# A Novel and Robust Authentication Factor Based on Network Communications Latency

Zuochao Dou, *Student Member, IEEE,* Issa Khalil, *Member, IEEE,* and Abdallah Khreishah, *Member, IEEE,*

***Abstract*—We propose a new authentication factor based on Network Round Trip Time ($NRTT$). We show how $NRTT$ can be used to uniquely and securely identify login locations and hence can support location-based web authentication mechanisms. The first research challenge is how to securely measure and verify $NRTT$ to hamper potential forgery attempts. We address the first challenge by introducing a novel forwarding device in the path between the server and the client, dubbed delay mask (DM), which prevents any entity, but the server, from being able to measure the $NRTT$ for any client. The second research challenge is how to reliably measure $NRTT$ in the face of variable Internet latencies and connectivity conditions. The second challenge is addressed by (1) computing the average of a number of $NRTT$ measurements after outlier removal; (2) applying multiple profiles per user through the deployment of multiple DMs in diverse geographical locations. We design a two-factor authentication scheme (dubbed AMAN) that uses legacy passwords as a first factor and $NRTT$ as a second authentication factor. We conduct extensive experiments to evaluate Security-Usability-Deployability properties of AMAN and compare it with state-of-the-art authentication mechanisms. The results show that AMAN achieves the best combination of these properties.**

***Index Terms*—Web service, Network communications latency, Gaussian distribution, Authentication, Password compromise**

## I. Introduction

Legacy password authentication suffers from many obvious usability and security limitations. The credentials of the users not only hacked through social engineering and dictionary attacks, but also databases storing such credentials have been hacked, exposing massive number of user accounts [2] [3]. To address the security concerns in legacy password-based authentication, many new authentication factors have been introduced and tested, including: (1) random strings delivered through out-of-band channels such as mobiles and emails; (2) human biometrics such as fingerprints and iris scans; (3) profile-based factors such as profiling normal user behavior, browser fingerprinting, IP address information, and login location; (4) physical factors such as cards, hardware tokens, and mobiles; (5) knowledge-based factors such as recognizing someone based on photos provided by social websites [4]. However, each of these authentication factors has its own inherent weaknesses and security limitations. For example, phishing is still effective even when using out-of-band-channels to deliver second factor Personal Identification Numbers (PINs) or passwords. Internal observation can also defeat many of these factors, especially human biometrics and system fingerprinting [5]. Additionally, some of these authentication factors are static, such as browser fingerprints and IPs, and hence can be forged or leaked across different verifiers. Physical factors, on the other hand, can be lost, stolen, or compromised. Furthermore, some factors have high false negative rate such as keyboard typing rhythms and end user profiling. It is also worth noting that, in addition to the security limitations, many of these factors have usability issues due to the requirement of extra information, devices, or channels. Appendix A provides a more detailed and systematic analysis and comparisons of the vulnerabilities of the state-of-the-art authentication factors.

In this work, we propose a new authentication factor that does not share the above mentioned properties with the commonly used authentication factors. That is, the new authentication factor is oblivious to clients and is not communicated to the server, but rather is completely measured and verified at the server. Our proposed authentication factor utilizes what initially appears to be counter-intuitive, the Network Round-Trip communications Time ($NRTT$). $NRTT$ is defined as the summation of the time a packet takes to travel from the server to the client and the time its acknowledgment takes to travel back from the client to the server. Authentication using $NRTT$ is straightforward. At registration, $NRTT$ statistics (i.e., mean and standard deviation) between the client and the authenticator are measured and stored at the authenticator as a reference profile. The $NRTT$ statistics are then re-measured with every login attempt in real-time, and login is granted only if the new statistics fall within predefined boundaries from the corresponding registration statistics.

It has been observed, through extensive experiments and monitoring of Internet communications, that $NRTT$ follows distributions that can be modeled as an approximate Gaussian distribution ([6] [7] [8]) as detailed in Section III-A. We adopt Gaussian approximation to theoretically guide the selection of different $NRTT$ related parameters in our experiments and theoretical analysis. The mean and the standard deviation of $NRTT$ measurements vary when the login location changes, that is, users can be uniquely identified based on their login locations. This important observation indicates that a login attempt will only succeed if conducted from the same location as that of registration, which reduces the attack surface of compromised identities from anywhere in the world to only the registration location. Note that, location-based authentication is very important in many areas, such as electronic health record access, sensitive financial transactions, military communications, industrial control systems, etc. [9] [10] [11].

Z. Dou is with the Electrical Computer Engineering Department, New Jersey Institute of Technology, Newark, USA, E-mail: zd36@njit.edu.

I. Khalil is with the Qatar Computing Research Institute, Hamad bin Khalifa University, Doha, Qatar. E-mail: ikhalil@qf.org.qa.

A. Khreishah is with the Electrical Computer Engineering Department, New Jersey Institute of Technology, Newark, USA, E-mail: abdallah@njit.edu.

Part of this work (a short paper) is published in the 11th ACM Asia Conference on Computer and Communications Security (AsiaCCS 2016), Xian, China [1].

In addition, within this work, the login location refers to the last network segment, access point or 3G/4G cell of the communicating party.

However, the naive measurement and usage of $NRTT$ has two main challenges: (i) Attackers can easily estimate $NRTT$ for any location by simply pinging the server from that location. This represents a serious security challenge because an attacker can impersonate a user by simulating her communications latency. As detailed in Section IV-D, this challenge has been addressed by inserting a special device, delay mask, in the round-trip path between the client and its authenticator. The goal of the DM is to prevent any entity, but the authenticator, from being able to estimate $NRTT$. This marks the two differences between $NRTT$ and the state-of-the-art authentication factors. The first difference is that the client does not know the value of its $NRTT$, which makes it resilient to client compromise. The second difference is that the value of $NRTT$ is not communicated to the server, but rather is computed by the server. (ii) Network instabilities may cause communications latency to vary and consequently result in legitimate login failures, which leads to poor user experience. As detailed in Section IV-D, this challenge has been addressed in two steps: (a) averaging of multiple $NRTT$ measurements after outlier removal to alleviate transient network instabilities, and (b) building multiple profiles per user through the deployment of multiple DMs in diverse geographical locations. Then, novel algorithms are applied to the multiple profiles to identify and isolate variances caused by changes in local network conditions such as congestion.

In this work, we show how to turn the insecure and potentially unstable $NRTT$ into a robust authentication factor that is resilient to both client compromise and communication channel compromise. $NRTT$ offers unique security features such as resiliency to Phishing, MitM, leakage by other verifiers, and social engineering, which complements the security features of other authentication factors. Moreover, $NRTT$ has the advantage of being user transparent (i.e., it does not require clients to memorize or input any information) and has negligible overhead, which enables it to be smoothly integrated with other authentication factors in multi-factor authentication schemes without introducing extra overhead or degrading usability. However, $NRTT$ can only be used to provide authentication for low mobility users and static users, similar to location-based authentication schemes. It is intuitive to see that arbitrarily mobile users cannot benefit from authentication based on $NRTT$ because such users require to be able to login from any arbitrary location, while $NRTT$ can only accept logins from previously profiled locations. Nevertheless, $NRTT$ provides reliable and secure location-based authentication, which is generally used to ensure that users can perform sensitive operations (e.g. change password, initiate funds transfers) or access valuable information (e.g., personal medical information) only from authorized locations. Additionally, secure location identification is important for other security purposes. For example, geographical location is one of the most commonly used indicators to detect phishing based on the observation that phishing websites are most likely to be hosted in locations different from those of the corresponding legitimate websites [12].

We summarize our contributions in this work as follows:

- Propose a novel secure and usable web authentication factor based on Network Round Trip Time, $NRTT$.
- Design and implement a novel network architecture that enables secure measurement of $NRTT$.
- Design and implement algorithms to alleviate network instabilities and expand authentication sample space of $NRTT$.
- Design, implement and deploy a prototype for a use case of two-factor authentication (AMAN) with legacy passwords as the first factor and $NRTT$ as the second factor. The prototype helps to practically evaluate the security, usability, and deploy-ability properties of $NRTT$-based two factor authentication and to assess its performance overhead.
- Provide comparative evaluation of $NRTT$ against state-of-the-art authentication factors using a famous authentication benchmark framework.

The rest of this paper is organized as follows: Section III presents background knowledge on Gaussian distribution and network instabilities. Section IV presents the design and the implementation details of AMAN. In Section V, we discuss limitations of AMAN and potential solutions to address arbitrary mobility and same location attacks. Section VI gives the experimental setup and results. Section II presents the related work and Section VII presents the concluding remarks. In Appendix A, we perform thorough comparative evaluation of $NRTT$-based authentication factor against state-of-the-art authentication factors.

