Improving Usability of Complex Authentication Schemes Via Queue Management and Load Shedding

Larry Koved
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
koved@us.ibm.com

Bo Zhang
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
zhangbo@us.ibm.com

ABSTRACT

Complex authentication schemes, such as some forms of biometric authentication and context analysis, require large quantities of sensor data and the identify verification can be computationally intensive. This can result in long latencies from the time of the authentication challenge until the authorization decision is determined. This can be worse when there is congestion in the system due to excessive authentication requests, such as at the start of the business day or shift change. Interruptions can impact the user's short term memory, slow down task performance, as well as result in user dissatisfaction with the authentication system. This paper proposes stochastic models to represent the authentication process and offers queue management and load shedding solutions whereby various parts of the authentication process can be addressed to mitigate the delays.

1. INTRODUCTION AND APPROACH

Interaction with mobile devices is often brief and can be dominated by the time to authenticate [2]. Since authentication is a secondary task, we would like to minimize the effects of task interruption due to authentication (see [3, 4, 5] for more on this motivation). When considering new authentication schemes, including various forms of biometric authentication, behavioral authentication, and authentication schemes that consider contextual factors, there can be substantial delays introduced by the system to process the data. For example, processing a 10 second audio signal for voiceprint identification or verification can take 5-10 seconds. Identifying nearby bluetooth devices can also take 5-10 seconds. Cellular network data communication congestion can also slow down the authentication process.

We can model an authentication system as a series of queues, starting from the point of acquiring data from the sensor on an authentication device or *client*, network delays, server processing of the authentication data, and eventually to the response back to the client. When performing multi-factor authentication on a mobile device and authenticating to a network service, we may be collecting contextual data, such as location, visible network devices or access points, biometric data, etc. This data is sent over the network to an authentication and authorization service. Each of the steps in this process introduces delays. As the network or authentication service reaches saturation, delays increase. Figure 1 in the Appendix depicts a representative mobile biometric authentication system and the queues in the system.

We also can take into consideration the sensitivity of the user's requested operation, and compute a risk estimate for the operation (e.g., risk-based authentication [1]). In light of the risk, we also consider the values or weights that each of the authentication factors (context and authentication) contribute to the authentication confidence. Based on these factors, we can estimate how various authentication context factors and authentication results (e.g., password, biometric verification results) may contribute to an authorization decision. This may be a subset of all possible authentication factors. We can also consider history of authentication challenge requests and estimate the cost of performing the various stages of authentication, from sensor data acquisition to processing of the sensor data.

We know that some of the authentication factors require user interaction (e.g., biometric data acquisition), and others do not do so (e.g., context factor data collected from sensors). Collecting some of these factor data can require non-trivial amounts of time, such as speaking a passphrase, or collecting network device information from base stations or other mobile devices (e.g., 802.11, bluetooth MAC addresses).

Modeling the authentication process as a series of queues, we can aim to minimize authorization delays subject to the security constraints with respect to required authentication confidence. In effect, we can produce an improved schedule of requests to the client during the authentication process.

2. MODEL

We now describe a basic version of our proposed class of stochastic models for complex authentication schemes. It is a queueing model for systems with multiple classes of user requests and multiple authentication methods. Each user request is assigned into one class according to the desired level of authentication confidence in the identity of the user. Each authentication method produces a level of authentication confidence of the user's identity. We do not intend to explicitly capture in this basic model every aspect of the complexity of modern authentication schemes mentioned in the previous section, e.g., the class assignment of each re-

Symposium on Usable Privacy and Security (SOUPS) 2014, July 9–11, 2014, Menlo Park, CA.

quest can be a topic of interest by itself, and leave the development of more general models to a future paper. Yet, this basic model is already novel and effective: to the best of our knowledge, it is the first mathematical queueing model for complex multi-factor authentication systems from an operational perspective; also, this model facilitates the understanding of several key tradeoffs in such systems and leads to the formulation of a joint stochastic optimization and control problem that can be solved to address these tradeoffs. From a queueing theoretical point of view, our model differs from all conventional ones (see [6]) in that each arriving authentication request is explicitly prescribed a service time probability distribution, rather than endogenously possessing such a distribution.

