Using Dropped Call as an Authentication Factor

Balwinder Sodhi
Dept. of Computer Science and Engineering
Indian Institute of Technology Ropar, PB 140001, India
Email: sodhi@iitrpr.ac.in

***Abstract*—Increasing numbers of remotely accessed software applications are adopting Two Factor Authentication (TFA) methods, particularly when performing sensitive actions such as payment transactions. TFA methods, though addressed several weaknesses of purely password based authentication systems, have their own challenges such as their adverse effect on usability and, most notably, the operating cost. For instance, in a TFA mechanism that relies on sending a one-time password (OTP) to user's phone via SMS, the cost of just sending OTPs can be prohibitive for high volume transactions e.g. in case of an e-commerce payment gateway.**

We introduce "dropped call" initiated from a user's phone as a new authentication factor (AF), and present a novel authentication system that uses this new AF. We refer to a phone call which is instantaneously rejected by the callee as a dropped call. This system eliminates operational costs associated with a second factor of authentication. The proposed system can also be used as a sole authentication factor to build a passwordless authentication system. Analysis and evaluation of proposed system w.r.t various attack scenarios, performance and cost implications has been discussed. We show that cost savings in comparison to SMS based OTP transmission system are proportional to the volume of transactions. Considering a volume of 50,000 daily transactions and current pricing of sending bulk SMS (in UK), the cost of proposed system is less than 1% of the SMS based OTP alternative. An actual implementation of this system is deployed at: http://www.dcauth.in

***Index Terms*—dropped call; missed call; user authentication; authentication factor**

I. Introduction

Internet based Web applications such as email service, e-shopping portals and a variety of mobile phone apps are examples of applications that require their users to authenticate before allowing them access. Another significant use case occurs in e-commerce systems where payment gateways often send an SMS based OTP for authentication to a buyer making an online payment transaction. In most such systems the authentication protocols are centred around one or more *authentication factors* (AF). Such an AF may take the form of an entity that a user: (i) carries e.g. a secure token, (ii) endowed with e.g. some biometric attribute such as voice, finger prints etc., or (iii) knows e.g. a password/PIN. Depending upon the need, an application may use one or more of these AFs to authenticate a user/client.

When a server wants to authenticate a user, it presents to the user a challenge in response to which the user is expected to provide a quantity which the user can compute only if he/she has access to one or more AFs. Large number of Web applications and mobile apps make use of password (something that user *knows*) based authentication. A significant and growing number of applications, particularly e-commerce payment gateways, make use of two factor authentication (TFA), where in addition to a password the user is expected to supply a pseudorandom code. Such a code is either generated at the user's end by a hardware or software device, or it is sent to the user over another channel such as telephone or email.

Purely password/PIN based as well as existing TFA based systems both have certain shortcomings. For instance, an average Internet user today has more than half a dozen login/password pairs. Remembering and keeping each such password safe is a hassle. A password based authentication requires that the channel over which password needs to be sent to server during authentication must be secure (e.g. using SSL etc.). Protecting passwords stored at the server is also challenging. Similarly, in case of TFA mechanisms too there are significant challenges (discussed in Sec. I-A).

Therefore, a user friendly, affordable yet strong authentication mechanism is always desirable. Work presented in this paper is mainly targeted for use cases (e.g. e-commerce payment gateways) where applications must rely on a second factor of authentication. However, the proposed authentication mechanism can also be used for implementing a user login for an online system.

A. Related work

A detailed survey of various recent authentication mechanisms including TFA based systems has been presented by [1]. Most of the authentication mechanisms studied in [1] use two AFs: a password or PIN as one AF, and as a second AF they use an ever changing pseudo random code, called One Time Password (OTP). The OTP is typically generated by a hardware or software based device which often employ an implementation of Time-Based One-Time Password Algorithm [2]. Google Authenticator [3] is one such example. Several of the recent authentication systems make use of SMS based mechanism [4] to send an OTP to the user who transmits it back to authenticating server. The study carried out by [1] does a detailed analysis of several aspects (e.g. cost, usability, performance etc.) of current authentication systems. It confirms usefulness of TFA and also clearly points out the non-negligible additional costs that arise due to introducing second factor in authentication. Such costs are attributed to things such as sending OTPs via SMS and cost of a secure