## II. Related Work

In [8], a packet delay-based scheme is proposed to detect man-in-the-middle (MitM) attacks. The delay is calculated using TCP packet headers with the assumption that the delay increases in the presence of MitM attacker. However, packet delay can easily be manipulated by, for example, pinging the network service.

Recently, Gmail has launched a service that enables its users to detect suspicious account login activities based on their IP information [13]. A suspicious attempt is detected by matching the relevant IP address(es) to a broad geographical location(s). However, IP-based verification (fixing a range of IPs) can be easily bypassed via (1) proxy-server; (2) VPN; (3) IP-hijacking. Many web clients are behind proxies (or VPN). The client and the proxy may be far apart. For example, the AOL network, which has a centralized cluster of proxies at one location (Virginia) for serving client hosts located all across the U.S. [14]. 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 [15] [16]. Furthermore, IP address 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) [17]; (2) complex and various IP address configuration policies by different ISPs. Therefore, IP address based authentication is more suitable for LAN other than general web service authentication.

Universal two factor authentication (UTF) [18] utilizes public key cryptography to enable secure authentication, however, it has serious usability issues and suffers from the same public key cryptography issues including key revocation and trusted third party requirement. It has also been shown that UTF is susceptible to man-in-the-middle-script-in-the-browser attack [19].

The work in [1] presents a single profile system using $NRTT$ to strengthen web service authentication. However, it fails to efficiently address network instabilities and sufferers from very low entropy, which makes it very susceptible to throttling attacks. On the other hand, our proposed technique successfully addresses these challenges and is supported by detailed benchmark evaluations against state-of-the-art authentication factors.

## III. Background

In this section, we present the background knowledge of (1) the $NRTT$ Gaussian approximation which is the basis of the decision algorithm and the theoretical analysis; (2) the $NRTT$ profiling sample size based on Gaussian approximation; and (3) the network instabilities which affect the usability of $NRTT$ as an authentication factor.

### A. Gaussian Approximation

$NRTT$-based authentication is motivated by the results presented in [6], which show that network communications latency approximately follows a Gaussian distribution. This observation is validated by experiments that measure network communications latency among 130 PlanetLab nodes [20].

We have also conducted extensive and wider set of similar experiments using GENI nodes, campus and residential users both with wire-line and wireless connections. Our results validate the results in [6] and further support the observations about the Gaussian approximation of network communications latency. Figure 1 shows examples of $NRTT$ distributions and the corresponding Gaussian approximations for three different locations.

Though fine-grained mathematical model (e.g., Rayleigh distribution) may provide a better approximation, it will introduce much more complicated theoretical analysis with marginal or no additional benefit for the real world implementation of the proposed algorithms.More importantly, the empirical results of many of the existing research on round trip network communications latency ([6] [7] [8]) show that Gaussian distribution is an adequate approximation for $NRTT$. These conclusions are further supported by our experiments and mathematical analysis based on the Gaussian approximation of $NRTT$.

### B. Profiling Sample Size

$NRTT$ profile is built by exchanging a number of small packets, dubbed profiling signals, with the user. The number of profiling signals is known as the profiling sample size. The larger the number of profiling signals, the more accurate the profile will be. However, the larger the number of profiling signals, the higher the bandwidth overhead and the longer the login latency. Therefore, it is critical to find a profiling sample size that leads to an acceptable trade-off between profile accuracy, bandwidth overhead, and the average time it takes a user to login.

TABLE I: List of all the acronyms

| | |
|---|---|
| $\mu$ | Mean of the reference profile |
| $\sigma$ | Standard deviation of the reference profile |
| $1-\alpha_i$ | Confidence level |
| $p$ | Confidence interval (error tolerance) |
| $x$ | Mean of the real-time profile |
| $y$ | Standard deviation of the real-time profile |
| $N$ | Profiling Sample size |

To have an initial estimate of the profiling sample size, we use the Gaussian approximation of $NRTT$ distribution. Assume a population with Gaussian distribution that has standard deviation $\sigma$ and mean $\mu$. The goal is to find the minimum sample size, $N$, that produces a mean, $x$, within a certain error margin (aka, error tolerance), $p$, with a certain confidence level, $1-\alpha$. The error margin $p$, is the maximum allowed distance between $\mu$ and $x$. The confidence level represents how confident we are that the measured mean ($x$) falls within the confidence interval. For Gaussian distributions, it has been shown ([21]) that the minimum sample size $N$ can be calculated as:

$$N \geq (Z_{1-\alpha}/p)^2\sigma^2 \tag{1}$$

Where $Z$ is the critical value for the normal distribution. In other words, for a sample size of $N$, we are 1-$\alpha$ confident that the measured mean ($x$) will fall in the range of:

$$\mu - p \leq x \leq \mu + p \tag{2}$$

### C. Network Instabilities

Network communications latency may vary due to different reasons including congestion, queuing delays, server load, contention ratio in local network, and ISP throttling or traffic shaping operations. Therefore, the naive measurement of $NRTT$ may result in poor performance if such network instabilities are not carefully handled. Our $NRTT$-based authentication factor is designed with such instabilities in mind and hence it incorporates the necessary measures to alleviate the impact of such instabilities. In the following, we classify network instabilities into three categories, namely, instantaneous instabilities, long-term instabilities, and routing instabilities, and in Section IV, we show how to mitigate the impact of each category on the measurement of $NRTT$.

**Instantaneous instabilities** are instabilities which lead to transient changes in communications latency and hence, it only affects a few of the profiling signals. This type of instability is the most common one and can be addressed through outlier filtering (**Algorithm 2**) as detailed in Section IV-C.

**Long-term instabilities** are instabilities that stay long enough to affect all or most of the of profiling signals, however, they are not permanent. Such instabilities are mainly caused by low bandwidth, congestion, or variable traffic volume at the location of the user (i.e., the local network segment connecting the user to the network backbone). For example, if a user has a low bandwidth Internet, she will experience longer communications latency while her roommate is watching an HD movie on-line. We address this category of network

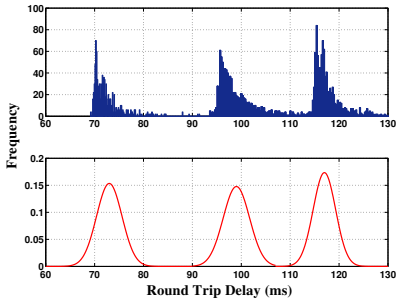


Fig. 1: Sample of $NRTTs$ and their Gaussian approximations

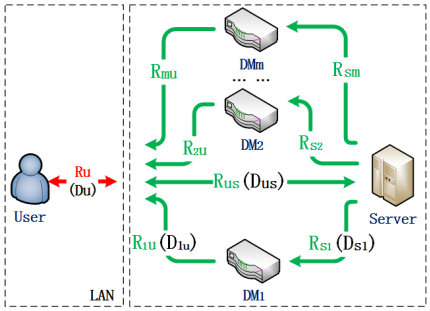


Fig. 2: The Ecosystem of AMAN

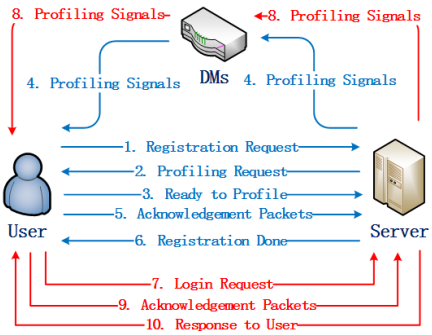


Fig. 3: AMAN Authentication Flowchart

instabilities by establishing multiple profiles, as detailed in Section IV-E.

**Routing instabilities** are instabilities that result in permanent changes in network communications latency due to, for example, permanent network routing changes. We introduced a dynamic temporal profiling technique (re-profiling the user every week based on the most recent login instances) to address long term routing instabilities. The main motivation of this profiling technique is to capture and handle changes in $NRTT$ over extended time periods. Changes of the $NRTT$ over long time periods are mainly due to routing changes in the Internet. We have designed the temporal profiling algorithm to handle such fairly uncommon cases. The design of our temporal profiling algorithm (such as the selection to profile every week) is guided by the long line studies of the routing behavior over the Internet. For example, earlier studies ([22] [23] [24] [25]) showed that most of the important IP prefixes have stable routes and that instabilities only exist in a small portion of the global Internet. A recent study [26], which was 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 and the rate of change in routing decisions is stable over time, despite of the overall growth in the size of the Internet. Furthermore, it also reveals that only a small fraction of ASes are responsible for the vast majority of routing changes. Additionally, the 2013 experimental data of [6] indicates that only 2 out of 150 PlanetLab nodes showed 1-hop change while the others remain unchanged and the latency variance due to the 1-hop change is negligible. Therefore, routing instabilities are not common and can be addressed, like other unexpected events (e.g., forced traffic-reroute, login during DoS attack), using the backup failure techniques discussed in Section V.

