the FP and FN based on the Gaussian approximation; (2) conducting 3 sets of new experimental evaluations; (3) laying out more detailed research challenge analysis and corresponding solutions/extensions; (4) performing a security analysis about defense methodologies of CLAS for many types of attacks that cannot be addressed by many state-of-the-art authentication schemes; and (5) providing a prototype website.

3. Communications Latency Based Authentication Scheme (CLAS)

In this section, we present the attack model, the design, and implementation details of CLAS. As depicted in Figure 1, CLAS ecosystem comprises three entities, namely, User, Server, and *Stealthy Relay* (SR). Users log in to servers to access services. Servers authenticate users based on their traditional credentials and previously stored measurements of network communications latency. Servers build reference profiles for users once they join the offered services. Servers, later, grant/deny access after comparing real-time profiles built during each login attempt with stored reference profiles built at the registration time. Servers include SR in the round trip route to users to thwart possible attempts to learn profile parameters.

3.1. Attack Model. We assume that the registration phase when reference profiles are established is secure. This is a reasonable assumption because secure registration (or secure profiling phase) is considered the root of trust of all cybersecurity systems that utilize profiling techniques (e.g., profiling the user's behaviors for authentication [30, 31]). We also assume that the *Stealthy Relay* (to be introduced later) is secure and is connected to a network segment that cannot be accessed by attackers. That is, attackers cannot connect from the network segment of the *Stealthy Relay* (SR) in a bid to learn profile parameters of users. Remote access compromise and denial of service attacks are out of the scope of this work. In remote compromise attacks, attackers may manage to remotely access the computer of the user, through, for example, compromise of remote desktop credentials or infection by a Trojan Horse, while in denial of service attacks, attackers try to prevent login of legitimate users.

3.2. CLAS Authentication Process. CLAS authentication consists of registration phase and authentication phase (Figure 1). Registration is used to build profiles while authentication verifies future login attempts. To demonstrate the authentication process of CLAS, we use the typical web-page login scenario depicted in Figure 1.

Registration starts when a user creates a new account with the intended server (Step (1) in Figure 1). The registration daemon at the server starts the profiling process through which multiple packets are exchanged with the user to establish its reference RTL profile. The server asks for the user's permission to engage in the profiling process (Step (2) in Figure 1). After the user accepts the request (Step (3) in Figure 1), the server sends a sequence of packets, called *Profiling Signals*, to the SR and the SR forwards them to the user (Step (4) in Figure 1). Then the user acknowledges the reception of each profiling signal by sending direct acknowledgments back to the server (Step (5) in Figure 1). A profiling signal is a very small packet (similar to ping request and reply messages) intended to measure the RTL between the server and the user.

Detailed analysis of the number of profiling signals required to establish sufficiently accurate profiles is provided in the analysis section (Section 3.5). The RTL for each profiling signal (Steps (4) and (5) in Figure 1) is computed at the server as the time elapsed between sending out the signal and receiving its acknowledgment back. The average and the standard deviation of the RTL values of all the profiling comprise the profile of the user.

In the authentication phase, downstream (Steps (8) and (9) in Figure 1) login packets from the server are acknowledged by the user. Using the profiling signals and acknowledgments, the server computes the mean and the standard deviation of the RTL in real-time (Step (11) in Figure 1). The server, then, compares the real-time measured profile parameters with those of the reference profile. Based on the result of the comparison, access is either granted or denied.

3.3. Secure Measurement of RTL. An important security concern in the naive measurement of RTL is that it allows attackers to easily figure it out. For example, an attacker can simply ping the server from the location of the user. The round trip delay of the ping packets provides excellent estimation of the round trip delay between the user and the server and, hence, can be used to compute profile parameters. If the attacker learns the profile parameters of a user and if he/she is in a location with communications latency less than that of the user, he/she can easily mimic the profile of the user. The attacker simply adds appropriate delay before acknowledging the profiling signals. To address this important security concern, CLAS introduces a special network component called *Stealthy Relay* (SR) in the round trip route between the user and the server (Figure 2). SR is a secure one-way packet forwarding device owned and controlled by the service provider. SR has one and only one function: it is dedicated to receive the profiling signals from the server and relays them to the user; that is, the SR serves as a special purpose router. The profiling signal received by the user has the IP of the server as the source address, while the IP address of the SR is only known to the server.

SR is configured to refuse any other kind of communication with any entity, including the server itself. For example, the SR does not respond to any ping requests. Additionally, the SR is installed on a separate dedicated secure network segment; that is, attackers cannot connect to the location of the SR nor can they compromise it. Therefore, attackers cannot use SR or the network segment where SR is connected to send ping packets (or any other packets) to any entity. The detailed configuration of SR is presented in Section 6.1.

The purpose of SR is to make it extremely hard for any entity, *including the users themselves*, to learn the RTL. SR achieves this by creating new path-segments in the round trip path between the server and its users. The communications delay over these new path-segments cannot be measured by outsiders. In Figure 2, the RTL is the sum of the delay over the path-segment from the server to the SR (D_{ss}), the delay over the path-segment from the SR to the user (D_{su}), and the delay over the path-segment from the user to the server (D_{us}). The attacker may be able to estimate D_{us} by pinging the server from the location of the user; however, it is not possible for him/her to figure out D_{su} and D_{ss} due to the one-way communications architecture of the SR. Recall that SR does not respond to any communication *even from the server itself*. In other words, round trip cycles cannot be established on the path-segment between the user and the SR or between the SR and the server. In this architecture, the only cycle that can be established is the one initiated by the server using the profiling signals and is completed by receiving the corresponding acknowledgments. Therefore, the server is the only entity of CLAS that knows the profile parameters. *No other entity, including the user herself, can compute or learn even his/her own profile parameters.*

There are two possible ways for attackers to estimate the round trip profile parameters: (i) establish round trip cycles on D_{su} and D_{ss} path-segments by pinging the user and the server from the SR or its location. However, we have shown earlier that this is not possible because the SR is assumed to be secure and connected to a location that cannot be accessed by attackers. Furthermore, recall that the network location of the SR cannot be determined because the IP address of the SR is only known to the server. Therefore, attackers cannot use ping packets from SR or its location to estimate D_{su} and D_{ss}. (ii) An insider who has access to the location of the server may monitor the profiling signals and the acknowledgments to estimate profile parameters. This is thwarted by encrypting the profiling signals at the server before sending them to the SR; that is, the insider cannot map the incoming acknowledgments and the outgoing profiling signals. More details are presented in Section 3.4.

Someone may argue that another possible way to estimate D_{su} and D_{ss} is by compromising the server itself. However, this scenario does not apply to our discussion here, simply because it would be irrational for an attacker, in this case, to compromise users in a bid to access their services; he/she has already compromised the service itself.

The service provider may set up more than one SR at different locations for the purpose of load balancing and recovery from loss. Multiple SRs could be deployed using existing infrastructures (e.g., Google has many offices, repair branches, and data centers) such that the deployment cost can be substantially reduced. The users are randomly assigned to different SRs. Additionally, some service providers may optionally change the assignment of the SR to change the profile parameters of the users as an extra security precaution against possible leakage of users' profile parameters. Note that SR reassignment is quite a simple configuration that is performed by the server and is completely transparent to the user. After the user logs in using his/her current profile, a new profile is established using the new SR (Section 3.2) by sending a sequence of profiling signals through the new SR. The new profile replaces the old one and the new SR is used for future logins.

3.4. Profiling: Protocol and Implementation. In this section, we show how SR is used to securely measure RTL. The server first encrypts each profiling signal using the symmetric key it shares with the SR. A programming interface implemented on top of the MAC layer at the server records the send time *(sendtime)* of each downstream profiling signal. The SR decrypts the profiling signal and then relays it to the user. Finally, the user sends an acknowledgment for each received profiling signal directly to the server. The profiling signals received at the user have the IP of the server as the source address and, hence, the IP address of the SR is not exposed to the users. This setup also has the advantage that no special agent is needed at the user side, a feature that makes CLAS completely user transparent. When the programming interface at the server receives the acknowledgment from the user, it records its reception time *(rcvtime)* and computes RTL = *rcvtime* − *sendtime*. When enough profiling signal acknowledgments are received, the mean and the standard deviation are computed and used to determine whether to grant or deny access.

3.5. Profiling Sample Size. In this section, we use the Gaussian approximation of RTL to analyze the minimum requirements to build sufficiently representative profiles while controlling bandwidth and login latency overhead. Time and bandwidth overhead incurred during the registration phase may not be of a material concern since it is a one-time process. However, authentication happens with every login and authentication profiles have to be established in real-time. Therefore, it is of paramount importance to ensure that the incurred login latency and bandwidth overhead are within acceptable limits. The main source of both login latency overhead and bandwidth overhead is the extra (beyond the regular login packets) profiling signals that are required to build sufficiently accurate profiles. The accuracy of profiles is determined by the acceptable rate of false positives and false negatives. In this section, we compute the approximate number of profiling signals (i.e., sample size) that produces acceptable profile accuracy.