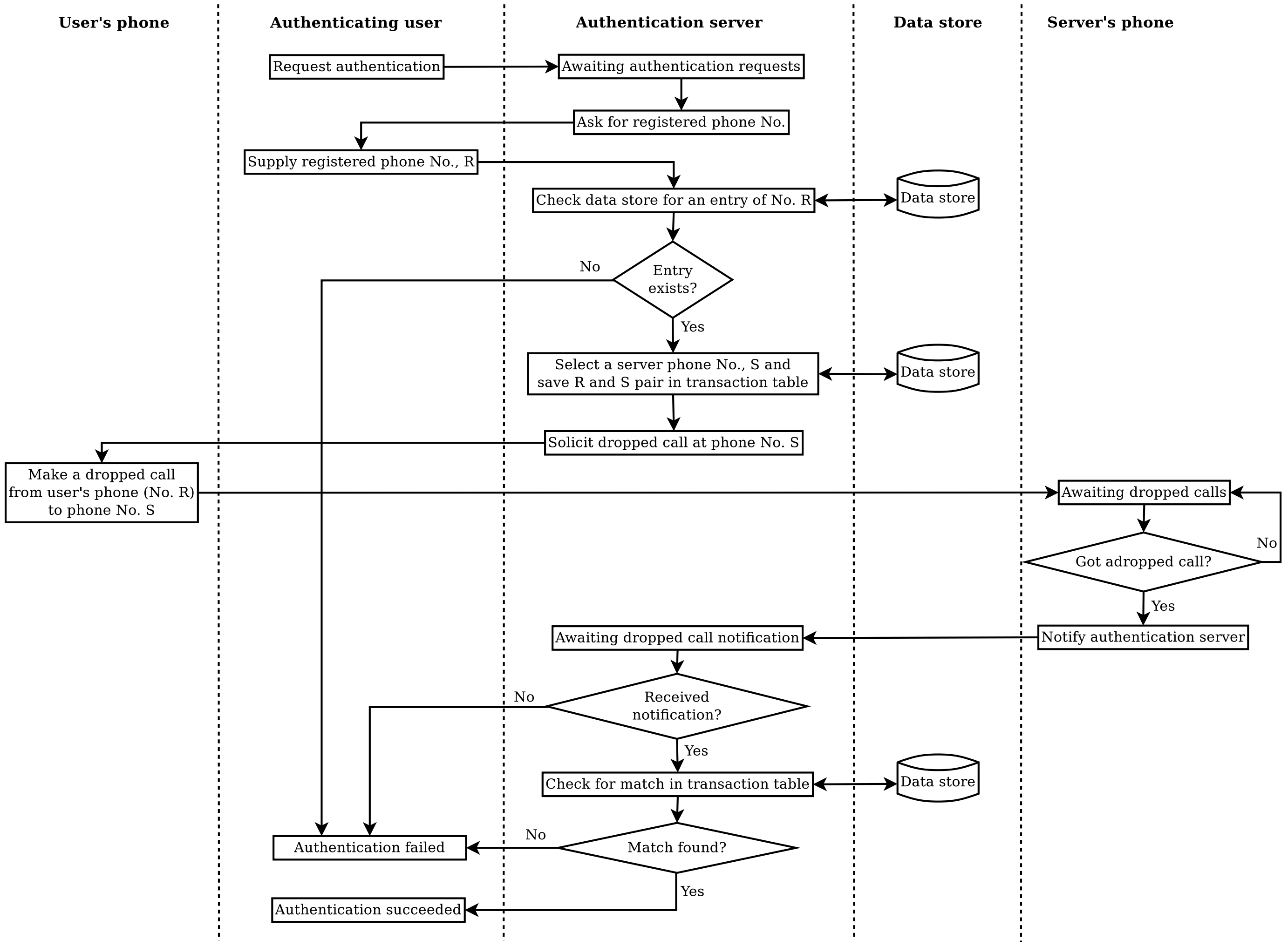
User's phone
Authenticating user
Authentication server
Data store
Server's phone
Request authentication
Awaiting authentication requests
Ask for registered phone No.
Supply registered phone No., R
Check data store for an entry of No. R
Data store
No
Entry exists?
Yes
Select a server phone No., S and save R and S pair in transaction table
Data store
Solicit dropped call at phone No. S
Make a dropped call from user's phone (No. R) to phone No. S
Awaiting dropped calls
No
Got adropped call?
Yes
Awaiting dropped call notification
Notify authentication server
No
Received notification?
Yes
Check for match in transaction table
Data store
No
Authentication failed
Match found?
Yes
Authentication succeeded