## IV. Design and Implementation

In this section we demonstrate the design and the implementation of secure and reliable $NRTT$-based authentication factor through a use case of two-factor authentication scheme, dubbed AMAN. AMAN uses traditional passwords as the first authentication factor and $NRTT$ as the second authentication factor.

### A. Assumptions

In the context of this work, we assume a powerful attack model, in which attackers already compromised traditional passwords of users. However, it is intuitive to conclude that AMAN, solely by itself, does not defend against perpetrators who both compromise legitimate user credentials and have access to her profiled location. Such attacks can be thwarted by augmenting AMAN with additional authentication factors, such as browser fingerprinting, as explained in Section V. Additionally, Denial of Service (DoS) and remote access attacks (e.g., Rootkits and RATs on end user devices) are out of the scope of this work. We assume that the registration phase, when reference profiles are built, is secure, which is a reasonable assumption as it is required by all profiling schemes. We also assume that delay masks are secure and connected to network segments that are not accessible by attackers.

### B. The Authentication Protocol

As depicted in Figure 2, the ecosystem of AMAN comprises three entities, users, web server (aka, authenticator or verifier), and delay masks. To demonstrate the authentication protocol in AMAN, we use the typical web-page login scenario depicted in Figure 3. The authentication protocol has two-phases, registration phase and login phase. The registration phase is a one-time process that initializes reference profiles, while the login phase is initiated with every login attempt to build real-time profiles. A profile is represented by the mean and the standard deviation of a number ($N$) of $NRTTs$ measured between user and server. Based on the distance between real-time profiles and reference profiles, the decision algorithm at the authenticator grants or denies access. To enhance the accuracy of reference profiles, they are updated with every successful login, that is, the new reference profile after a successful login is the combination of the current reference profile and the real-time profile of the last login. The new reference profile replaces the current one in server database while the real-time profile vanishes after the access decision is made. Using Figure 2, **Algorithm 1** presents the detailed steps of the registration phase and login phase, where each step in the algorithm maps to the corresponding step in Figure 3.

### C. Computing Profile Parameters

As shown in steps #4 and #5 of **Algorithm 1**, the server measures $NRTT$ statistics of its users by sending out a set of small packets (similar to ping messages), dubbed as profiling signals. In current prototype implementation, the profiling signals are a set of UDP/TCP packets with user login session information in the payload (e.g., session ID, index number, user's IP address, etc.) An estimate of the number of profiling signals ($N$) that is required to establish a sufficiently accurate profile is discussed in Section III-B. Before sending

profiling signal $S_i$ ($i \in [1, N]$), the server records its send time ($SndTime_i$). When the corresponding acknowledgment ($ack_i$, $i \in [1, N]$) is received, the server records the reception time ($RcvTime_i$). Then, the server computes the round trip time of each signal as $NRTT_i = RcvTime_i - SndTime_i$. After computing the ($N$) $NRTT$ values, the server calls **Algorithm 2**, which uses the scheme in [27] to first remove the outliers and then, it computes the mean and the standard deviation of the remaining $NRTT$ values. Filtering outliers aims at alleviating potential instantaneous network instabilities (Section III-C).

**Algorithm 1** AMAN Authentication Process

*Input*: Total number of DMs ($D$), number of profiles per user ($r \leq D$), number of selected profiles per user ($m \leq r$), number of profiling signals per profile ($N$)
*Output*: Grant or deny access

1: *Registration Request*: User sends request to server
2: *Profiling Request*: Server seeks the permission of the user
3: *Ready to Profile*:
- User accepts profiling request
- Server randomly picks $r$ out of the $D$ available DMs. Each of the selected ($r$) DMs will be used to generate a different profile for user

4: *Sending profiling signals*: Server,
- Sends $r \cdot N$ profiling signals interleaved among the $r$ DMs. Let $d_s = \{d_1, d_2, ..., d_r\}$ be the set of selected DMs and let $p_s = \{S_1, S_2, ..., S_{r \cdot N}\}$ be the set of profiling signals. Signal $S_i$ in $p_s$ is forwarded through DM $j$ in $d_s$, where $j = i \mod r + 1$.
- Records $SndTime_i$ of each $S_i$ ($i \in [1, r \cdot N]$)

5: *Acknowledgments*: For $i \in [1, r \cdot N]$,
- User sends $ack_i$ of $S_i$ directly to server
- Server records $RcvTime_i$ of $ack_i$
- Server computes $NRTT_i = RcvTime_i - SndTime_i$

*Compute profile parameters*: For $j \in [1, r]$, if all the ($N$) $NRTT$ values of $d_j$ are ready,
- Server calls **Algorithm 2** to compute mean and standard deviation for each of the $r$ profiles

6: *Registration done*: Server,
- Chooses the $m$ most stable profiles out of the computed $r$ ones (i.e. the $m$ profiles that has the smallest standard deviation values) as the set of reference profiles for the user. This step helps in alleviating routing instabilities.
- Server acknowledges user and returns

7: *Login Request*: User sends login request to server
8: *Profiling Signals*: Server repeats step #4 using $m$ instead of $r$
9: *Acknowledgments*: Repeat step #5 using $m$ instead of $r$
10: *Response to User*: Server sends the result of calling **Algorithm 4**

### D. Delay Mask

The naive measurement of $NRTT$ allows attackers to easily figure it out. For example, the attacker can simply ping the server from the location of the user. Ping packets provide excellent estimation of the $NRTT$ 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 she is in a location with $NRTT$ less than that of the user, she can easily mimic the profile of the user; she simply adds appropriate delay before acknowledging the profiling signals. To address this important security concern, AMAN introduces a special one-way forwarding device, delay mask, in the route between the server and its clients. The DM is deployed and controlled by the server and is set to only relay profiling signals from the server to its users.

The main objective of the DM is to prevent any entity, except the authenticator, from being able to estimate $NRTT$. The DM achieves this by creating new path-segments in the round trip path between the server and its users. Therefore, the communications time over the newly created path-segments cannot be measured by outsiders. In Figure 2, consider DM1 for instance, the $NRTT$ can be computed as:

$$NRTT = D_{s1} + D_{1u} + 2 \cdot D_u + D_{us}$$

where $(D_u + D_{us})$ is the delay over the path-segments from the user to the server ($R_u$ and $R_{us}$), and $D_{1u}$ and $D_{s1}$ are the delays over the path-segments from DM1 to the user ($R_{1u}$) and from the server to DM1 ($R_{s1}$), respectively. the attacker may be able to estimate $(D_u + D_{us})$ by pinging the server from the location of the user, however, it is not possible for her to figure out $D_{1u}$ and $D_{s1}$ due to the stealthy nature of DM and its one-way communication architecture. In other words, round trip cycles can neither be established on the path-segment between the user and the DM nor on the path-segment between the DM and the server. Additionally, hiding the location of the DM (by hiding its IP) prevents estimates of the delays to the DM through measurement of the delay to close by entities. Therefore, the server is the only entity of AMAN that can measure profile parameters. We note here that legitimate users do not learn anything about their profiles or the profiles of other users. This not only makes $NRTT$ transparent to the users as they do not need to memorize or remember anything, but also prevents compromised users from breaching the security of AMAN.

Finally, we note that even in the worst case scenario in which $NRTT$ is disclosed, the attacker can impersonate the user (assuming her password is already compromised) only from locations that have similar or lower $NRTT$ values compared to that of the legitimate location, which reduces attack service and makes it harder. Additionally, as we see in the next section, the deployment of multiple DMs adds to the complexity and sophistication of such attacks and makes it highly unlikely because the attack can only succeed from locations that have lower $NRTT$ values for all the used DMs; which can be made difficult to achieve by careful deployments of the DMs.

### E. Multiple Profiles

Delay mask represents a novel idea in the design of AMAN because it makes $NRTT$ measurement robust and extremely hard to manipulate. However, AMAN with single DM suffers from two main limitations in **the case of password compromise**. The first is its vulnerability to un-throttled guessing due to the low entropy ($E$) in $NRTT$ ($E \approx 10$ bits [1], the

detailed mathematical analysis of $NRTT$ entropy is omitted for the sake of space). The second is the impact of network instabilities on the usability of AMAN due to the potential increase in the number of legitimate login failures. To address these limitations, AMAN deploys multiple DMs in different network locations. Multiple DMs are used to create multiple different profiles per user. **Algorithm 1** shows how AMAN generates multiple profiles using multiple DMs.