Assume a population with Gaussian distribution that has standard deviation S and mean M. The goal is to find the minimum sample size, N, that produces a mean, μ, within a certain error margin, δ, with a certain confidence level $1-\alpha$. The error margin (aka the confidence interval), δ, is the maximum allowed distance between M and μ. The confidence level represents how confident we are that the measured mean (μ) falls within the confidence interval.

For Gaussian distributions, it has been shown ([32, 33]) that the minimum sample size N can be calculated as

$$N \geq \left(\frac{Z_{1-\alpha}}{\delta}\right)^2 S^2, \tag{1}$$

where Z is the critical value for the normal distribution. In other words, for a sample size of N, we have $1-\alpha$ confidence that the measured mean (μ) will fall in the range of

$$M-\delta \leq \mu \leq M+\delta. \tag{2}$$

Similarly, the range of the real-time measured standard deviation σ can be computed using Chi-Square (χ) table as

$$\sqrt{\frac{\chi_L^2 \cdot S^2}{N-1}} \leq \sigma \leq \sqrt{\frac{\chi_R^2 \cdot S^2}{N-1}}, \tag{3}$$

where χ_L and χ_R are computed for specific values of α using the Chi-Square table.

For example, to be 99% confident that μ is in the range of $\pm\delta = \pm 0.5 \cdot S$ (i.e., the error tolerance is half of the standard deviation), the number of measurements should be $N \geq (Z_{1-\alpha}/\delta)^2 S^2 = 4 \cdot (Z_{0.99})^2 \approx 27$. In other words, with more than 27 profiling signals, we will be 99% confident that the real-time profile mean will be within $0.5 \cdot S$ from the reference profile mean.

3.6. Access Decision. The decision to grant or deny access for a certain login attempt is simple: if the real-time RTL mean is within the range defined in (2) and the real-time RTL standard deviation is within the range defined in (3), access is granted; otherwise, access is denied.

When out-of-band-channels such as phone SMS or email are available, the access decision accuracy is enhanced by incorporating an additional state: *further information is required*. If the login attempt has (i) a real-time RTL mean outside the range defined in (2) but within the range defined in (2) ± γ_1 and (ii) a real-time RTL standard deviation outside the range defined in (3) but within the range defined in (3) ± γ_2, the decision will be *further information is required*. If further information is required, the user has to provide additional information that he will receive through the out-of-band-channel.

3.7. Network Instabilities. Network instabilities may be caused by different factors including permanent routing changes, congestion, DDoS, and traffic rerouting. Such instabilities may cause network communications latency to vary overtime. To understand network instabilities and to effectively alleviate their impact on CLAS, we classify them into three broad categories.

Instantaneous instabilities are instabilities which lead to transient changes in network communications latency and, hence, it only affects a few profiling signals. For example, when sending 100 signals to the server, only a few of them experience unexpected large delay due to traffic congestion (e.g., queuing at intermediate routers). This type of instability is the most common case and is handled by removing the outlier RTL values before computing the mean and the standard deviation (i.e., a statistical outlier removal algorithm: median absolute deviation method based on the Gaussian approximation).

Long-term instabilities are instabilities that stay long enough to affect the whole or most of the profiling signals but are not permanent. For example, if a user has a low bandwidth Internet, he/she will experience longer communications latency while his/her roommate is watching an HD movie online. This type of instability is a less common case and usually happens on the local network. CLAS addresses this case by increasing the error tolerance of the user according to his/her historical records and utilizing out-of-band-channels to enhance the access decision accuracy. Additionally, shared increment removal algorithm based on the design of multiple profiles to address the long-term instabilities is planned for the future work.

Routing instabilities are instabilities that result in permanent changes in network communications latency, due to, for example, permanent network routing changes. It has been shown that most of the important IP prefixes have stable routes and that instabilities only exist in a small portion of the global Internet [34–37] and, hence, the impact of these instabilities on CLAS is limited. In [34], the authors examine BGP routing information at five of the major US network exchange points: AADS, Mae-East, Mae-West, PacBell, and Sprint. The experiments show that the majority of routing information exhibits the same significant weekly, daily, and holiday cycles. More importantly, the experiments show that network instabilities exhibit strong periodicity. A recent study [38], which is based on 3-year daily data and 8-year monthly data, confirms the results of the earlier studies and further shows that routing changes have strong weekly periodicity, despite the overall growth in the size of the Internet. Consequently, the impact of this type of network instabilities in CLAS is marginal. Even for the small portions of global Internet with obvious instabilities, the instabilities exhibit strong periodicity and temporal properties. Therefore, CLAS can leverage these temporal properties and the strong periodicity of these instabilities to create long-term dynamic profiles. For example, CLAS can create temporal profiles such as daytime profile, night profile, weekday profile, and weekend profile. Also, these temporal profiles can be dynamically adjusted according to the user's valid login records or by using continuous profiling. In continuous profiling, the reference profile is updated with every successful login. Using such long-term dynamic temporal profiling alleviates the impact of network instabilities on CLAS.

In addition, for any sudden, unexpected, significant, and permanent changes, which are rarely incurred, users can use the login failure techniques discussed in Section 5 to rebuild their reference profiles.

4. Security Analysis

In this section, we discuss the defense methodologies of CLAS for many types of attacks.

4.1. Physical Observation. Refers to the potential leakage of authentication credentials by physical observation of users during login (e.g., shoulder surfing). It is important to note that the end user does not know and does not need to know his/her profile parameters. The profile parameters are measured, stored, and used by the authenticating server (as detailed in Section 3.3). Therefore, it is completely resilient to physical observation.

4.2. Internal Observation. It refers to the attack in which a perpetrator captures credentials by intercepting the input of the user inside his/her device. As detailed in Section 3.3, CLAS does not require any user input. Therefore, it is completely resilient to internal observation.

4.3. Password Compromise. In CLAS, even if the traditional credentials of the user are compromised, attackers cannot use them to successfully login from arbitrary locations. For such attacks to succeed, in addition to compromising credentials, attackers have to log in from locations that have similar parameters to those of the profiled location of the legitimate user, which is extremely hard to achieve. To have similar parameters to those of the legitimate user, the attacker has to either use brute force tactics or simulate the legitimate parameters. In the first case, the attacker tries to log in from multiple different locations until he/she, hopefully, succeeds in hitting a matching location. This can be easily thwarted by using the common practice techniques of enforcing maximum number of login retries. In the latter case, the attacker must know the legitimate profile parameters to be able to successfully simulate them.

4.4. Leaks from Other Verifiers. We note that many people use the same password on multiple services (web mail account, social account, billing account, etc.). Table 2 presents a summary of multiple surveys about password reuse [9–13]. The summary clearly shows that 77% of the surveyed people

Table 2: Password reuse survey summary.

Institute	Sample size	Nation	Reuse
2010 BitDefender [9]	Over 250k	Online	75%
2011 CIS [10]	Over 1k	Australia	63%
2012 CSID [11]	1,200	USA	61%
2013 OFCOM [12]	1,805	UK	55%
2014 UIUC [13]	224	Online	77%

in 2014 reused passwords across multiple services. For CLAS, thanks to the SRs, profile parameters across different verifiers are independent, and, hence, user profiles in one verifier are decoupled from his/her profiles in other verifiers.

4.5. Active Phishing. Phishing is a type of man-in-the-middle (MitM) attack. Active phishing describes the attack in which perpetrators use forged websites to capture authentication credentials and then use them in real-time to impersonate legitimate users. CLAS could defend against active phishing because its users do not send credentials that could be intercepted by attackers.

4.6. Mimic Attack. There are two possible ways to "mimic SR to get similar RTL": (1) the attacker takes down the SR and replaces it; (2) the attacker tries to mimic the RTL by getting closer to the location of the SR. The first type of attacks is very difficult to perform because the SR is assumed to be located in a secure place and the communications between the server and SR are protected by strong authentication and encryption mechanisms. The second type of attacks requires two things to materialize: (1) the perpetrator needs first to find the physical location of the SR and get close to it and (2) the perpetrator needs to try many times to brute force the RTL of the user. However, it is hard for the attacker to find out the exact location of the SR because the profiling signal received by the user has the IP of the server as the source address, while the IP address of the SR is only known to the server (as detailed in Section 3.3). Moreover, brute force attacks can be easily countered by limiting the number of login attempts within a certain time period. For example, the user's account will be frozen after 3 consecutive fail login attempts and the corresponding user will be notified via an out-of-band channel (e.g., email or SMS). Additionally, the security of CLAS can be further elevated by using multiple SRs. The server periodically changes the reference profile of each user by reprofiling using a different SR, which helps to not only defend against SR compromise attempts but also defeat guessing attempts and increase reliability.

5. CLAS Extensions

In this section, we discuss the potential integration of additional profiling features with baseline to achieve more robust and flexible defenses against password compromise. Such integration mainly targets the mobility and same location challenges that the baseline CLAS faces. We outline here the high level solutions to these challenges and leave the detailed implementation and validation for a future work.