Fig. 1. Protocol steps

electronic token etc. Use of software based OTP generating clients such as Google Authenticator [3] have helped in reducing such costs, but they still require initial investment by the service provider for developing such software based OTP generators. Moreover, such OTP generators often require the users to have a smartphone, which leaves a significant number of users out.

This paper presents a novel technique to strongly authenticate users in a cost effective and usable manner. This technique introduces the use of "dropped calls" initiated from a user's phone[1] (*something that user carries*) as a new AF. A phone call that is deliberately not answered or completed is referred to as a *dropped call*. The proposed authentication technique assumes that the user carries a *secure* phone. Phone is termed as *secure* in the sense that no one else can make calls without permission and knowledge of the phone's user. Since dropped calls typically do not incur any costs therefore neither server nor the user incurs any costs for phone communication during authentication.

Rest of the paper is organized as: Section II describes the system's design. Section II-A describes a simple implementation and detailed steps involved in proposed authentication system are described in Section II-A1. In Section III we discuss evaluation of proposed system from cost perspective as well as main attack scenarios. Main conclusions have been presented in Section IV.

II. Design of the proposed system

As highlighted in preceding sections, existing TFA techniques depend heavily on secure generation and transmission of OTPs. Each of these TFA techniques incur a non-negligible cost for generating, exchanging and validating the OTP.

Purpose of OTP in existing TFA techniques is to verify user's identity via a separate channel (e.g. SMS, hardware or software OTP generator etc.). If we make use of some properties of modern telephony systems, we can eliminate the use of OTPs. Two properties of telephony networks are of particular interest in this context:

[1] In remainder of this paper we use "user's phone" to refer to the phone registered as user's identity with authenticating server.

1) A telephone network always[2] transmits a caller's identity (i.e. phone number) to the callee regardless of whether a call is answered or not.
2) Phone calls are typically not charged to any party unless answered at the callee's end.

Design of proposed system leverages the above two properties of telephony networks. Main tenet, and a novel aspect, of our design is the use of "dropped call" initiated from user's phone as the main AF. Dropped calls do not incur any costs for caller or the callee. Therefore the system proposed in this paper significantly lowers or eliminates such costs. User's phone number is treated as his/her identity at the server.

The steps involved in a basic operation of the proposed authentication method are depicted in Fig. 1.

A. Implementation example

An example implementation of the authentication system has been described in Fig. 2 which shows the logical structure of implementation. Its major elements are as follows:

1) **Authenticating client** has two main parts: User's registered phone and a web browser (or another client such as a phone app).
2) **Authentication system** comprises of the following main parts:
    a) Pool, $P = \{p_s^i : 1 \leq i \leq |P|\}$, of programmable phones, p_s^i, to receive dropped calls from users' phones.
    b) Dropped call notification component deployed on (or along) server's programmable phones, p_s^i.
    c) An application server to host components for handling the user enrolment, authentication request handling and dropped call notifications from server phones, p_s^i.
    d) A data store for storing enrolled users identities and authentication transactions data.
    e) Suitable communication media that connects application server with the server's programmable phones, the data store and the authenticating client.

Before a user can be authenticated by the system, he/she must be enrolled with the authentication system. The application server handles the enrolment requests via a component deployed in it. A user is expected to provide his/her phone number as the identity for enrolment. Application server stores the user's identity in a data store.