In the following, we demonstrate how multiple profiles can deter un-throttled guessing and alleviate network instabilities, then we present the authentication decision algorithm.

*1) Defense against un-throttled guessing::* In general, un-throttled guessing is a brute force attack in which the attacker is allowed to try all possible credential combinations until she hits the right one. However, it is extremely hard for the attacker to try all possible combinations in AMAN due to physical limitations. Attacker can only simulate latencies that are higher than her own by appropriately delaying acknowledgments of profiling signals, that is, there is no possible way for an attacker to impersonate a user whose $NRTT$ is lower than her own.

Even when passwords are compromised, multiple profiles help in defending against un-throttled guessing by both expanding authentication sample space (i.e., the collection of all possible credential combinations) and by making guessing extremely hard (if not impossible) to perform. It is fairly easy to show that multiple profiles considerably expand single-profile sample space. Assume that the entropy of the single-profile sample space is $E$, then the entropy of the $m$-profile sample space is simply $m \cdot E$, because the profiles are independent.

More importantly, multiple profiling further reduces the set of possible users that the attacker can possibly impersonate. For an attacker to impersonate a user, her $NRTT_i$ through $DM_i$ $(i \in [1, m])$ has to be faster than the corresponding $NRTT_i$ (i.e., through the same DM) of the user. If *any* of the $NRTT_i$ values of the user is faster than the corresponding one of the attacker, the attack definitely fails. The larger the number of DMs ($m$), the harder the attack can be performed. In fact, by carefully positioning the DMs, the attack can be made extremely hard to succeed.

**Algorithm 2** Compute Profile Parameters

*Input:* Set $R = \{NRTT_i, i \in [1, 2, ..., N]\}$; *Output:* $mean$ and $standard\ deviation$

**procedure** FILTER OUTLIERS
  Computes the median value of $R$: $M = Median(R)$
  Computes median absolute deviation (MAD) of $R$: $MAD = b * Median(abs(NRTT_i - M))$; where $b = 1.5$ for normal distribution
  Remove $NRTT_i > M + \tau \times MAD$ from $R$; where $\tau = 2$ (moderately conservative)
**end procedure**
Return $mean$ and $standard\ deviation$ of $R$

*2) Alleviating long-term instabilities::* Network latency comprises delay in local network and backbone delay. The work in [28] shows that traffic congestion is the main cause of local network delay, and it only marginally affects backbone delay. It also shows that the main contributing factor of the backbone delay is the speed of light where the delay jitter is extremely low [28]. These results lead to the conclusion that backbone network is much more stable than local network. The empirical results in [6] and our experiments also support this conclusion, that is, the main contributor of long-term network instabilities is traffic congestion in local networks. We design here a novel algorithm based on multiple profiling to mitigate the impact of such instabilities on $NRTT$ measurements.

To see that, consider the DMs depicted in Figure 2, which are used to establish different user profiles. All the profiles share the same local network segment ($R_u$) but have different and more stable backbone routes ($R_{s1}+R_{1u}$, $\cdots$, $R_{sm}+R_{mu}$) [28]. Therefore, congestion in the local network ($R_u$) will introduce similar noise in all the real-time profiles. To filter out such noise, AMAN uses **Algorithm 3**. AMAN first measures the difference between real-time mean ($x$) and reference mean ($\mu$) of each profile as $\triangle T_i = x_i - \mu_i$, where $i \in [1, m]$. Then, AMAN verifies that (i) either all the $\triangle T_i$ values are greater than $p$ or all the $\triangle T_i$ values are less than $-p$, where $p$ is the error tolerance defined by Equation (1) in Section III-B, and (ii) the variance of the $\triangle T_i$ values is less than $q = 0.5$, where $q$ is an experimentally predefined value. If these two conditions are true, then it is highly likely that the noise is caused by local congestion. In this case, AMAN simply subtracts the average noise value ($\triangle T = mean\{\triangle T_i\}$) from the mean of each real-time profile before applying the authentication decision algorithm (Section IV-E3).

**Algorithm 3** Filter out Long-term Instability

*Input:* real time means $x_j$; reference means $\mu_j$, $p$, $q$
*Output:* shared increment $\triangle T$

**procedure** CALCULATE-INCREMENTS
  Initialization: $\triangle T_0 = 0$
  **for** $j \in [1, 2, ..., m]$ **do**
    $\triangle T_j = x_j - \mu_j$
    **if** $\triangle T_j \cdot \triangle T_{j-1} < 0$ (opposite trend) **then**
      **return** $\triangle T = 0$ ;
    **end if**
    **if** $-p < \triangle T_j < p$ (within the error tolerance of the Gaussian distribution) **then**
      **return** $\triangle T = 0$ ;
    **end if**
  **end for**
  $std$ = standard deviation$\{\triangle T_j\}$
  **if** $std > q$ (increments vary a lot) **then**
    **return** $\triangle T = 0$;
  **end if**
  $\triangle T$ = mean$\{\triangle T_j\}$
  **return** $\triangle T$
**end procedure**

*3) The access decision algorithm::* The access decision algorithm (**Algorithm 4**) presents the logic by which AMAN grants or denies access to end users based on the real-time profiles and the stored reference profiles. As explained in

Section IV-B, the server keeps the $m$ most stable profiles as the reference profiles for each user. With each login attempt, the server builds $m$ corresponding real-time profiles. After computing the mean ($\mu_i$) and the standard deviation ($\sigma_i$) of each of the $m$ real time profiles (Section IV-C), AMAN uses the Gaussian PDF algorithm [29] to compute the distance ($score_{PDF}$) between the reference profiles and the the real-time profiles:

$$score_{PDF} = \frac{1}{m}\sum_{i=1}^{i=m} e^{-\frac{(x_i-\mu)^2}{2\cdot\sigma^2}}$$

Access is granted if $score_{PDF} \geq Threshold_{PDF}$, otherwise access is denied. The threshold is experimentally selected to match the desirable trade-off between false positives and false negatives as detailed in Section VI.

To further refine and tighten the access decision, AMAN can use out-of-band channels such as SMS or email to request supporting evidence in doubtful or borderline situations. In this case, the decision will be either clear accept, clear deny, or supporting evidence is required. The supporting evidence could be a random number delivered to the user through an out-of-band channel. This enhancement serves multiple purposes: (i) it can alleviate legitimate login failures that may occur due to unexpected events on the Internet such as network traffic re-route, (ii) it can support arbitrarily mobile clients, (iii) it can be used as a backup channel to recover from long-term login failures such as exceeding the maximum number of login retries, and (iv) it can be used to establish new spacial profiles for the user in new login locations.

**Algorithm 4** Authentication Decision Algorithm

*Input:* real time parameters $(x_j, y_j)$
reference parameters $(\mu_j, \sigma_j)$
*Output:* authentication decision: `True` or `False`

**procedure** COMPARE-PROFILES
  $\triangle T$ = call **Algorithm 3**
  $score_{PDF} = 0$
  **for** $j \in [1, 2, ..., m]$ **do**
    $x_j = x_j - \triangle T$
    $score_{PDF} += e^{-\frac{(x_j-\mu)^2}{2\cdot\sigma_j^2}}$
  **end for**
  $score_{PDF} = score_{PDF}/m$
  **if** $score_{PDF} \geq threshold$ **then**
    **return** `True` ← Grant access
  **else**
    **return** `False` ← Deny access
  **end if**
**end procedure**

## V. DISCUSSION

The cyber security threat landscape is complex and continuously evolving, and hence, no security mechanism is foolproof. In this section, we discuss the limitations of the proposed scheme and provide corresponding solutions, including: (1) the potential integration of AMAN with other authentication factors to support arbitrarily mobile clients; (2) defense against sophisticated attacks, in which attackers both compromise legitimate credentials and have access to the legitimate login locations; and (3) defense against low rate DDoS attack; and (4) considerations of the deployment for the delay masks.