5.1. Mobility and Legitimate Login Failures. CLAS identifies users based on the mean and the standard deviation of the network communications latency, which is highly dependent on the login location of the user. Two users have the same location if they are connected to the same local area network segment (for example, the same switch/hop), connected to the same access point, or connected to the same 3G/4G cell. Therefore, baseline CLAS may fail in the case of mobile users. If a legitimate user logs in from a location other than the profiled one, he/she may be denied access with high probability. Additionally, due to Internet instabilities, the user may sometimes fail to log in from his/her profiled location.

In this section, we propose to augment baseline CLAS with solutions that can handle both mobility and legitimate login failures.

Selective Mobility. Selective mobility refers to the case in which a user frequently logs in from a set of locations such as home, office, and library. In this case, CLAS simply creates a separate profile for each location. For each new location, the user registers with the service provider using exactly the same protocol used for the first-time account registration (Section 3.2). The server builds a new reference profile for each new location following exactly the same protocol used to build the first-time account reference profile (Section 3.2). A user will be granted access if his/her real-time login profile matches any of the stored profiles. The advantage of this solution is that it maintains the enhanced security benefits of CLAS (resiliency to MitM, phishing, etc.).

Arbitrary Mobility. Arbitrary mobility refers to the general mobility pattern, in which users may log in from arbitrary locations. We propose three options to handle the general mobility pattern.

(i) In the first option, the user uses one of his/her profiled locations to obtain a temporary token that can be used later to log in from a new nonprofiled location. For example, if the user plans to travel for a 3-day conference, he/she first logs in from one of his/her profiled locations and requests a temporary token with specific validity period. The system sends back a temporary token to the user. The user can then provide the temporary token as a second authentication code used by the server to bypass the regular profile authentication. This allows the user to log in from any location during the validity period of the temporary token.

(ii) In the second option, users may specify out-of-band communication channels (e.g., email, phone, etc.) as part of their reference profiles. These additional channels can be used later to deliver temporary authentication codes to users when logging in from new nonprofiled locations or when the login fails after exhausting the maximum number of login retries.

(iii) In the third option, CLAS can be integrated with any traditional second-factor authentication mechanism. The advantage of the second option over the third is that it does not require any other authentication scheme besides CLAS. The usability of the last two options is obviously better than that of the first. This is because the user gets authentication codes on-demand and uses them in real-time without carrying the burden to a priori request and safely store codes. However, both options are susceptible to the same multifactor limitations mentioned earlier in the Introduction. Therefore, the last two options, while providing the same security level as 2-factor authentication, lack the enhanced security protection provided by CLAS.

5.2. Same Location Attacks. As mentioned earlier in the Introduction, CLAS can tolerate compromise of either the password or the profiled location of a user but not both, which makes it vulnerable to what we call same location attacks. Same location attacks can only be launched by attackers who possess the password and have access to the profile location of the legitimate user. To thwart such sophisticated attacks, we propose to integrate CLAS with additional profiling features, such as browser/hardware fingerprints [39–41] or keystroke dynamics [42].

In browser fingerprinting [39], browser characteristics such as browser plug-ins, time zone, user agent, system fonts, and other measurements are used to generate unique fingerprints, which can be used to identify users. Multiple devices and multiple agents per device can be profiled to accommodate the flexibility of possible multidevice multiagent need. Keystroke dynamics, exhibited in a user's typing pattern, provides a unique signature to authenticate the user. Latency between successive keystrokes, keystroke durations, finger placement, and applied pressure on the keys can be used to construct a unique signature per individual [42].

Augmenting CLAS (location authentication) with browser/hardware fingerprinting (device authentication), keystroke dynamics (user authentication), or a combination of both can effectively defend against same location attacks. For example, when a login attempt passes the RTL check but presents a different browser fingerprint or keystroke dynamics, CLAS will raise an alert and request the user to answer a few security questions or authenticates using a second channel. In addition, the integration can also increase the authentication sample space such that the security guarantee of CLAS could be further improved.

6. Experimental Evaluation

In this section, we evaluate the performance overhead and the security guarantees of CLAS. More precisely, we evaluate how likely it is for legitimate users to be denied access and how likely it is for a perpetrator in possession of the credentials of legitimate users to be granted access. Additionally, we evaluate the experience of the clients in terms of how much more time it takes them to log in using CLAS compared to the baseline system. Finally, we evaluate the extra resources consumed by CLAS compared to the baseline system. The baseline system is a system that authenticates using simple credentials of username and password. Specifically, our evaluation focuses on the following metrics:

(i) False negative rate (FN): the probability that a legitimate user fails to log in from his/her profiled location

(ii) False positive rate (FP): the probability that a perpetrator who possesses legitimate user credentials successfully authenticates on behalf of the legitimate user

(iii) Login latency overhead: the extra time it takes a user to successfully authenticate using CLAS compared to a baseline system that authenticates using only username and password

(iv) Bandwidth and storage overhead: the extra network bandwidth incurred and the extra storage required by CLAS

For all these metrics, the lower the value is the better. CLAS mainly has four parameters: the error tolerance (ET), the maximum number of failed login retries (LR), the server workload (WL) in terms of login requests per second, and the number of profiling signals (N). Both ET and LR impact the trade-off between FN and FP. Intuitively, the higher the ET or the higher the LR, the lower the FN, but the higher the FP. Additionally, the higher the WL or N, the higher the login latency and the bandwidth overhead. Next, we are going to present the experimental results.

6.1. Experimental Setup. We build a password-based web login service that supports CLAS authentication. The service is implemented using HTML and PHP and runs on an Apache HTTP Server Version 2.4 [43] *The prototype website can be reached at* `52.24.162.70/login.html`. The server runs Ubuntu 12.04 on an Amazon EC2 instance. Users' credentials (i.e., username, password, profile mean, and profile standard deviation) are stored in a *MySQL* database that runs on the same server. The programming interface that constructs users' profiles is implemented on top of the MAC layer at the server. The interface implements (in Java and C) the logic that generates and sends profiling signals, measures delay, computes profile parameters, and takes access decisions. We configure and deploy another PC (Ubuntu 12.04 LTS 64-bit operating system, 4 GB memory, Intel i5 CPU ×2) to perform the SR functionality. We develop a programming interface on top of the MAC layer at the SR, which directly captures and forwards related packets in the cache of the network interface card. This interface is responsible for relaying profiling signals from the server to the user. We disable all the application layer traffic in both directions in SR (i.e., *sudo iptables-p INPUT OUTPUT DROP*) and disable Internet forwarding (i.e., *sudo echo "0" > ip_forward*) to

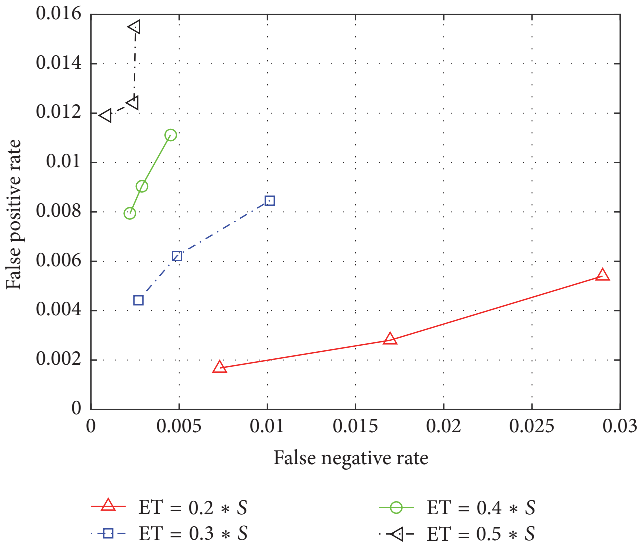


Figure 3: The ROC curves of the FN and the FP, when varying N; WL = 200; LR = 3.

prevent any potential attacks from the application layer. AES (advanced encryption standard) is used to secure the server-SR channel by encrypting the profiling signals before being sent to the SR.

The client population consists of 10 instances of Amazon EC2, 130 PlanetLab nodes, 25 GENI nodes, and 63 residential users randomly selected from different places in USA. Each client runs a browser that logs in to the web service. The selection of the operating system and the browser at the client does not affect the results and hence any option can be used. All our clients run either Chrome or Firefox browsers on Ubuntu 12.04 LTS operating system.

In all the following experiments, unless otherwise stated, we use $N = 91$, WL = 200 requests per second, LR = 3, and the login occurs during weekday time periods.

6.2. FN and FP Trade-Offs. To measure FN rate, we set each of the 25 GENI clients to uniformly log in 300 times during 4 different time periods: weekday daytime, weekday nighttime, weekend daytime, and weekend nighttime. The daytime period extends between 1 p.m. and 5 p.m. and the nighttime period extends between 9 p.m. and 2 a.m. The FN rate is computed as the number of failed login trials divided by the total number of login requests. To measure FP rate, we create and profile a different user account for each of the 155 clients (GENI + PlanetLab). Then, we set each of the 25 GENI clients to play attackers' role who try to log in using the credentials of the remaining 154 clients. Each attacking client launches 300 attacks against each of the remaining clients. Therefore, the total number of attack instances is 1,155,000 $(154 \cdot 300 \cdot 25)$. Each failed login is retried for a maximum of 3 times. The FP rate is measured as the number of successful attack trials divided by the total number of attack trials. The EC2 instances are used to create the appropriate server workload.