An actual system based on this design which offers dropped call based authentication service is deployed at: http://www.dcauth.in.

1) Authentication process: Steps that an authenticating client performs are described as follows.

1) Authentication application server awaits requests for authentication from client devices.
2) A user sends an authentication request to the authentication server at its designated URL.

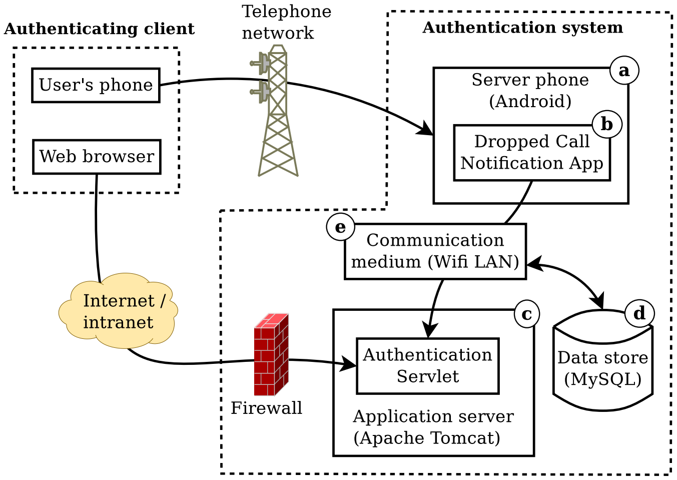

Fig. 2. Implementation example

3) In response to authentication request from a client the authentication server asks the user for his enrolled phone number.
4) If the user responds to the authentication server by supplying his enrolled phone number, p_u, then the authentication server will check its data store for an entry of p_u. If the entry is not found then authentication fails.
5) If an entry for p_u is found in data store then the authentication server: a) randomly selects an available phone number, p_s, from its phone pool P, b) prompts the user, at time t_{auth}, to make a dropped call from p_u to p_s and c) saves the tuple $< t_{auth}, p_u, p_s >$ in a transaction table.
6) Phones in server's pool of phones await dropped calls. If at time t a dropped call from any number p_u^i is detected by a phone p_s^j in the pool, a notification containing the tuple $< t, p_u^i, p_s^j >$ is sent by the phone to the authentication server.
7) On receiving a dropped call notification $< (t_{auth} + \Delta t), p_u, p_s >$ the authentication server checks whether a pending authentication transaction for $< t_{auth}, p_u, p_s >$ exists in data store. If such a pending authentication transaction exists and if the notification was received within a predefined time, Δt from t_{auth} then authentication succeeds. Else the authentication fails.

III. Evaluation and Analysis

As discussed in the preceding section, central idea of the proposed authentication system is the use of dropped calls as an AF. We discuss below the attack scenarios arising *mainly due to* the use of dropped calls.

A. Attack scenarios

Following are the main scenarios in which authentication system may break.

1) If a denial of service (DoS) attack is launched by keeping the server phone ringing for extended duration.

[2] Unless caller has subscribed to a private number.

2) If the user's phone is compromised such that a malicious entity is able to initiate phone calls from that number to an arbitrary phone number.
3) If the telephony network can be spoofed to send (or otherwise sends) to a call recipient an arbitrary phone number as the caller's identity. Call recipient in this system refers to the phones in server's phone pool.

The above factors are main determinants for the strength of proposed authentication system. About DoS attack, we believe that probability of an attacker launching such an attack will be quite low because his/her identity (i.e. the phone number) is always disclosed to the server phone. Even if an attacker makes dropped calls from a private number his/her identity can be easily obtained from phone service providers. Identifying attackers is much easier (an effective deterrent) in this case than in the case os IP network based DoS attackers. Furthermore, impact of such DoS attack can easily be contained because dropped call notification component immediately disconnects the caller as soon as phone state changes from idle to ringing and caller's number becomes available.

Several well-known implementations [5] of TFA which assume security of user's phone (i.e. OTP generator/recipient device) have stood the test of time. We, therefore, believe that in practice it is safe to assume that the probability of a user's phone getting compromised is low. The robustness and security of telephony network can also be safely assumed.