### A. Mobility and login failures

Mobile clients can be broadly classified into: (i) ***Low mobile clients*** who frequently login from a number of locations such as home, office, and library. In this case, AMAN simply creates a separate profile for each location. A user will be granted access if her real-time login profile matches any of the stored profiles. (ii) ***Arbitrarily mobile clients*** who may login from any arbitrary location. In applications that support such mobile clients (also in case of login failure), AMAN could use other authentication factors such as browser fingerprinting or random strings delivered through out-of-band-channels such as SMS and emails and then apply the enhanced decision algorithm as detailed in Section IV-E3. However, we note here that if other factors are used to enable arbitrary mobility, mobile users do not benefit from the added security features provided by $NRTT$ such as resiliency to Phishing and MitM.

### B. Sophisticated attackers

As clarification, a sophisticated attacker, in our attack model, is that an attacker who compromised the password, knows the login location of the user, and has physical access to the login location of the user. We acknowledge that such attacker may succeed in impersonating the user, however, we would like to highlight the following facts: (1) We do not and can not claim that our authentication mechanism can be used anonymously for any generic application, but rather we have clearly stated that our $NRTT$ based authentication mechanism could greatly enhance the security of authentication in certain applications such as location-based authentication applications. Similar to most of the security mechanisms, we acknowledge that persistent and targeted attacks (attackers attempt to detect and exploit individual details of users) are very challenging and hard to block [30]. However, we note that, most of the authentication-based cyber attacks (e.g., stealing username/password by compromising the database of the web server) do not incorporate the users' location information and simply try to use these credentials as soon as possible from arbitrary locations, before being revoked. Therefore, we are confident that even though our authentication mechanisms is not completely fool-proof, it decreases the probability of attack success by greatly reducing the attack surface from anywhere in the world to only the login location of the user. (2) The "login location" as defined in our work does not mean the geographical nearby places but the same login network. For example, $NRTT$ will vary substantially if the user logs in through different networks (e.g., 4G, WiFi, etc.), even in the same room. (3) As mentioned in the paper, the login location issue can be addressed by augmenting our authentication mechanism with additional factors such as browser fingerprints and on-demand dynamic passwords.

### C. Low rate DDoS attacks

DDoS attacks are out of the scope of this work. However, instead of completely taking down the service, low rate DDoS attacks increase the network traffic (i.e., increase the

$NRTT$) and hence, it may hinder the intended functions of our authentication mechanism. This type of attacks may occur in different scenarios and hence can be alleviated according to each specific scenario: (1) Low rate DDoS attack in the server or the local network of the user is implicitly addressed and is already alleviated by the proposed shared increment removal algorithm utilizing the deployment of multiple DMs. This is simply because Low rate DDoS attacks in this scenario cause similar delays for all the real time profiles and hence can be easily filtered out; (2) Low rate DDoS attack against one of the DMs is alleviated by the design of multiple profiles and the use of the adaptive decision algorithm. For example, for a design with 3 DMs, even if one of the DM paths is under low rate DDoS attack, the overall output of the decision algorithm wouldn't be affected much because it is an averaging of all the three profiles. In addition, the adaptive decision algorithm will warn the server if one of the profiles goes beyond a certain threshold while the others remain at normal level; (3) The probability of low rate DDoS attack in multiple DM paths with different rates is very low, however, it still can be addressed by randomising the DMs per session per user.

### D. Considerations for the deployment of the DMs

Recall that the DM is assumed to be connected to a separate dedicated secure network segment such that attackers (and also legitimate users) do not know its IP and can neither connect from the location of the DM nor can they compromise it. One way of hiding the IP of the DM is by spoofing the IP address of the server. Before forwarding the profiling signal, the DM sets the source IP field of the packet to the IP address of the server instead of its own IP. In addition to hiding the existence of the DM, this also makes AMAN transparent to end users. However, controlled spoofing of the IP address of the server by the DM is very challenging because it requires collaboration with the ISP hosting the DM to avoid dropping of the spoofed packets.

In addition, Software Defined Network (SDN) provides the possibility of a lightweight solution due to its centralized control plane design. The ISP controller could easily modify the forwarding policy globally and route the packets arbitrarily.

Furthermore, the deployment cost of multiple DMs could be reduced by utilizing existing infrastructure of the web service. For example, Google has many offices, repair branches, and data centers where DMs could be deployed.

## VI. Experiments

We conduct a thorough set of experiments to evaluate the usability (in terms of false negative rate, $FN$) and the security (in terms of false positive rate, $FP$) trade-offs of AMAN. We also study the impact of network instabilities on AMAN and assess its performance overhead. Specifically, we measure the following five metrics: (i) the false negative rate ($FN$), which is the probability that a legitimate login attempt fails, (ii) the false positive rate ($FP$), which is the probability that a perpetrator who *possesses the password* of a legitimate user successfully authenticates on her behalf, (iii) the login latency overhead ($LLO$), which is the average extra time it takes a user to successfully authenticate in AMAN compared to legacy one-factor password authentication, (iv) the storage overhead ($SO$), which is the extra storage space required per user in AMAN, and (v) the bandwidth overhead ($BO$), which is the extra network bandwidth incurred per login instance in AMAN. For all these metrics the lower the value is the better. We measure the variations in these metrics by varying four parameters: (i) the number of profiles per user ($m$), (ii) the decision threshold ($DT$), (iii) the number of profiling signals ($N$), and (iv) the maximum number of login retries ($LR$). In all the following experiments, unless otherwise stated, we use $N = 45$, $LR = 2$, $m = 2$, $DT = 0.85 : 0.005 : 0.92$, and Gaussian PDF in the access decision algorithm. We re-emphasize that the $FP$ values presented here are for powerful attackers, that is, attackers who already know passwords of legitimate users they try to impersonate.

### A. Experimental Setup

We build a test-bed that implements the three entities of AMAN with the following configurations:

- **Users**: The user population consists of 10 Amazon EC2 instances, 130 PlanetLab nodes, 25 GENI nodes, and 63 residential WiFi users randomly selected from different places in USA and Canada.
- **Authenticators**: The authenticator runs Apache HTTP Server version 2.4 and a web service implemented in HTML and PHP. The authentication credentials (username, password, profile mean, profile standard deviation, decision threshold, etc.) are stored in a $MySQL$ database.
- **Delay Mask**: We configure 3 PCs to work as the DMs. A $C$ program is implemented on top of the MAC layer to relay the profiling signals from authenticators to users. The DMs are deployed in Oregon (USA), Germany, and Qatar.

### B. Experimental results

We first present the usability and security trade-offs, then we present the impact of traffic conditions on the access decision algorithm, and finally, we analyze the storage and performance overhead of AMAN.

*1) Usability and Security trade-offs::* In this section, we study the trade-offs between usability ($FN$) and security ($FP$) properties of AMAN under different system parameters.

To measure $FN$, for each test instance (e.g., a test with $P = 2$, $N = 45$, $DT = 0.92$, $LR = 2$), each of the 25 GENI nodes and 63 residential WiFi users attempted to login 300 times. The data were collected within one day period for GENI nodes (i.e., wireline connections). For residential users (i.e., WiFi connections), the data were collected at random time within one month period). To measure $FP$, each of the 25 GENI nodes tries to login 300 times within one day period using the username and password of the other 217 clients (i.e., the other 24 GENI nodes, 63 residential users, and 130 PlanetLab nodes), a total of 1,627,500 (25*217*300) impersonation attempts are launched for each test instance. In all the following experiments, we vary the $DT$ from 0.85 to 0.92 with 0.005 step and measure $FP$ and $FN$ values for each threshold point.

Note that, the experiments are indeed conducted over long time periods, not within a few days. The experimental data

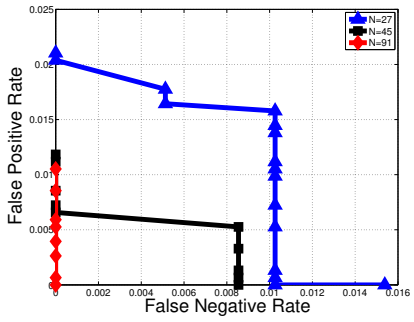


Fig. 4: ROC curves of $FP$-$FN$; Varying the # of profiling signals $N$; $P = 2$, $LR = 2$, $DT = 0.85 : 0.005 : 0.95$

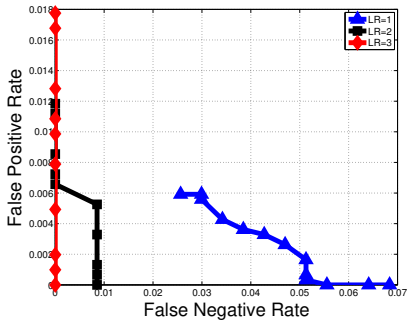


Fig. 5: ROC curves of $FP$-$FN$; Varying the maximum # of login retries $LR$; $N = 45$. $P = 2$. $DT = 0.85 : 0.005 : 0.95$