6.2.1. Varying the Number of Profiling Signals (N). We study here the trade-off between FP and FN, when we vary the number of profiling signals. Figure 3 shows the FP-FN ROC (receiver operator characteristics) curves of four ET values $(0.2 \cdot S, 0.3 \cdot S, 0.4 \cdot S$, and $0.5 \cdot S)$ when we vary N (27, 54, and 91). The server workload is set to 200 requests per second and the LR is set to 3 trials. In the figure, the lower the value, the better both FN and FP. The figure shows negative correlation between FN and FP as a function of the error tolerance. As the error tolerance increases, FN decreases, while FP increases. We see that CLAS can achieve FP of around 0.004 for FN around 0.002 when ET equals $0.3 \cdot S$.

6.2.2. Varying the Error Tolerance (ET). We study here the trade-off between FN and FP, when we vary the ET. Figure 4 shows the FP-FN ROC curves of three N values (27, 54, and 91) while varying the ET. The server workload is set to 200 requests per second and LR is set to 3 trials. The figure shows positive correlation between FN and FP when varying N. Both FP and FN improve when N increases. The figure shows poor FN and FP behavior when N is 27. However, the enhancements in FN and FP diminish beyond $N = 91$. This behavior, as the mathematical analysis suggests, is because both FN and FP are proportional to the square root of N. In other words, FP and FN are only marginally improved by further increasing N. On the other hand, the higher the value of N, the higher the login latency overhead and the bandwidth overhead. Most importantly, the figure presents the trade-off between the security guarantees and the functional performance of CLAS for the selected parameters. For example, the figure shows that CLAS can achieve FP as low as 0.0017 while FN is below 0.007. In other words, out of 1000 legitimate login trials, there are less than 7 fails on average. Additionally, perpetrators who possess user's credentials have only 1.7 in 1000 chances to authenticate on

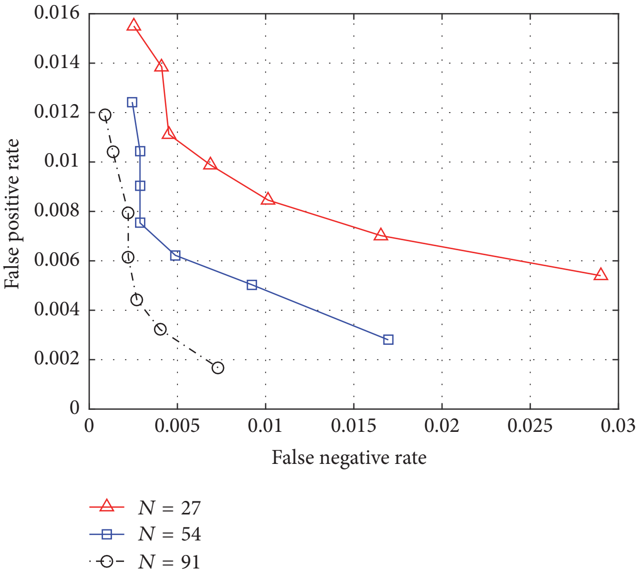


Figure 4: The ROC curves of the FN and the FP, when varying ET; WL = 200; LR = 3.

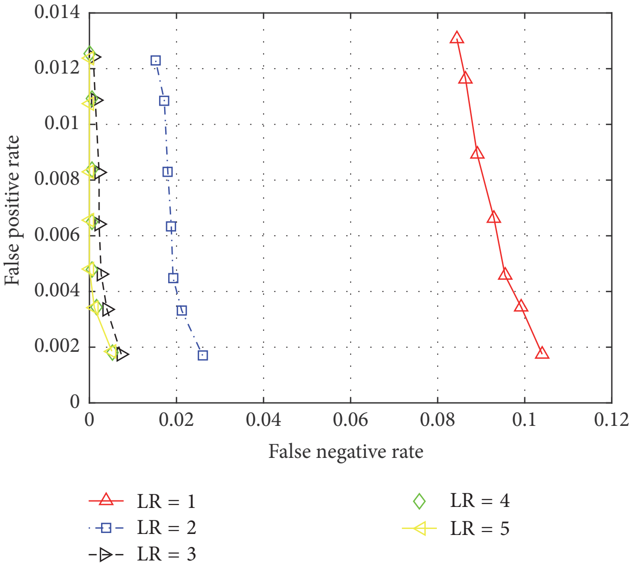


Figure 5: The LR ROC curves of the FP and the FN when varying ET; WL = 200.

average. This can be achieved when ET equals $0.5 \cdot S$, N equals 91 signals per login, LR equals 3 retries, and WL equals 200 requests per second. Other trade-offs, such as tolerating higher FP to achieve lower FN, can be achieved by selecting the appropriate operating point on the ROC curve. For example, CLAS can achieve FN of around 0.0008 when FP around 0.01.

6.2.3. Varying the Maximum Number of Login Retries. Figure 5 shows the FN-FP ROC curves for different LR values while varying the ET. Each curve represents one LR value between 1 and 5. The figure, intuitively, shows that LR has more impact on FN compared to FP. The chances that the legitimate real-time profile parameters match the profiled parameters in a next login trial is way higher than the chances of a perpetrator's real-time profile parameters that match the legitimate profiled parameters in the next trial. The figure shows that LR of 3 provides the best trade-off between FP and FN because the impact of LR on the trade-off diminishes with higher LR values. Again, the figure shows that CLAS can achieve FP as low as 0.0017 while FN is below 0.007.

Figure 6 shows the impact of the server workload on the trade-off between FP and FN rates. The figure shows small variations in FP and FN rates when the workload varies. In general, as the workload increases both FN and FP rates slightly increase. The variations are mainly due the slightly extra delays at the server when sending

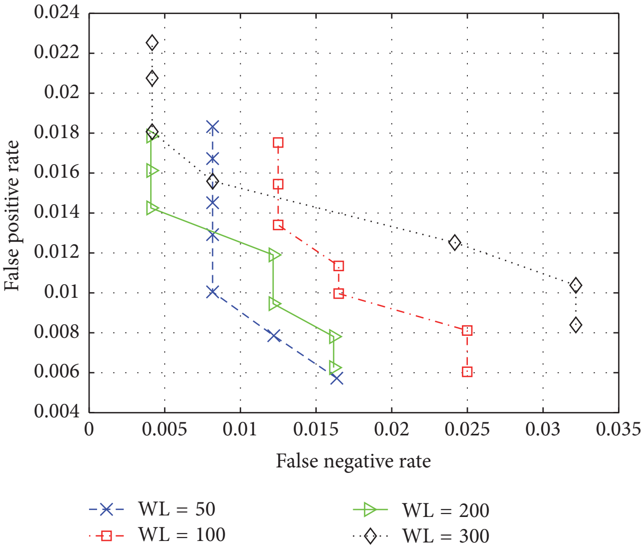


FIGURE 6: The WL ROC curves of the FP and the FN when varying ET; $N = 91$; LR = 3.

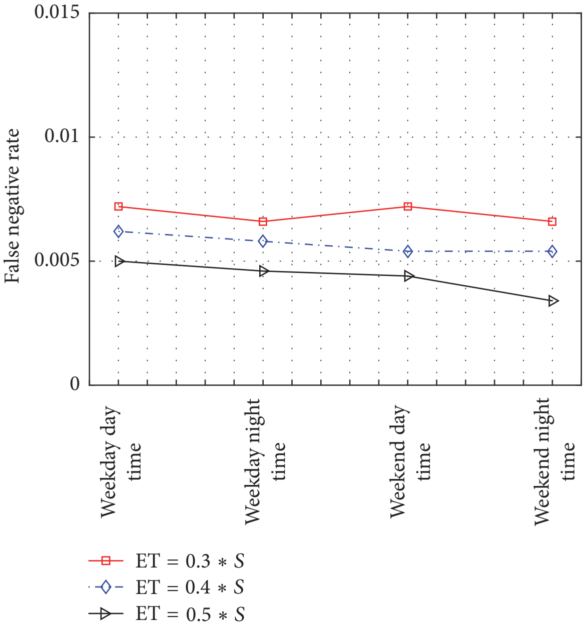


FIGURE 7: The impact of the login time period on the FN; $N = 91$; WL = 200; LR = 3.

the profiling signals, receiving the corresponding acknowledgments, and computing the real-time profile parameters.

6.2.4. Varying the Login Time. We study here the impact of the login time on FN. We set each of the clients to log in for 300 times during the 4 different time periods mentioned earlier. The daytime period extends between 1 p.m. and 5 p.m. and the nighttime period extends between 9 p.m. and 2 a.m. Figure 7 shows that, in general, the FN is almost stable irrespective of the login time period. However, FN is slightly higher during the weekday due to the impact of peek time on network communications latency.