Specifically, we consider an authentication system consisting of c identical authentication servers operating in parallel (e.g., biometric verification engines). There are I classes of user requests arriving to the system and class i requests, $i = 1, ..., I$, arrive according to a Poisson process with rate γ_i. We assume that the lower-indexed class a request belongs to, the greater the confidence is required in the user's identity. More specifically, each class i request is made by an imposter with probability p_i and $p_1 < p_2 < ... < p_I$. One may think of each user request in our model as a resource access request initiated by a principal.

The system has J authentication methods and if a request is processed by method j, it requires from one of the servers a service time exponentially distributed with rate μ_j, $j = 1, ..., J$. Between the I authentication request arrival streams and the pool of c servers there is a controller uniquely characterized by a function $\pi : \{1, ..., I\} \to \{1, ..., J\}$. This controller, or this function $\pi(\cdot)$, is chosen by the system designer to prescribe for (or recommend to) each of the I classes of requests one of the J authentication methods to be used. For each fixed $\pi(\cdot)$, we have J streams coming out of the controller, each stream corresponding to one *type* of authentication *jobs*, and clearly the rate of the J job streams depends on function π as well as all the γ_i's. Each 'job' in our model corresponds to an authentication challenge to the client. We emphasize the terminology used here: each *class of requests*, whose arrival rate is exogenously given, is associated with a particular imposter likelihood and each *type of jobs*, whose rate can be regulated by the controller, is associated with one of the J authentication methods. Figure 2 in the Appendix shows the model diagram with $I = 5$ and $J = 3$.

As mentioned above, each type j job requires an $\exp(\mu_j)$ service time. We assume a first-come-first-served discipline and certainly other service disciplines may be chosen. For type j job, or equivalently a request authenticated via method j, a 2-tuple parameter (α_j, β_j) is known: α_j represents the false rejection probability (type I error probability) and β_j the false acceptance probability (type II error probability).

Based on the above model description, some interesting tradeoffs arise. The first tradeoff is between security and delay (or congestion). From a pure security perspective, the system designer wants to set up the controller function $\pi(\cdot)$ in a way that all types of user requests are authenticated by methods with the highest degree of authentication confidence, i.e., with the lowest β_j value. However, these methods may be the ones with the longest service times, i.e, smallest μ_j, and doing so may lead to significant system delays. In addition, there is clearly an economic tradeoff between the system performance (including security and delay) and cost. The system performance obviously can be improved by increasing the number of servers c, which is done at a cost. A third tradeoff is between security and usability. The usability of an authentication method or system is partly correlated with the tail probability of the delay time in using it (i.e., the probability of each request's experienced delay exceeding a certain threshold, rather than the average delay, significantly affects the usability). In our model, the choice of both $\pi(\cdot)$ and c determines the delay tail probability, which in turn affects the usability.

The joint stochastic optimization and control problem in the context of our model is to choose the number of servers c and a controller π to optimize the security, delay, and usability subject to practical constraints.

Finally, we note that real-time scheduling may be adopted to further improve the performance of the system. For example, some of the user requests may be *delay-tolerant*, i.e., from a mathematical modeling point of view the request arrival times may be altered to any other point within a time window (say, 1-5 minutes). This load shedding approach can be useful for achieving temporal load balancing especially when the system workload varies over time. Consequently, the overall latency for authentication can be substantially reduced, as illustrated by Figures 3, 4, and 5 in the Appendix.

Figure 3 depicts a straightforward approach to implementing an authentication system where the data is collected on demand as needed. Figure 4 depicts a system where we have load shifted the context challenges in anticipation of the need for context data. The result is that the overall perceived responsiveness of the system has improved by reducing the authentication latency. Figure 5 depicts a more aggressive approach to authentication where passive collection of authentication credentials can be employed, such as with soft biometrics or behavioral biometrics. It is worth noting that, while the load-shifting approach has been used in other contexts such as smart grids, transportation systems, and data networks (e.g., see [7]), one unique feature in the authentication application is the effect of weakening authentication confidence. We shall investigate this effect in future work.