B. Performance implications

We use throughput as the performance indicator for our analysis. We estimate throughput as the number of authentication requests completed per unit of time. Total time taken to handle one authentication request can be expressed as:

$$T_{auth} = T_{dial} + T_{setup} + T_{algo} \tag{1}$$

Here,

- T_{dial} is the time taken by a human user (or an automated agent such as a client App) to dial a missed call to a number sent by the server.
- T_{setup} is the time taken by the telephony network to set-up the call, i.e., duration from the moment when caller completes dialing the server's number until the server's phone detects the incoming call.
- T_{algo} is the time taken by authentication server to execute authentication algorithm steps.

We have assumed that the server's phones (which receive dropped calls) are programmed to reject the incoming calls as soon as they detect the caller's phone number.

In our experiments we have observed that a human user can read a 10-digit phone number displayed (using adequately readable font and format) on a screen and dial it using his/her phone within 12 s in normal situations.

However, T_{setup} varies depending on the network technology, among other factors. Even a relatively older GSM specification [6] requires the expected call set-up time to be below 5 s for normal set-up and below 10 s for slow set-up. As an indicator of real call set-up time we looked at its values as observed [7] on some of the major roads of mainland Portugal: maximum is about 14 s and average is just below 5 s and minimum being just below 3 s. On a modern PC having a dual-core CPU and 4GB RAM, T_{algo} is typically below 100 ms and is very small in comparison to T_{dial} and T_{setup}. Therefore, we can safely assume:

$$T_{dial}^{max} \leq 12 \text{ s}, T_{setup}^{max} \leq 14 \text{ s}, T_{algo}^{max} \leq 100 \text{ ms, and}$$
$$T_{auth}^{max} \leq 12 + 14 + 0.1 \approx 26 \text{ s} \tag{2}$$

Since SMS service is asynchronous and unreliable, transmitting and delivering an OTP via SMS on a given telephony network will always take more time than setting up a call on that network.

When performing authentication using SMS based (or token based) OTP a user typically reads the OTP from phone device (or hardware token) and then types that OTP into a suitable interface, typically a web form. In case of dropped call based authentication system, the user will read a phone number displayed on a suitable interface (typically a web page) and then type that number on phone's dial pad.

Our experiments with different users have shown that the latter takes slightly less time than the former. This may be due to small display format of most SMS applications that are available on phones. Therefore, for completing an authentication operation via SMS based OTP exchange will likely take more time than completing it using the proposed dropped call based authentication system.

1) Throughput and phone pool size: In order to ensure a reasonable service quality the authentication server maintains a pool of phones for receiving the dropped calls. On receiving an authentication request the server selects an *available* phone number from the pool. In preceding section we have given the estimate of time (T_{auth}) required to complete one authentication operation. Throughput (η requests/s) and pool size, $|P|$, are related as:

$$\eta = |P|/T_{auth} \text{ where, } T_{auth} \text{ is in s} \tag{3}$$

A server phone is likely to remain *allocated* to an authentication request for a time T_{auth}. Estimated upper bound for T_{auth} is 26 s (see Equation-2). Server processes authentication requests in FCFS manner by queuing them. Thus, to be able to handle one authentication request per second we need 26 phones in the pool. Throughput can be increased by increasing the pool size and reducing T_{auth}.

C. Cost implications

We have compared the cost of proposed system with two widely used TFA techniques: i) secure token and ii) SMS based OTP. Cost is compared for a scenario involving 10,000 users and 50000 transactions (typical for an average e-commerce portal) per day.

One of the leading secure token product is RSA SecurID Authenticator SID700. Its cost comes to approximately 16 USD a piece per year when purchased in bulk [8]. For a workforce/population of 10000 the yearly cost of just the secure tokens will be $16 \times 10000 = 160000$ USD.

Lets consider the SMS based TFA. Lowest price of bulk SMS packages [9], [10] in UK is about 0.03 USD per outgoing SMS message. For above scenario, yearly cost of just sending the SMS messages would be about $0.03 \times 50000 \times 365 = 547500$ USD.