6.2.5. Theoretical versus Experimental Results. Figure 8 compares the theoretical and the experimental FN and FP as a function of ET for $N = 91$, LR = 3, WL = 200, and weekday time login. The detailed mathematical analysis of FP and FN is presented in Appendix A. The figure shows that our mathematical model provides acceptable approximate value for FP and FN. The figure also shows that the mathematical model provides a good approximation for FP across the whole range of ET. On the other hand, the mathematical model of FN becomes more accurate with higher ET values.

6.2.6. Residential User Study. In the previous experiments, we evaluate the functionality and the security guarantees of

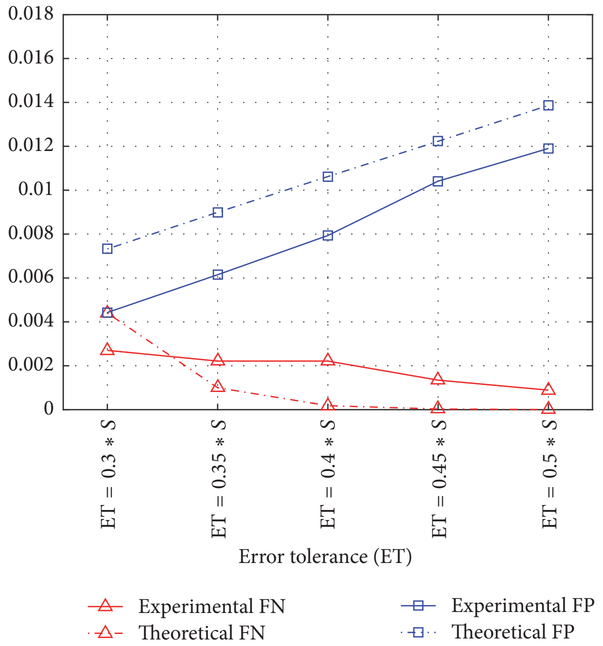


FIGURE 8: Theoretical versus experimental FP and FN, when varying the ET.

CLAS under various system parameters for enterprise and cloud users (for simplicity, we refer to those users as fixed users). In this experiment, we study the security and the functionality of CLAS for residential users. Residential users usually have less stable connections compared to enterprise and cloud users due to varying traffic and congestion of the last wireless hop. Such less stable connections may result in higher packet loss rate and, hence, higher variance in network communications latency. Figure 10 presents an example of the RTL values for (a) a fixed user (b) a residential WiFi user and (c) the residential WiFi user after outlier removal (i.e., remove extreme values before computing the mean RTL). In this experiment, each user type tries to log in every 12 minutes for 50-hour period from Friday 8:00 a.m. to Sunday 10:00 a.m. The figure shows that the RTL values of the residential WiFi user have larger variance compared to that of the fixed user, and this variance has been efficiently alleviated by utilizing any standard outlier removal algorithm [44].

In a second experiment, 63 residential users in Canada and USA use CLAS to authenticate to our testbed web service over a one-month period using home/work WiFi connections. Figure 11 compares FP-FN ROC curves of the two user types (residential and fixed). However, we believe that if the congestion rates are high and the last wireless hop quality is poor and unpredictable as in the case of busy public WiFi connections, FN will be high. In such scenarios, CLAS falls back to other authentication options such as browser fingerprinting or second-factor authentication to verify users' identity. As expected, the figure shows that the average FN rate of residential users is higher than that of the fixed users for the same average FP rate. For example, the residential users' FN rate is 0.0094 compared to 0.007 for fixed users when the average FP rate for both is 0.0048.

6.3. Performance Overhead

6.3.1. Login Latency Overhead. In this experiment, we study the variations of the login latency overhead under different server workloads. The login latency overhead is defined as the extra time it takes a user to successfully authenticate using CLAS compared to a baseline system, which authenticates using only username and password. The login latency is an important indicator for users about the usability of the system. The appropriate WL is generated by the 10 Amazon EC2 instances. We set each of the GENI clients to log in 300 times within one-hour period and measure the login latency overhead for each of the 7500 logins ($25 \cdot 300$) under workloads of 1, 10, 100, 200, and 300 requests per second. The login latency is measured at the client as the time elapsed from sending the login request until access is granted. Figure 9 shows the empirical cumulative distribution function (CDF) of the login latency overhead for each server workload. The figure shows that more than 95% of logins have overhead latencies below 0.2 seconds with WL up to 200 requests per second. The latency overhead experienced with 300 requests per second slightly increases to less than 0.24 seconds for more than 95% of the logins. This increase is mainly due to the queuing delays at the server. As the number of requests per second reaches the physical limits of the testbed server, some requests are queued for later processing. The results clearly show that the login latency overhead is unnoticeable by humans and hence the performance overhead of our scheme is negligible.

6.3.2. Bandwidth and Storage Overhead. To reduce the login latency overhead and the bandwidth overhead of the authentication process, CLAS leverages the regular login packets that are normally sent to users during login to serve as

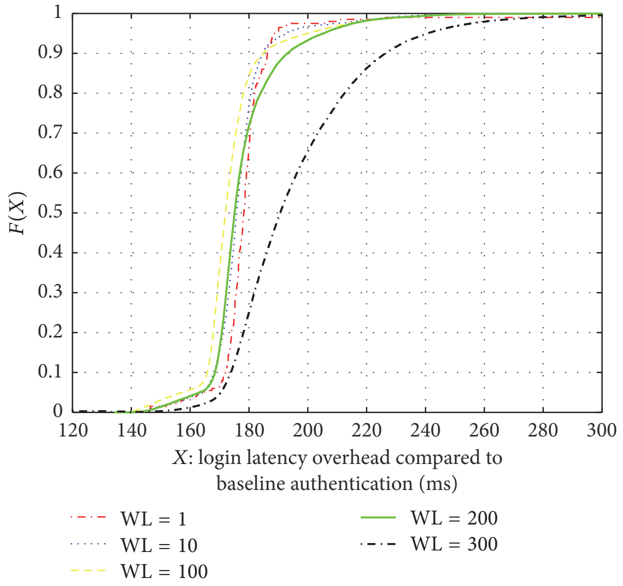
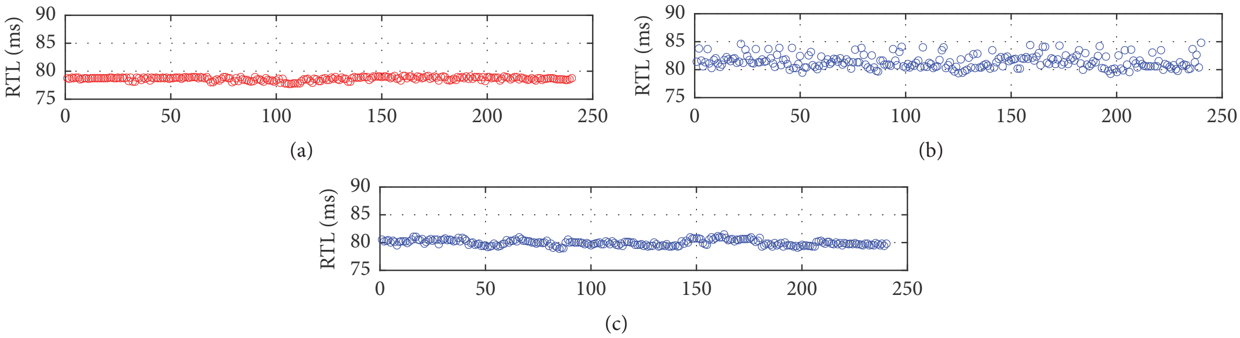


FIGURE 9: The CDF of the login latency overhead under various WL; $N = 91$; LR = 3; ET = $0.3 \cdot S$.



(a)

(b)

(c)

FIGURE 10: RTL values over 50-hour period for (a) fixed user, (b) residential WiFi user, and (c) residential WiFi user with outlier removal.

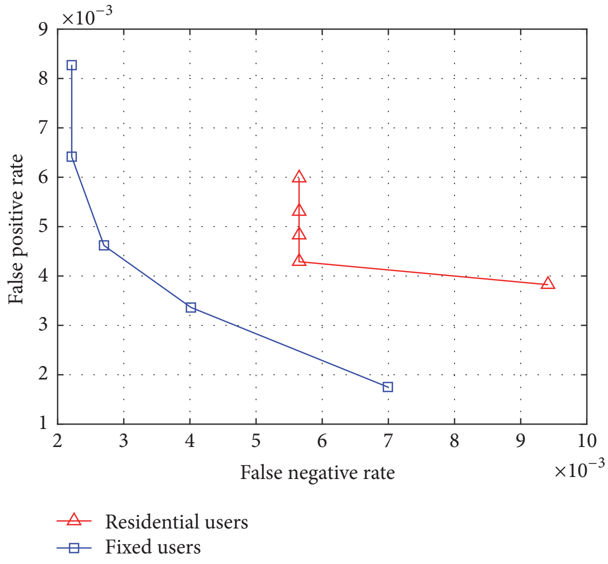


FIGURE 11: FP-FN ROC curves for fixed and residential users; $N = 91$; LR = 3.

Table 3: The approximate number of login page packets for a sample of popular websites.