3. SUMMARY

In this paper we outlined the challenge of complex authentication systems that can introduce undesirable latencies, thus reducing the overall usability of the system. We propose using stochastic models to understand the system performance and identify the performance bottlenecks. Through various queue management mechanisms, we can reduce the perceived authentication latency. Also, as we can see from the timing diagrams (Figures 4 and 5), it is possible to reduce the overall perceived system delay in authenticating a user by load shifting.

Appendix

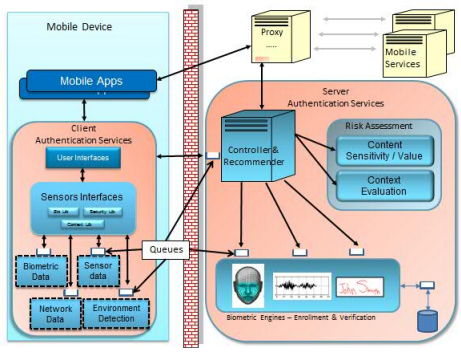

Figure 1: Example of modern authentication systems.

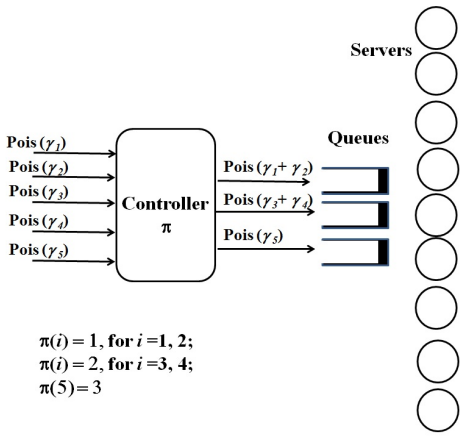

Figure 2: $I = 5, J = 3$.

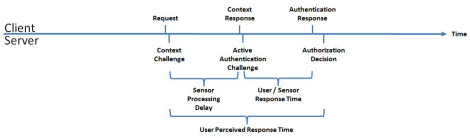

Figure 3: Authentication with on-demand data collection.

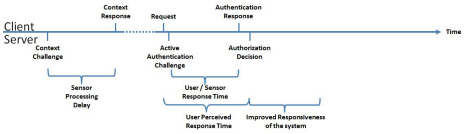

Figure 4: Authentication with time-shifted context challenge.

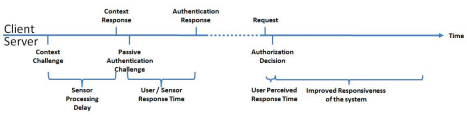

Figure 5: Authentication with passive authentication challenge.

4. REFERENCES

[1] J.A. Clark, J.E. Tapiador, J. Mcdermid, P. Cheng, D. Agrawal, N. Ivanic, D. Slogget. 2010. *Risk based access control with uncertain and time-dependent sensitivity*. In Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT '10). IEEE, Athens, Greece, 1-9.

[2] P. Bao and J. Pierce and S. Whittaker and S. Zhai. 2011. *Smart phone use by non-mobile business users*. In Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11). ACM, New York, NY, USA, 445-454.

[3] S. Nagata. 2003. *Multitasking and Interruptions During Mobile Web Tasks*. In Proceedings of the Human Factors and Erognomics Society. HFES, Santa Monica, CA, USA, 1341-1345.

[4] J. G. Trafton, E. M. Altmann, D. P. Brock. 2005. *Huh, what was I doing? How people use environmental cues after an interruption*. In Proceedings of the Human Factors and Erognomics Society. HFES, Orlando, FL, USA, 468-472.

[5] S. Trewin, C. Swart, L. Koved, J. Martino, K. Singh, and S. Ben-David. 2012. *Biometric authentication on a mobile device: a study of user effort, error and task disruption*. In Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC '12). ACM, New York, NY, USA, 159-168.

[6] D. Gross and C. M. Harris. 1998. *Fundamentals of queueing theory*. Wiley. 3rd Ed.

[7] P.M. Van de Ven, B. Zhang, A. Schorgendorfer. 2014. *Distributed backup scheduling: modeling and optimization*. Forthcoming in Proceedings of the 33rd Annual IEEE International Conference on Computer Communications (IEEE INFOCOM '14). IEEE, Toronto, Canada.