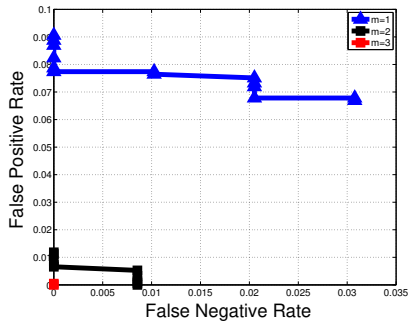


Fig. 6: ROC curves of $FP$-$FN$; Varying the # of profiles per user $P$; $N = 45$, $LR = 2$, $DT = 0.85 : 0.005 : 0.95$

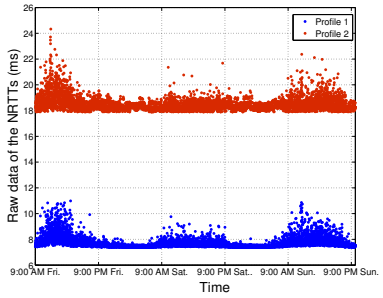


Fig. 7: Raw data of $NRTTs$ for two different profiles taken every 20 minutes for 60 hours

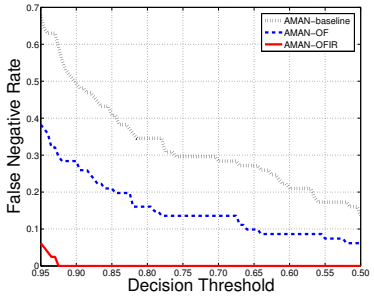


Fig. 8: The $FN$ curve with variable $DT$ for three different decision algorithms.

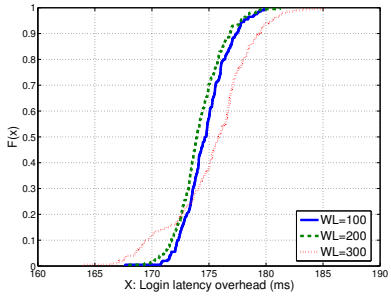


Fig. 9: The CDF of login latency overhead with variable $WL$

for the 63 residential users was collected and tested over 1 month period. In addition, our testbed has been running for about 1 year. Even though the tests are not performed continuously, they still demonstrate good performance for each individual test, thanks to the proposed dynamic temporal profiling algorithm described in section III-C.

- ***Varying the number of profiling signals*** $(N)$: Figure 4 shows the ROC (receiver operator characteristics) curves of $FN$ and $FP$ with variable $DT$ for $N$ equals 27, 45 and 91. The figure emphasizes the analysis results in Section III-B as it clearly shows that both $FP$ and $FN$ improve (lower values) as $N$ increases. For example, AMAN can achieve $FP = 0$ and $FN \approx 0.85\%$ for $N = 45$. The figure also shows that with $N = 91$, AMAN achieves $FP = 0$ and $FN = 0$.
- ***Varying the maximum number of login retries*** $(LR)$: Figure 5 shows the ROC curves of $FN$ and $FP$ with variable $DT$ for $LR$ values of 1, 2, and 3. The figure clearly shows that $LR$ has big positive impact on $FN$. When the user has a second chance to login after the first failed one, $FN$ can be significantly reduced. On the other hand, $LR$ has a relatively very small negative impact on $FP$. Increasing $LR$, only slightly increases $FP$ because attacker almost gains nothing when given another login chance, even if the new login chance is performed from a new location, thanks to the large authentication sample space of AMAN. The figure shows that for $LR = 3$, AMAN can achieve both $FN = 0$ and $FP = 0$.
- ***Varying the number of profiles per user*** $(m)$: Figure 6 shows the ROC curve of the $FN$ and $FP$ with variable $DT$ for $m$ values of 1, 2 and 3. The figure clearly shows that the number of profiles per user has significant positive impact on $FP$. The 1-profile $FP$ value is much larger than that of the 2-profile and the 3-profile cases. This is intuitive because multiple profiles both increase the authentication sample space, and considerably decrease impersonation by physically limiting the number of locations from which impersonation could be lunched (Section IV-E1). Similarly, the larger the number of profiles, the lower the $FN$. Recall that multiple profiles help to filter out network instabilities and hence decreases $FN$. The figure shows that with $m = 3$, AMAN can achieve $FP = 0$ and $FN = 0$.

*2) Impact of network instabilities:* For this experiment, we set a WiFi user connected to the university campus to login every 20 minutes for 60 hours, from 9AM Friday through 9PM Sunday. Figure 7 shows all the $NRTT$ values measured for two different profiles during the test period, a total of 8100 (60*3*45) $NRTTs$ per profile. The figure clearly shows the variations in $NRTT$ values over different time periods. To evaluate the impact of such variations on the $FN$ rate and to assess the capabilities of AMAN to cope with them, we use the data in Figure 7 to compute the $FN$ rate over all the login attempts. Recall that AMAN uses outlier filtering (**Algorithm 2**) and shared increment removal (**Algorithm 3**) to alleviate instantaneous and long-term instabilities, respectively. Figure 8 shows $FN$ as a function of $DT$ for baseline AMAN (no filtering), AMAN with outlier filtering alone (AMAN-OF), and AMAN with both outlier filtering and shared increment removal (AMAN-OFIR) with $p = 0.5ms$ and $q = 0.5ms$. The figure shows that network instabilities can badly hurt the usability of AMAN as evidenced by the 10% $FN$ rate, which is the best $FN$ rate that baseline AMAN can achieve. On the other hand, the figure shows the effectiveness of AMAN in

coping with network instabilities as evidenced by the 0 $FN$ rate achieved by AMAN-OFIR, almost irrespective of the $DT$ value used. The flexibility in $DT$ is very important as it allows AMAN-OFIR to achieve lower $FP$ rate as well, because the higher the $DT$, the better the $FP$.

### C. Performance Overhead

*1) Login Latency Overhead ($LLO$)::* In this experiment, we evaluate the login latency overhead ($LLO$) per user in AMAN compared to that in legacy password authentication. We use the 10 Amazon instances in our test-bed to generate variable server workload ($WL$), which is measured by the number of login requests per second. Then, we measure the $LLO$ of 12,420 logins under server workloads of 100, 200 and 300 login requests per second. Figure 9 shows the empirical cumulative distribution functions (CDFs) of $LLO$ under each $WL$. The figure shows that more than 99% of the login instances have $LLO$ less than 0.185 seconds, which is unnoticeable to humans.

*2) Bandwidth Overhead::* The bandwidth overhead ($BO$) is caused by the profiling signals and the acknowledgments of every login attempt. Each of the $m$ profiles requires $N$ profiling signals and $N$ acknowledgments per login attempt. The size of the profiling signal/acknowledgment is about 50 bytes including all the headers. Therefore, $BO = 100 \cdot m \cdot N$ bytes/login instance. Based on our experiments, $m = 3$, $N = 45$ provides excellent trade-off between $FP$ and $FN$, and hence, $BO = 100 \cdot 3 \cdot 45 = 13500 \approx 13K$ bytes. This is a negligible overhead given the fact that login bandwidth consumed by most of the popular websites (e.g., Chase.com, Facebook.com, Amazon.com) ranges between 10.98K and 125.47K bytes, according to a study that we have conducted on 12 popular web services (omitted for the sake of space).

## VII. Conclusions

In this paper, we proposed a novel, secure and usable web authentication factor based on Network Round Trip Time ($NRTT$) that strengthens the security of web service authentication by offering robust defenses against password compromise. We introduced a novel network component, delay mask, which turns $NRTT$ into a secure and robust authentication factor. More importantly, we designed and implemented various algorithms, with the help of multiple DM deployments, to alleviate network instabilities and expand authentication sample space of $NRTT$. The benchmark comparative results (Appendix A) showed that $NRTT$ has superior security, usability, and deploy-ability properties among state-of-the-art authentication factors.

We designed, implemented and deployed a prototype for a use case of two-factor authentication (AMAN) with legacy passwords as first factor and $NRTT$ as a second factor. The experimental results showed that AMAN can achieve false positive and false negative rates as low as 0, while maintaining the login latency overhead below 185 ms.

In the future, we plan to make AMAN completely end user agnostic by replacing passwords with mechanisms that can automatically collect unique user characteristics. Such techniques may utilize biometric tools such as typing behavior, screen resolution, device signatures and others.

## References


[1] I. Khalil, Z. Dou, and A. Khreishah, "Your credentials are compromised, do not panic: You can be well protected," in *11th ACM AsiaCCS*, 2016.