Website name	Number of packets sent to user before entering username and password Cached Yes	 No
Chase	10	354
Bank of America	10	140
City Bank	24	450
Wells Fargo	21	55
FNC	10	500
HSBC	23	556
US Bank	15	248
SunTrust Bank	33	470
Capital One	62	260
Bank of Montreal	9	683
Facebook	13	168
Amazon	150	440

profiling signals. Most of the commercial websites (e.g., e-banking, e-government, web-based email, etc.) send many packets to users during the login process. We run an experiment to count the number of regular login packets sent to users for a sample of popular websites. Table 3 shows the results. If the website is cached, the approximate number of regular login packets ranges between 10 and 150. If the site is not cached, the approximate number of packets ranges between 55 and 683. Specialized profiling signal packets are generated only if the number of regular login packets is less than the number required to measure sufficiently accurate profile parameters. Based on Table 3 and the mathematical and experimental analysis of the required number of profiling signals (Section 3 and Appendix A), the website login content packets are sufficient to construct users' profiles in most of the websites. Therefore, the bandwidth overhead of CLAS is negligible.

The storage overhead is also negligible. In addition to username and password, the credentials database stores, for each user, profiled mean, error tolerance, and standard deviation range. This overhead adds less than 32 bytes per user even when using floating point representation (8 bytes) of profile parameters.

6.3.3. Impact of the WL on Profile Parameters. In this experiment, we measure the impact of the server workload on the measurements of the real-time profile parameters. We set each of the GENI clients to log in 300 times within one-hour period under workloads of 1, 10, 100, 200, and 300 requests per second. For each client we measure the average mean and the average standard deviation of the 300 logins under each workload. Figure 12 shows the empirical cumulative distribution function (CDF) of the relative maximum variations of the mean and the standard deviation of each client. For each client, the relative maximum variation in the mean is computed as the percentage of the maximum variations in the mean relative to the average mean across all server workloads. Similarly, the relative maximum variation in the standard deviation is measured as the percentage of the maximum variations in the standard deviation relative to the average standard across all server workloads. The figure shows that more than 90% of the clients have relative maximum mean variations less than 6.5% of the average mean. In other words, if the mean is 100 ms, the maximum variation in the measured mean under different server workloads is less than 6.5 ms. Additionally, the relative maximum standard deviation variation is less than 3.3% for more than 97% of the clients. These results explain the earlier behavior of FN and FP when varying WL (Figure 6). WL may only slightly affect the measured profile parameters and, hence, FN and FP are marginally affected by WL.

7. Conclusions

In this work, we design and implement a novel highly secure and usable scheme (CLAS), which complements the state-of-the-art authentication mechanisms and strengthens the security of web authentication. CLAS uses, in addition to the traditional credentials, the round trip network communications latency (RTL) to uniquely profile users. The novel architecture of CLAS makes its profiling parameters robust and highly resilient to manipulation. It is highly unlikely for attackers, and even legitimate users, to learn and manipulate profile parameters. This key security feature protects CLAS users against impersonation even when their traditional credentials are compromised. Moreover, CLAS, unlike many state-of-the-art authentication mechanisms, is resilient to active phishing, man-in-the-middle, and social engineering attacks. More importantly, CLAS is completely transparent to end users. Finally, CLAS can be augmented with additional profiling features such as key stoke dynamics, as well as other authentication mechanisms to offer more robust and flexible web authentication.

Our analysis and experiments show that CLAS can achieve false positive rate as low as 0.0017 while the false negative rate is below 0.007. Moreover, the results show that the login latency overhead is negligible and cannot be noticed by humans (i.e., less than 0.2 seconds).

In the future, we plan to develop and implement the techniques we propose to support anywhere authentication and to address the same location attacks. We also plan to develop more techniques to further alleviate the potential impact of network instabilities on RTL measurements. In addition, we plan to use standard software evaluation tools (e.g., function point [45]) to perform the cost and effort estimation of CLAS to ensure it is practical to use.

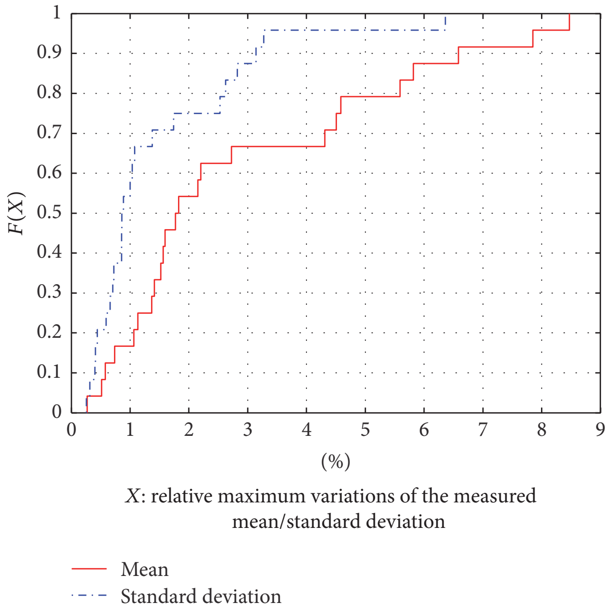


Figure 12: The CDF of the relative maximum variations in the mean and the standard deviation of the real-time profile, when varying the WL; $N = 91$; LR = 3; ET = $0.3 \cdot S$.

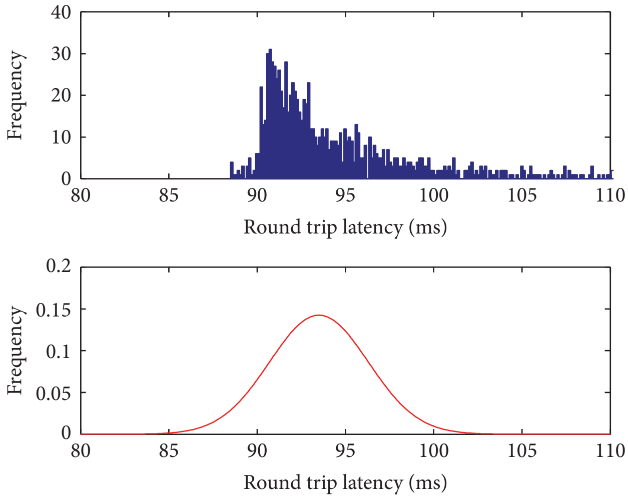


Figure 13: Sample of communications latency distribution and its Gaussian distribution.

Appendix

A. Mathematical Analysis

A.1. Gaussian Approximation of Network Delay. It has been observed that network communications latency approximately follows a Gaussian distribution [18]. This observation is validated by conducting extensive experiments to measure communications latency from a pool of 130 PlanetLab nodes [18]. Each one of the 130 nodes receives 400 TCP packets from one single server, which is located in University of Rochester, and sends them back for measuring the round trip communications latency. We have also conducted similar experiments using GENI nodes and come up with the same conclusion: *Gaussian is a good approximation for network communications latency*. Figure 13 shows an example of the latency distribution and the corresponding Gaussian approximation for a client pinging a sever with 300 packets. In our experiments, we measure pair-wise communications latency distribution between nodes randomly selected from a pool of 25 worldwide GENI nodes [46]. The outcome of all these experiments confirms that Gaussian is an acceptable approximation for the communications latency distribution.

A.2. False Negative Rate (FN). As mentioned in the previous section, the FN rate of the Gaussian distribution is α, which

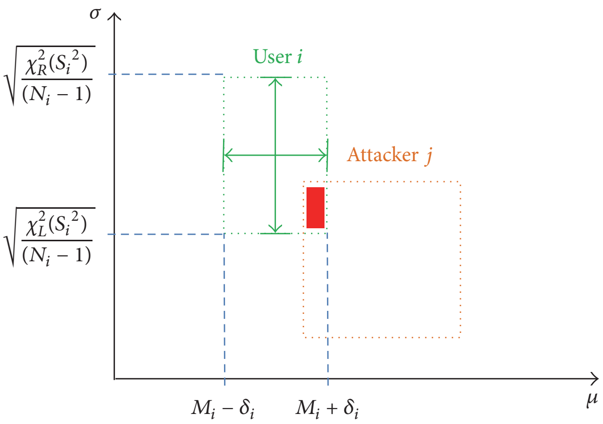

σ
$\sqrt{\frac{\chi_R^2(S_i{}^2)}{(N_i-1)}}$
User i
Attacker j
$\sqrt{\frac{\chi_L^2(S_i{}^2)}{(N_i-1)}}$
$M_i-\delta_i$
$M_i+\delta_i$
μ

Figure 14: Graphical illustration of arbitrary grant-access area.

stands for the probability that a legitimate user fails to authenticate from his/her profiled location. Let $\delta_i = C_i \cdot S_i$, where C_i is the error tolerance coefficient for user i and let U be the total number of users. To compute FN, we plug δ_i in (1):

$$Z_{1-\alpha_i} = C_i \cdot \sqrt{N_i},$$
$$\text{FN} = \frac{\sum_{i=1}^{U}\left(1 - Z^{-1}\left(C_i \cdot \sqrt{N_i}\right)\right)}{U}. \tag{A.1}$$