For the above scenario proposed system incurs the following cost: a small monthly cost per phone line subscription plus the one-time cost of each basic phone set. For instance, a phone number without any data, SMS or voice minutes will cost about 2 USD/month plus about 50 USD as the cost of a basic phone set. Assuming 26 phones in the server's phone pool, total cost comes to be about $26 \times (2 + 50) = 1352$ USD. This cost is less than 1% of the cost of alternatives discussed above.

IV. Conclusions

A strong authentication system is a critical component of most applications which are accessed over the Internet. Keeping usability and security of software applications in a good balance has always been a challenge for engineers. Adoption of multi-factor authentication (MFA) mechanisms has helped significantly in addressing this issue. However, such systems have their own challenges and associated cost. For public facing Internet based applications, two most commonly used authentication factors in combination with a password/PIN are: (i) a One Time Password (OTP) generated by the server and sent to user's phone via SMS and (ii) use of a hardware or software based OTP generator device by the users. Implementing any of these mechanisms incurs non-negligible capital and operational cost.

This paper presents a novel technique to strongly authenticate users in a cost effective and usable manner. The proposed technique introduces use of "dropped calls" initiated from a user's phone as a new authenticate factor. Since dropped calls typically do not incur any costs therefore neither authenticating server nor the user being authenticated incurs any costs due to use of telephony service during authentication.

We have shown that the throughput to cost ratio of proposed system is better than existing mechanisms that make use of SMS based or hardware/software token based OTP generator for implementing a TFA system. Considering current prices of bulk SMS, we have shown that in comparison to the mechanism using SMS based OTP our method can save more than 0.5 million USD in yearly SMS costs for a scenario involving 50000 daily transactions. Cost effectiveness and strengths of the proposed method have been verified by a real implementation which is available at http://www.dcauth.in.

References

[1] E. D. Cristofaro, H. Du, J. Freudiger, and G. Norcie, "Two-factor or not two-factor? A comparative usability study of two-factor authentication," *CoRR*, vol. abs/1309.5344, 2013. [Online]. Available: http://arxiv.org/abs/1309.5344

[2] D. M'Raihi, D. M'Raihi, M. Pei, and J. Rydell, "Totp: Time-based one-time password algorithm," https://tools.ietf.org/html/rfc6238, Internet Engineering Task Force (IETF), retrieved: April 2015.

[3] Google-Inc., "Google authenticator," https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2, Google Inc., retrieved: April 2015.

[4] F. Aloul, S. Zahidi, and W. El-Hajj, "Two factor authentication using mobile phones," in *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on*, May 2009, pp. 641–644.

[5] Google-Inc., "Google 2-step verification," https://www.google.com/landing/2step/, Google Inc., retrieved: April 2015.

[6] ETSI, "Gsm technical specification gsm 02.67, version 5.0.1," http://www.etsi.org/deliver/etsi_gts/02/0267/05.00.01_60/gsmts_0267v050001p.pdf, European Telecommunications Standards Institute, retrieved: April 2015.

[7] P. ANACOM, "Mobile communications systems (gsm/umts) quality of service assessment, 2011," http://www.anacom.pt/streaming/QoS_Portugal_Continental_sumarioexecutivo_UK.pdf?contentId=1127763&field=ATTACHED_FILE, Autoridade Nacional de Comunicaes, Portugal, retrieved: April 2015.

[8] TokenGuard.com, "Rsa securid 700 authenticator: The gold standard in two-factor authentication," http://www.tokenguard.com/RSA-SecurID-SID700.asp, TokenGuard.com A division of Virtual Graffiti, Inc., retrieved: April 2015.

[9] SMSCountry.com, "Bulk sms pricing and coverage," http://www.smscountry.com/SMSCoverage.aspx, SMSCountry Networks Pvt. Ltd, retrieved: April 2015.

[10] www.twilio.com, "Sms pricing for text messaging - twilio," https://www.twilio.com/sms/pricing/gb, Twilio Inc., retrieved: April 2015.