[2] P.-H. Kamp, P. Godefroid, M. Levin, D. Molnar, P. McKenzie, R. Stapleton-Gray, B. Woodcock, and G. Neville-Neil, "Linkedin password leak: Salt their hide.," *ACM Queue*, vol. 10, no. 6, p. 20, 2012.

[3] I. M. Khalil, A. Khreishah, and M. Azeem, "Cloud computing security: a survey," *Computers*, vol. 3, no. 1, pp. 1–35, 2014.

[4] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth-factor authentication: somebody you know," in *13th ACM CCS*, 2006.

[5] Y. Zhang, Z. Chen, H. Xue, and T. Wei, "Fingerprints on mobile devices: Abusing and eaking," in *Black Hat Conference*, 2015.

[6] 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 *IEEE 7th International Conference on Cloud Computing, 2014*.

[7] M. Tahara, N. Tateishi, T. Oimatsu, and S. Majima, "A method to detect prefix hijacking by using ping tests," in *Challenges for Next Generation Network Operations and Service Management*, Springer, 2008.

[8] V. A. Vallivaara, M. Sailio, and K. Halunen, "Detecting man-in-the-middle attacks on non-mobile systems," in *CODASPY*, ACM, 2014.

[9] K. Benson, R. Dowsley, and H. Shacham, "Do you know where your cloud files are?," in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, pp. 73–82, ACM, 2011.

[10] J. A. Muir and P. C. V. Oorschot, "Internet geolocation: Evasion and counterevasion," *Acm computing surveys (csur)*, vol. 42, no. 1, p. 4, 2009.

[11] D. E. Denning and P. F. MacDoran, "Location-based authentication: Grounding cyberspace for better security," *Computer Fraud & Security*, vol. 1996, no. 2, pp. 12–16, 1996.

[12] Netcraft, "Anti-phishing extension: Netcraft," 2017. http://toolbar.netcraft.com/.

[13] Google, "Gmail: Detecting suspicious account activity," 2010. http://googleonlinesecurity.blogspot.com/2010/03/detecting-suspicious-account-activity.html.

[14] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for Internet hosts," in *ACM SIGCOMM Computer Communication Review*, ACM, 2001.

[15] P.-A. Vervier, O. Thonnard, and M. Dacier, "Mind your blocks: On the stealthiness of malicious bgp hijacks.," in *NDSS*, 2015.

[16] A. Gavrichenkov, "Breaking https with BGP hijacking," *Black Hat. Briefings*, 2015.

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

[18] S. Srinivas, D. Balfanz, E. Tiffany, and F. Alliance, "Universal 2nd factor (u2f) overview," *FIDO Alliance Proposed Standard*, 2013.

[19] N. Karapanos and S. Capkun, "On the effective prevention of tls man-in-the-middle attacks in web applications," in *USENIX Security 14*, 2014.

[20] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: an overlay testbed for broad-coverage services," *ACM SIGCOMM Computer Communication Review*, 2003.

[21] NIST/SEMATECH, "e-Handbook of statistical methods," 2013. http://www.itl.nist.gov/div898/handbook/.

[22] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Transactions on Networking*, 1998.

[23] 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*, 2002.

[24] M. Lad, J. H. Park, T. Refice, and L. Zhang, "A study of Internet routing stability using link weight," tech. rep., 2008.

[25] A. Shaikh, A. Varma, L. Kalampoukas, and R. Dube, "Routing stability in congested networks: Experimentation and analysis," in *ACM SIGCOMM Computer Communication Review*, 2000.

[26] G. Comarela, G. Gürsun, and M. Crovella, "Studying interdomain routing over long timescales," in *Proceedings of the 2013 conference on Internet measurement conference*, pp. 227–234, ACM, 2013.

[27] 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," *JESP*, 2013.

[28] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. C. Diot, "Packet-level traffic measurements from the sprint ip backbone," *Network, IEEE*, 2003.

[29] P. S. Teh, A. B. J. Teoh, T. S. Ong, and H. F. Neo, "Statistical fusion approach on keystroke dynamics," in *Signal-Image Technologies and Internet-Based System, 2007. Third International IEEE Conference on*.

[30] S. Le Blond, A. Uritesc, C. Gilbert, Z. L. Chua, P. Saxena, and E. Kirda, "A look at targeted attacks through the lense of an ngo.," in *USENIX Security*, pp. 543–558, 2014.

[31] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *IEEE S&P*, 2012.
[32] L. Simson, "Email-based identification and authentication: An alternative to PKI?," 2003.
[33] S. Hallsteinsen, I. Jorstad, *et al.*, "Using the mobile phone as a security token for unified authentication," in *ICSNC 2007.*, IEEE.
[34] RSA, "RSA securid," 2010. http://www.emc.com/collateral/solution-overview/10695-sidtfa-sb.pdf.
[35] E. Grosse and M. Upadhyay, "Authentication at scale," *IEEE Security & Privacy*, vol. 11, no. 1, pp. 15–22, 2013.
[36] V. Coskun, K. Ok, and B. Ozdenizci, *Near Field Communication (NFC): From Theory to Practice*. John Wiley & Sons, 2011.
[37] A. Wiesmaier, M. Fischer, M. Lippert, and J. Buchmann, "Outflanking and securely using the pin/tan-system," *arXiv preprint cs/0410025*, 2004.
[38] A. Ross, J. Shah, and A. K. Jain, "From template to image: Reconstructing fingerprints from minutiae points," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2007.
[39] 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.
[40] M. Pusara and C. E. Brodley, "User re-authentication via mouse movements," in *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pp. 1–8, ACM, 2004.
[41] T. Morris, "Trusted platform module," in *Encyclopedia of Cryptography and Security*, Springer, 2011.
[42] I. Khalil, Z. Dou, and A. Khreishah, "Tpm-based authentication mechanism for apache hadoop," in *International Conference on Security and Privacy in Communication Systems*, pp. 105–122, Springer, 2014.
[43] Z. Dou, I. Khalil, A. Khreishah, and A. Al-Fuqaha, "Robust insider attacks countermeasure for hadoop: Design and implementation," *IEEE Systems Journal*, 2017.
[44] J. Brainard, A. Juels, R. L. Rivest, M. Szydlo, and M. Yung, "Fourth-factor authentication: somebody you know," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006.

## APPENDIX A
## COMPARATIVE EVALUATION

In this section, we perform thorough comparative evaluation of $NRTT$-based authentication factor against state-of-the-art authentication factors by employing a famous benchmarking framework for web authentication, which has been proposed by Joseph Bonneau et al. [31]. We compare $NRTT$ (represented by **H** in Table II) against: **A1**: Second Authentication Code (SAC) through email [32], **A2**: SAC through mobile SMS/voice [33], **B1**: disconnected hardware token (e.g., RSA SecurID [34]), **B2**: connected hardware token (e.g., smartcard-like USB token, NFC, etc. [35] [36]), **C**: paper token (e.g., PIN+TAN [37]), **D**: biometrics (e.g., fingerprint, iris, voice recognition, etc. [38]), keystroke [39], **E1**: non-keystroke device characteristics (e.g., mouse click pattern, [40]), **E2**: hardware/system signature (e.g., browser fingerprint, trusted platform module, etc. [41] [42] [43]), **F**: knowledge-based information (e.g., photo recognition of somebody you know [44]).

The benchmark encompasses twenty five properties grouped into three categories, namely, **usability**, **deploy-ability** and **security**. Each property has two-dimensional score $(VU)$. The first entry of the score $(V)$ indicates whether the authentication scheme offers the property $(V = Y)$, does not offer the property $(V = N)$, or partially offer the property $(V = A)$. The second entry of the score $(U)$ indicates whether this property offering is better than $(U = \text{" + "})$, similar to $(U = \text{" "}$, i.e., no symbol), or worse than $(U = \text{" − "})$ that of the legacy password scheme (the baseline). For example, if an authentication scheme offers a certain property better than the baseline, the score of this authentication scheme against that property will be $Y+$. Table II presents the comparison among factors. The results clearly show that $NRTT$ has the best combination of usability, deploy-ability and security proprieties. In the following, we explain the benchmark properties that may not be straightforward and drop the explanation of the intuitive ones for the sake of space.

### A. Security

*1) Resilient-to-Physical-Observation:* It evaluates the potential leakage of authentication credentials by physical observation of users during login. For $NRTT$-based authentication (AMAN), even legitimate users can not learn or compute their own profile parameters (Section IV-D). Therefore, it is completely resilient to physical observation. On the other hand, SAC and legacy password schemes are susceptible to physical observation. Biometric factors (e.g., fingerprint) based schemes are partially resilient to physical observation due to the potential capture of individual biometrics using special tactics, such as lifting fingerprints from the glass surface of scanners.

*2) Resilient-to-Targeted-Impersonation:* It evaluates the potential of targeted impersonation by capturing or simulating specific user authentication factors. Authentication relies on $NRTT$ which has been shown to be robust and unforgeable factor (Section IV-D), and hence, $NRTT$ is completely resilient to targeted impersonation. Other factors, take biometric factor - fingerprint for example, it is obviously susceptible due to fixed fingerprint values.

*3) Resilient-to-Internal-Observation:* It evaluates whether the attacker can capture credentials by intercepting the input of the user inside her device. As detailed in Section IV-B, $NRTT$-based authentication does not require any user input. Therefore, it is highly unlikely, even for the legitimate users, to learn or compute their own $NRTT$. On the other hand, SAC is susceptible to internal observation since attackers might be able to intercept authentication codes in SMS/email.

*4) Resilient-to-Leaks-from-Other-Verifiers:* It evaluates whether specific user credentials could be leaked across different verifiers. For $NRTT$, thanks to the DMs, profile parameters across different verifiers are independent, and hence, user profiles in one verifier are decoupled from her profiles in other verifiers. However, biometrics factors (e.g., fingerprint) are completely susceptible.

*5) Resilient-to-Phishing:* It evaluates whether man-in-the-middle (MitM) attackers can capture credentials. Active-Phishing describes the attack in which perpetrators use forged websites to capture authentication credentials (legacy password, SAC, fingerprints, etc.) and then, use them in real time to impersonate legitimate users. The concept of MitM does not apply in the case of $NRTT$ because its users do not send credentials that could be intercepted by attackers, instead credentials are measured at the server. Similar to legacy passwords, both SAC and biometric factors are obviously susceptible to MitM attackers.

*6) Resilient-to-Theft:* It evaluates the potential leakage of credentials through loss or theft of special authentication devices, such as authentication tokens. This is not applicable to $NRTT$ as it does not require any devices. On the other

TABLE II: Evaluation of $NRTT$ and other state-of-the-art second authentication factors.

| | | A1 | A2 | B1 | B2 | C | D | E1 | E2 | F | G | H |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Security** | Resilient-to-Physical-Observation | A+ | A+ | A+ | Y+ | N- | Y+ | N- | N- | Y+ | Y+ | Y+ |
| | Resilient-to-Targeted-Impersonation | A+ | A+ | Y+ | Y+ | Y+ | N- | Y | Y | A+ | N- | A+ |
| | Resilient-to-Throttled-Guessing | Y | Y | Y | Y | Y | Y+ | Y | Y | Y | Y | Y |
| | Resilient-to-Unthrottled-Guessing | Y | Y | Y | Y | Y | Y+ | Y | Y | Y | A- | Y |
| | Resilient-to-Internal-Observation | N | N | Y+ | N- | N- | N- | N- | N- | N- | Y | Y+ |
| | Resilient-to-Leaks-from-Other-Verifiers | Y+ | Y+ | Y+ | Y+ | Y+ | N- | N- | N- | N- | Y | Y+ |
| | Resilient-to-Phishing | N | N | N | N | N | N | N | N | N | N | Y+ |
| | Resilient-to-Theft | Y | N- | N- | N- | N- | N- | Y | Y | Y | Y | Y |
| | No-Trusted-Third-Party | N- | N- | N- | N- | Y | Y | Y | Y | Y | Y | N- |
| | Requiring-Explicit-Consent | Y | Y | Y | A- | Y | Y | Y | Y | Y | Y | Y |
| | Unlinkable | N- | N- | Y | Y | Y | N- | N- | N- | N- | A+ | Y+ |
| **Usability** | Memorywise-Effortless | A+ | A+ | A+ | Y+ | Y+ | Y+ | Y+ | Y+ | Y+ | N | Y+ |
| | Scalable-for-Users | Y+ | Y+ | N- | N- | Y+ | Y+ | Y+ | Y+ | Y+ | N- | Y+ |
| | Nothing-to-Carry | Y | N- | N- | N- | N- | Y+ | Y+ | Y+ | Y+ | Y+ | Y |
| | Physically-Effortless | N | N | N | A+ | N | N- | N | Y+ | Y+ | N- | Y+ |
| | Easy-to-Learn | Y | Y | A- | Y | A- | Y | Y+ | Y+ | Y+ | Y | Y |
| | Efficient-to-Use | N- | N- | A- | A- | A- | N | Y | Y | Y+ | N- | Y+ |
| | Infrequent-Errors | Y | Y | Y | Y | Y | A- | A- | N- | Y | A- | A- |
| | Easy-Recovery-from-Loss | Y | Y | N- | N- | N- | N- | N- | N- | N- | Y | Y |
| **Deployability** | Accessible | Y | A | A | A | A | N- | Y | Y | Y | Y | Y+ |
| | Negligible-Cost-per User | Y | N- | N- | N- | Y- | N- | Y | Y | A- | Y | Y |
| | Server-Compatible | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- | N- |
| | Browser-Compatible | Y | Y | Y | N- | Y | N- | Y | N- | N- | Y | A- |
| | Mature | Y | Y | Y | Y | Y | N- | N- | N- | N- | A- | N- |
| | Non-Proprietary | Y | Y | N- | A- | Y | A- | A- | Y | N- | Y | Y |

"-" = Worse than Legacy Password; no symbol = Same as Legacy Password; "+" = Better than Legacy Password; Y = offer the property; N = does not offer the property; A = partially offer the property.

hand, SAC through mobile SMS and all token-based factors are susceptible to theft-related attacks.

*7) Unlinkable:* It evaluates whether colluding verifiers can determine user credentials on other verifiers. As explained in *leaks-from-other-verifiers*, $NRTT$ profiles across different verifiers are uncorrelated. However, biometric factors (e.g., fingerprints) are obviously linkable due to unique user biometrics.

### B. Usability

*1) Scalable-for-Users:* It evaluates the burden on users who may have multiple accounts on different web services. For example, in legacy password systems, a user has to create a different password for each service. $NRTT$ can be scaled to any number of accounts per user without creating any extra burden. $NRTT$ profiles for the same user on different verifiers are independent and are measured and stored by the server.

*2) Physically-Effortless:* It evaluates whether the authentication factor requires physical (as opposed to cognitive) user effort beyond, say, pressing a button. $NRTT$-based authentication does not require any such effort. On the other hand, SAC users need to transcribe passwords from their phones/email into browsers and fingerprint users need to scan their fingerprints.

*3) Efficient-to-Use:* It evaluates the time it takes the user to successfully login. We show that extra login latency overhead of $NRTT$-based authentication scheme (compared to the baseline of legacy passwords) is less than $185ms$, which is negligible and is completely unnoticeable by end users. On the other hand, SAC users have to wait for SMS message to get authentication code, which may take a few seconds, and biometric factor scanning may take several seconds.

*4) Easy-Recovery-from-Loss:* It evaluates the easiness by which users regain the ability to login if the login credentials are lost or forgotten. Authentication based on $NRTT$ is completely transparent to users as they do not keep or memorize any authentication credentials. It is highly unlikely (as proved by the results of our extensive experiments) for a legitimate user to fail login after three trials. However, in the very rare case of sudden and permanent changes in profile parameters, $NRTT$-based authentication (AMAN) generates new reference profiles after verifying the identity of the user using either: (i) out-of-band channels such as email or SMS, (ii) other complementary authentication schemes that may have been augmented with AMAN such as browser fingerprinting, (iii) if none of the previous options are available, a user can re-register with the web service as a new user. On the other hand, the loss of fingerprint credentials (e.g., physical damage to fingers) is permanent and impossible to be regained.

### C. Deploy-ability

*1) Accessible:* It evaluates whether users may be hindered from using the scheme due to disabilities or any other physical conditions. Again, $NRTT$-based authentication scheme is completely transparent to users and is easily accessible irrespective of any disabilities. On the other hand, SAC and biometric factor based schemes are less accessible. For example, blind users cannot read SAC from phones, and users who broke their fingers cannot scan fingerprints.

*2) Mature:* It evaluates whether the scheme has been tested and deployed in large scale real-world scenarios. $NRTT$-based authentication (AMAN) is a new authentication mechanism and hence this property does not apply. However, AMAN has been extensively tested with relatively good size prototypes (155 non-mobile users and 63 residential WiFi users). SAC is widely deployed but biometric factor based authentication has not been used for remote authentication.