A.3. False Positive Rate (FP). The FP, β, is the probability that a perpetrator passes authentication from a location other than the profiled one. In other words, false positive rate is the probability that the real-time measured latency mean and standard deviation of the perpetrator fall within the grant-access area of the legitimate user. We first derive a simplified estimate of the false positive rate and then enhance the derivation accuracy. Figure 14 shows the grant-access area for an arbitrary user (User i) and the grant-access area for a perpetrator (Attacker j) at a random location. Recall that the perpetrator also possesses the username and password of the legitimate user. Assume, for now, uniform distribution of the measured mean and standard deviation within the grant-access area (the green rectangle in Figure 14 shows the reference profile area of an arbitrary user (User i), dubbed as the grant-access area (GAA)). Access is granted for any login attempt with measured (μ, σ) point that falls within the grant-access area. Also assume that the locations from which an attacker may try to log in are known. Then, the probability that Attacker j successfully authenticates as User i equals the overlap area between the grant-access area of the user and the grant-access area of the perpetrator divided by the grant-access area of perpetrator averaged over all possible attack locations:

$$\beta_i = \frac{\sum_{j=1}^{A}\left(\left(\text{GAA}_i \cap \text{GAA}_j\right)/\text{GAA}_j\right)}{A}, \tag{A.2}$$

where A is the number of all possible locations from which the attacker may try to impersonate the user. The overall false positive rate of the system is computed as the average of false positive rates of all the users of the system:

$$\text{FP} = \frac{\sum_{i=1}^{U}\beta_i}{U}. \tag{A.3}$$

Note that even though this is a simplified estimate of the false positive rate, we next show that it provides an upper bound approximation of the false positive rate.

To derive a more accurate estimate of the false positive rate, we need to identify the real distribution of latency mean and standard deviation within the grant-access area, rather than just assuming it to be uniform. Moreover, we need to remove the assumption of previously known attack locations by acknowledging that attackers may use any arbitrary previously unknown location to log in. Let the mean of network communications latency be a random variable X in the range $[a, b]$ and let the standard deviation of the mean X be a random variable Y in the range $[c * X, d * X]$. According to the conclusions derived in [18], which is also validated by our experiments, both X and Y are approximately Gaussian with the following probability distribution functions (pdf):

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \cdot e^{-(x-\mu_x)/(2\cdot\sigma_x^2)},$$
$$f_Y(y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \cdot e^{-(y-\mu_y)/(2\cdot\sigma_y^2)}. \tag{A.4}$$

For the systems analyzed in [18], $a = 5\,\text{ms}$, $b = 700ms$, $c = 0.0155$, and $d = 0.196$. Therefore, the sample space of the communications latency X and its standard deviation Y

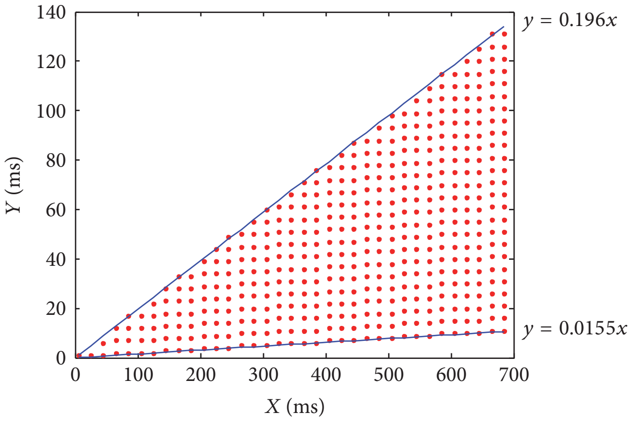


Figure 15: Sample space of the communications latency mean and standard deviation.

falls in the shaded area shown in Figure 15. Based on these values, the pdf parameters of X and Y distributions are

$$\begin{aligned}\mu_x &= 221,\\ \sigma_x &= 83.86,\\ \mu_y &= 0.0155x + \frac{(0.196x - 0.0155x)}{2} = 0.1058x,\\ \sigma_y &= \frac{\left(y_{0.005} - \mu_y\right)}{Q^{-1}(0.005)} = 0.035x.\end{aligned} \tag{A.5}$$

Using Figure 14 and assuming error tolerance $\delta = C \cdot S$, the grant-access area can be computed as

$$\begin{aligned}\text{GAA} &= [(M+\delta) - (M-\delta)]\\ &\quad \cdot \left[\sqrt{\frac{\chi_R^2 \cdot S^2}{N-1}} - \sqrt{\frac{\chi_L^2 \cdot S^2}{N-1}}\right].\end{aligned} \tag{A.6}$$

According to the experimental results, the optimal false positive rate occurs when $\delta = 0.2 \cdot S$. Therefore,

$$\text{GAA} = \frac{S^2}{10}. \tag{A.7}$$

Using (A.4), (A.5), and (A.7), the expected value of the grant-access area is computed as

$$\begin{aligned}&\int_{x=a}^{b} \int_{y=c\cdot x}^{d\cdot x} \frac{S^2}{10} \cdot \frac{1}{\sqrt{2\pi\sigma_x^2}} \cdot e^{-(x-\mu_x)/(2\cdot\sigma_x^2)} \cdot \frac{1}{\sqrt{2\pi\sigma_y^2}}\\ &\quad \cdot e^{-(y-\mu_y)/(2\cdot\sigma_y^2)} dx\, dy\\ &= 4.74 \times 10^{-17} \cdot \int_{x=5}^{700} \int_{y=0.0155x}^{0.196x} \frac{y^2}{x} \cdot e^{-7.11\times10^{-5}\cdot x^2}\\ &\quad \cdot e^{0.0314x + 86.367\cdot(y/x) - 408.16\cdot(y^2/x^2)} dx\, dy.\end{aligned} \tag{A.8}$$

The false positive rate is the probability that an attacker at a random location successfully impersonates a legitimate user, which is given by

$$\text{FP} = \frac{E\,[\text{GAA}]}{\text{Area of the Sample Space}} \tag{A.9}$$

Using (A.9), for the systems analyzed in [18], the expected value of the false positive rate is approximately 0.0034. In other words, if the attacker tries to log in from 1000 different locations, on average, he/she successfully authenticates from less than 4 of them. This is a very low probability and hence clearly proves the high security guarantees of CLAS.

Using (A.3), for the systems analyzed in [18], the expected value of the false positive rate is approximately 0.0068. Therefore, the simplified analysis provided earlier provides an upper bound estimate of the false positive rate.

A.4. False Positive and False Negative Trade-Offs. Optimally, we need to keep both false positive and false negative rates very low. However, these two indicators are dependent. Decreasing the false negative rate increases the false positive rate and vice versa. A possible trade-off is to maximize the security guarantees while maintaining an acceptable functionality level. Using the false positive and false negative formulas developed in Appendix A.3, the trade-off can be translated into the following optimization problem:

$$\begin{aligned}\text{Minimize}\quad \text{FP} &= \frac{\sum_{i=1}^{U} \beta_i}{U}\\ &= \frac{\sum_{i=1}^{U} \sum_{j=1}^{A} \left(\left(\text{GAA}_i \cap \text{GAA}_j\right)/\text{GAA}_j\right)}{A \cdot U}\\ \text{S.T.}\quad \text{FN} &= \frac{\sum_{i=1}^{U} \left[1 - Z^{-1} \cdot \left(C_i \cdot \sqrt{N_i}\right)\right]}{U}\\ &\le \text{FN}_{\text{required}}\\ &\qquad C_i \ge 0,\ N_i, U, A \ge 0,\ \text{integer}.\end{aligned} \tag{A.10}$$

Security administrators can use the optimization problem in (A.10) to guide their functionality and security configurations.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] "The Cyberwire: Breaking news from RSA, April 2015.," http://thecyberwire.com/issues/issues2015/April/CyberWire_2015_04_21.html.

[2] L-F. Aaron Han, F. Derek Wong, and S. Lidia Chao, Password cracking and countermeasures in computer security: A survey, arXiv preprint arXiv:1411.7803, 2014.

[3] R. W. Reeder and S. Schechter, "When the password doesn't work: Secondary authentication for websites," *IEEE Security and Privacy*, vol. 9, no. 2, pp. 43–49, 2011.

[4] H.-M. Sun, Y.-H. Chen, and Y.-H. Lin, "oPass: A user authentication protocol resistant to password stealing and password reuse attacks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 651–663, 2012.

[5] M. Adham, A. Azodi, Y. Desmedt, and I. Karaolis, "How to attack two-factor authentication internet banking," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7859, pp. 322–328, 2013.

[6] S. Gastellier-Prevost, G. G. Granadillo, and M. Laurent, "A dual approach to detect pharming attacks at the client-side," in *Proceedings of 4th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2011*, fra, February 2011.

[7] "ConsumerReports: Smartphone thefts, May 2014," http://www.consumerreports.org/cro/news/2014/04/smart-phone-thefts-rose-to-3-1-million-last-year/index.htm.

[8] "Internet Security Threat Report," Symantec, April 2015.

[9] BitDefender finds exposed social media credentials often provide access to email accounts. 2010. http://www.bitdefender.com/news/bitdefender-finds-exposed-social-media-credentials-often-provide-access-to-email-accounts-1682.html.

[10] "Password security: A survey of australian attitudes toward password use and management, 2011," https://www.paypal-media.com/assets/pdf/fact_sheet/cis_paypal_whitepaper_final.pdf.

[11] "Consumer survey: Password habits, 2012," http://www.csid.com/wp-content/uploads/2012/09/CS_PasswordSurvey_FullReport_FINAL.pdf.

[12] "Ofcom's adults media use and attitudes report, 2013," http://stakeholders.ofcom.org.uk/market-data-research/media-literacy/media-lit-research/adults-2013/.

[13] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse," in *Proceedings of the Network and Distributed System Security Symposium NDSS '14*, San Diego, CA, USA, 2014.

[14] N. Gunson, D. Marshall, H. Morton, and M. Jack, "User perceptions of security and usability of single-factor and two-factor authentication in automated telephone banking," *Computers and Security*, vol. 30, no. 4, pp. 208–220, 2011.

[15] L. Zhang, S. Yu, D. Wu, and P. Watters, "A survey on latest botnet attack and defense," in *Proceedings of 10th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communications, TrustCom 2011, 8th IEEE Int. Conf. on Embedded Software and Systems, ICESS 2011, 6th Int. Conf. on Frontier of Computer Science and Technology, FCST 2011*, pp. 53–60, chn, November 2011.

[16] K. Anup Ghosh, A. Schwartzbard, and M. Schatz, "Learning program behavior profiles for intrusion detection," *In Workshop on IDNM*, 1999.

[17] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *Communications Surveys and Tutorials*, 2014.

[18] M. Kwon, Z. Dou, W. Heinzelman, T. Soyata, H. Ba, and J. Shi, "Use of network latency profiling and redundancy for cloud server selection," in *Proceedings of 7th IEEE International Conference on Cloud Computing, CLOUD 2014*, pp. 826–832, usa, July 2014.

[19] I. Hababeh, I. Khalil, and A. Khreishah, "Designing high performance web-based computing services to promote telemedicine database management system," *IEEE Transactions on Services Computing*, vol. 8, no. 1, pp. 47–64, 2015.

[20] I. M. Khalil, "ELMO: Energy aware local monitoring in sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 4, pp. 523–536, 2011.

[21] R. K. Panta, S. Bagchi, and I. M. Khalil, "Efficient wireless reprogramming through reduced bandwidth usage and opportunistic sleeping," *Ad Hoc Networks*, vol. 7, no. 1, pp. 42–62, 2009.

[22] V. Vallivaara, M. Sailio, and K. Halunen, "Detecting man-in-the-middle attacks on non-mobile systems," in *Proceedings of 4th ACM Conference on Data and Application Security and Privacy, CODASPY 2014*, pp. 131–133, usa, March 2014.

[23] Gmail: Detecting suspicious account activity, http://googleonlinesecurity.blogspot.com//03/detecting-suspicious-account-activity.html.

[24] R. Dingledine, N. Mathewson, and P. Syverson, Tor: The second-generation onion router. Technical report, DTIC Document, 2004.

[25] "Dell secure work: Bgp hijacking for cryptocurrency profit, August 2014," http://www.secureworks.com/cyber-threat-intelligence/threats/bgp-hijacking-for-cryptocurrency-profit/.

[26] P. Vervier, O. Thonnard, and M. Dacier, "Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA, 2015.

[27] A. Gavrichenkov, *Breaking Https with Bgp Hijacking*, Black Hat, Briefings, 2015.

[28] S. Hogg, "Address authenticaion," *The Internet Protocol Journal*, 2013.

[29] I. Khalil, Z. Dou, and A. Khreishah, "Your credentials are compromised, do not panic: You can be well protected," in *Proceedings of 11th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2016*, pp. 925–930, chn, June 2016.

[30] K. O. Bailey, J. S. Okolica, and G. L. Peterson, "User identification and authentication using multi-modal behavioral biometrics," *Computers & Security*, vol. 43, pp. 77–89, 2014.

[31] K. Moritz, S. S. Aultman, J. J. A. Campbell et al., Behavioral profiling method and system to authenticate a user, November 10 2015. US Patent 9,185,095.

[32] NIST, e-Handbook of statistical methods. http://www.itl.nist.gov/div898/handbook/.

[33] Stat 300 materials 7-3a. http://flc.losrios.edu/eitel/Stat%20300/S-300%20Main%20Web%20Page.htm.

[34] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 515–528, 1998.

[35] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "BGP Routing Stability of Popular Destinations," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pp. 197–202, fra, 2002.

[36] M. Lad, J. H. Park, T. Refice, and L. Zhang, "A study of internet routing stability using link weight," Tech. Rep. UCLA/CSD-080003, 2008.

[37] A. Shaikh, A. Varma, L. Kalampoukas, and R. Dube, "Routing stability in congested networks: experimentation and analysis," in *Proceedings of the ACM SIGCOMM Computer Communication Review*, vol. 30, pp. 163–174, September 2000.

[38] G. Comarela, G. Gürsun, and M. Crovella, "Studying interdomain routing over long timescales," in *Proceedings of 13th ACM Internet Measurement Conference, IMC 2013*, pp. 227–233, esp, October 2013.

[39] P. Eckersley, "How unique is your web browser?" in *Privacy Enhancing Technologies: 10th International Symposium, PETS 2010, Berlin, Germany, July 21–23, 2010. Proceedings*, vol. 6205 of *Lecture Notes in Computer Science*, pp. 1–18, Springer, Berlin, Germany, 2010.

[40] Z. Dou, I. Khalil, A. Khreishah, and A. Al-Fuqaha, "Robust Insider Attacks Countermeasure for Hadoop: Design and Implementation," *IEEE Systems Journal*, pp. 1–12, 2017.

[41] I. Khalil, Z. Dou, and A. Khreishah, "TPM-based authentication mechanism for apache hadoop," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 152, pp. 105–122, 2015.

[42] F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, vol. 16, no. 4, pp. 351–359, 2000.

[43] "Apache HTTP Server Version 2.4, 2015," http://httpd.apache.org/docs/2.4/en/.

[44] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764–766, 2013.

[45] F. Ahmed, S. Bouktif, A. Serhani, and I. Khalil, "Integrating function point project information for improving the accuracy of effort estimation," in *Proceedings of 2nd International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP 2008*, pp. 193–198, esp, October 2008.

[46] GENI, http://www.geni.net/.

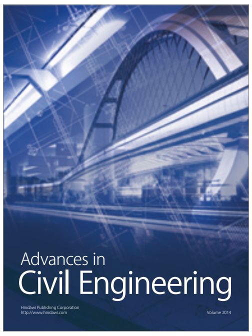
Journal of
Engineering

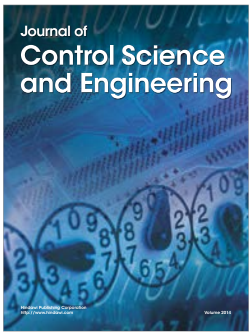
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

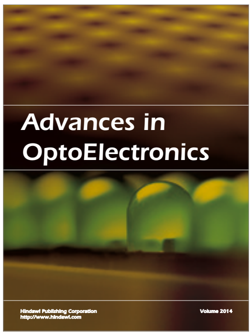
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

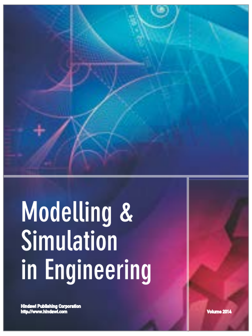
Journal of
Robotics

Journal of
Electrical and Computer Engineering

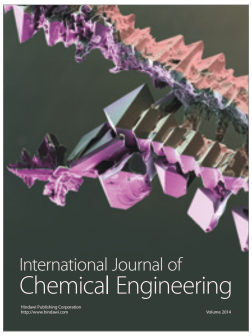

Advances in
OptoElectronics

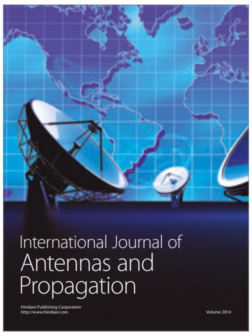
VLSI Design

International Journal of
Navigation and Observation

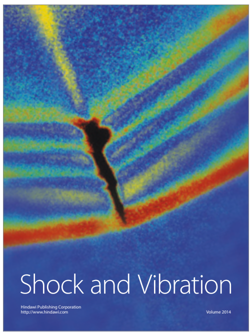

Modelling & Simulation in Engineering

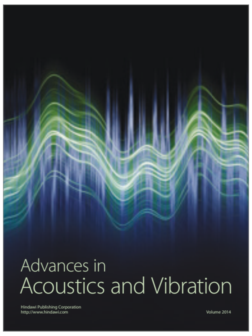
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration