A Dynamic Authentication Scheme for Mobile Transactions

Sathish Babu B. and Pallapa Venkataram
(Corresponding author: Pallapa Venkataram)

Protocol Engineering Technology (PET) Unit, Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore, 560 012, India (Email: {bsb, pallapa}@ece.iisc.ernet.in)
(Received Jan. 24, 2008; revised and accepted Mar. 07, 2008)

Abstract

Most of the existing authentication schemes for mobile communication are static in nature, and principally dependent on strength of authenticating identifiers for users identity. The acceptance of all the transactions of a user under a single authentication level is vulnerable. We propose a novel transaction based authentication scheme (TBAS) for mobile communication using cognitive agents. The proposed approach provides range of authentication based on mobile transaction sensitivity, and users behaviors. The TBAS uses mobile agents to gather user behaviors, and static agents for detecting transaction sensitivity, user history analysis, and for choosing appropriate authentication actions. The method has been simulated using the agent factory framework for cognitive agents generation, and their communication. The performance analysis, and the simulation of the proposed system shows that, there is a considerable reduction in the security cost compared to regular session based authentication schemes. By combining transaction based authentication with behavior analysis the authentication attacks can be effectively identified.

Keywords: Authentication, cognitive-agents, mobile communication, mobile transactions, security

1 Introduction

Mobile communication, and services over emerging wireless technologies provide anyone, anytime, and anywhere access. Increased importance in mobile telecommunication, and dominance of data communication promoted large segment of users to accept the mobile data communication as a part of their day-to-day activities. However, the wireless medium has certain limitations over the wired medium such as: open access, bandwidth insufficiency, complex system functioning, power confinement, and relatively unreliable network connectivity. These limitations make it difficult to design efficient security schemes for authentication, integrity, and confidentiality. Wireless networks, and the current generation of 3G networks have a packet switched core which is connected to external networks such as the Internet, making it vulnerable to new types of attacks such as denial of service, viruses, worms, channel jamming, unauthorized access, eavesdropping, message forgery, message reply, man-in-the-middle attack, session hijacking, etc., similar to the Internet [14]. Out of many security issues of mobile communication, the focus of this paper is designing an effective, dynamic, and intelligent decision based authentication technique for mobile communications.

1.1 Mobile Authentication

Authentication is a process to identify a mobile user (MU), in order to authorize him/her to use system resources for specified purposes. Authentication involves negotiating secret credentials between prover, and verifier for protecting communications. The primary aim of any authentication protocol or a scheme is "verifying the linkage between an identifier (usually claimed by the individual, but sometimes observed), and the individual [23]." Most of the existing authentication schemes may be broadly classified into three categories [27]: 1. *application level authentication,* where the user enters the application level data such as user-ID, password, PIN's, OTP's or some times bio-metric information as the basis for authenticating communications between the endpoint device, and the service provider's server. 2. *device level authentication,* in which the end systems may be servers or client devices, have some form of secrets used by cryptographic algorithms running on these systems. These secrets are either of type shared or unshared, which could be bound to hardware in use, for e.g., cryptographic key bound to the SIM of a mobile device, and finally 3. *network level authentication,* enables the exchange of session keys based on the public/private key pairs of the two mutual authenticators.

Device based authentication protocol is one of the common type of authentication practiced by mobile based application service providers [14]. Here, it is essential to register the device in advance to use the service. Even

though the authentication mechanism looks stringent, it does not be able to detect service misuse from compromised mobile devices. It also indirectly limits the users freedom of changing the device at his/her will, which is very common in a mobile environment.

Authentication services effect QoS in several ways. For example, the public/private-key based authentication mechanisms consume more time, and power due to the computational complexity of encryption, and decryption of data [15]. To achieve efficiency in authentication, challenge/response authentication mechanisms based on secret keys are widely used in wireless networks [3, 24, 26, 29, 34]. Most of the proposed schemes are static in nature, they provide a common scheme of authentication irrespective of the sensitivity of the communication going on. As a result of this, the authentication protocols fails to establish a proper relationship between correct identifier, and correct principal. This leads to a situation, where the correct identifier submitted by incorrect principal is validated, and authenticated to get the services.

The attackers are becoming successful in defeating single-factor application level authentication, using *social engineering; passphrase guessing; phishing; pharming; Trojans; malware* [5, 10, 13]. To overcome this a two-factor authentication at application level was introduced, which combines something that a user knows with something he/she possesses, but it is not foolproof, since the failure modes for different authentication factors are largely independent [23]. For example, the proper working of mobile device is independent of the user remembering passphrase or PIN. The session level implementation of a two-factor authentication makes the user remain authenticated for the complete duration of the session, i.e., until they log off or close the browser. But it is a catch-all approach, meaning users will be kept authenticated regardless of type of transactions, which attributes the same level of security to all the transactions.

Transaction-based authentication schemes are one of the solutions proposed in this direction. Such schemes enable a strong authentication at a transaction level, instead of only depending on the strength of the identifiers during authentication. Although the exact nature of transactions will not be revealed, certain characteristics of mobile transactions can be identified. These characteristics can be utilized to implement an efficient transaction based authentication scheme for mobile communication. We have shown here importance of the transaction based authentication at the application level in mobile networks, over the session based approach for authentication, which is commonly used in mobile networks.

1.2 Mobile Transactions

The tasks performed by users in mobile environments are categorized into two types [11]: The *transactional tasks* which update the database of the services, and *information retrieval tasks* which are limited to browsing, and searching activities. In the context of mobile computing, the mobile transactions (MTs) are, transactions whose execution environments involve mobile affiliations, i.e., a group of mobile hosts operating under necessary infrastructure. Any host in a mobile affiliation can initiate mobile transactions [25]. The research community has proposed many models for MTs, to name a few [2]: *Clustering, Two-tier replication, Pro-motion, Reporting, Semantics-based, Prewrite, and others.* These models generally classify the MTs into fixed host transactions, and mobile host transactions. The fixed host transactions include authentication services with large databases, and software systems usually available on the base stations, the authentication servers or on the cluster heads in the case of mobile ad hoc networks. Mobile host transactions work with limited data using compact software placed over the mobile devices.

1.3 Cognitive Agents

Our proposed authentication model use an intellective approach for mobile authentication using a type of intelligent agents, called Cognitive Agents (CAs). The reasoning capabilities of CAs enable them to infer, rather than looking up its responses to percepts generated. CAs are often intentional, which means that their actions are motivated by specific goals, and they store a symbolic representation of the world available. The cognitive agents knowledge organization, and deduction mechanisms are similar to human thinking. It includes knowledge quantifiers like behaviors, observations, beliefs, desires, and intentions. A rational approach towards identifying the correct principal can be established, by using these type of agents. The cognitive acts like thought, judgment, and assertions can be used for the following environment based decisions [35]: 1. Perceiving information in the environment; 2. Reasoning about those perceptions using existing knowledge; and 3. Acting to make a reasoned change to the external or internal environment [1, 19, 30, 31, 32].

In a desktop environment there is a consistency in the user behaviors, but seldom varies with the change of service he/she is using. But in a mobile environment, the user behavior is highly volatile, it changes with service, device, network, distance, time, location, cost, etc. Therefore the signature/anomaly detection types of schemes used in wired networks, may not be efficient in mobile systems. The proposed authentication scheme uses belief generation, and security analysis models to study the mobile user behaviors. The belief system grades an user, challenge him/her as per the deviation of their present, and past behaviors. The CAs are used for belief generation, and belief analysis.

The security concerns related to mobile agents are one of the main drawback to the widespread use of this technology. The security threats for mobile agents can be classified into four broad categories: *agent-to-agent, agent-to-platform, platform-to-agent, and other-to-agent platform* [21]. There are many techniques devised for pro-

tecting an agent platform which includes: *software-based fault isolation, safe code interpretation, state appraisal, path histories, proof carrying code. partial result encapsulation, mutual itinerary recording, execution tracing, and environmental key generation.* In the proposed system, the static agents digitally signs the mobile agent when it was instantiated, which greatly reduces the security vulnerabilities. Since an attacker can not change the code of the mobile agent to cause it to be malicious.

1.4 Proposed Authentication Scheme for Mobile Transactions

The transaction based authentication scheme (TBAS) uses two types of cognitive agents: *mobile cognitive agent (MCA)*, and *static cognitive agent (SCA)*, which are secured with respect to their construction, and inter-agent communications. The SCA creates the MCA, and sends it to the respective mobile node. This is done while authenticating the client. The total authentication scheme is distributed into two logical components: MCA-component, and SCA-component. The MCA generates beliefs over user transactions by observing various user behaviors periodically. The SCA dynamically generates authentication requirements, using the sensitivity of mobile transactions, and the changing beliefs on a user. A challenge/response protocol has been incorporated in the system to counteract, some common attacks such as: transaction interruption, transaction modification, and transaction fabrication. We analyze the effect of proposed authentication scheme over two of the QoS parameters: *Authentication delays*, and *Authentication costs.*

1.5 Organization of Rest of the Paper

The rest of the paper is organized as follows, Section 2 gives some of the related works, Section 3 provides definitions, and terminologies, Section 4 discuss the functioning of the TBAS, Section 5 provides analytical modelling of the belief analysis, authentication delay, and cost, Section 6 gives a simulation procedure with results, and finally Section 7 draws the conclusions.

2 Related Works

Hwang [20], presents a new WPKI (Wireless Public Key Infrastructure) architecture using a bridge certification authority cluster to achieve m-commerce security goals such as high scalability, fault-tolerance, cost-effectiveness in trust-path discovery, and in mapping the security policy. Li-Sha, and Zhang [16], focus on cryptography, and authentication protocols in mobile commerce. The work presents an asymmetric end-to-end authentication protocol using wireless access to home network of a mobile station to assist its authentication with a service provider.

Trusted transaction management [28, 38, 39], is a crucial component of mobile commerce. Transaction management in [37] focus on technical challenges of managing transactions in group-oriented mobile commerce services by presenting a framework, which includes requirements, membership-management, and support for dependable transactions. A dynamic work group (called mobile affiliation model) that supports sharing information (through an export-import repository both synchronously, and asynchronously) among mobile transactions is presented in [25].

Chen in [12] has proposed, a new authentication scheme for accessing contents, services, applications in both mobile device, and Internet. The services, and applications are divided into four groups according to their importance: extremely confidential group, very confidential group, confidential group, and free accessible group. The authentication usage levels are used to access the items in each of the four groups. The scheme doesn't make any attempts to categorize transactions happening in a particular group, as a result of this transaction based attacks are still possible.

A behavior based intrusion detection for mobile phone systems, proposed in [9], which is used for fraud detection of impostors, and improper use of mobile phone operations. In network security management with intelligent agents [8], two security layers have been proposed to manage the global security of a network, and local domain security. Three agents are used in each layer to perform security tasks. The manager layer agents interacts with the local layer agents by sending goals, delegating specific monitoring/detection tasks, receiving pertinent reports, and alarms.

VMSoar: A cognitive agent for network security [6], proposes a cognitive agent based intrusion detection model. The main aim of the model is recognizing plans of users, and what goal the user wants to achieve, whether it is a threat to the security of system, etc. It also claims to generate future expected behaviors of an user. This model is proposed for wired networks, which is not suitable for mobile environment due to intensity of computation involved.

3 Definitions

In this section we provide definitions for some of the concepts used in the paper.

Behaviors:

The behaviors refer to the actions or reactions of a user while formulating, and executing transactions. The behaviors are modelled using set of parameters. For example, an account usage behavior can be deduced based on: *transaction time, duration, frequency, mobile device used, velocity of mobility, and so on.* In general the probability P_{Bh_i} of generating a behavior Bh_i is computed using the behavior parameters set BP_i. The required number of behavior parameters varies from one behavior to another. It is also possible that, the same behavior

parameters may produce different behaviors depending on the value they acquire. For example the behavior parameters like, time-of-login, location-of-login, number-of-login-failure-attempts, and so on, may produce behaviors such as "stranger entering user account", and "the regular user behaving abnormally".

$$P_{Bh_i} = \frac{\sum_{k \in BP_i} W_{bh_k} * V_{bh_k}}{\sum_{k \in BP_i} Normal_{bh_k}} : \sum_{k \in BP_i} W_{bh_k} = 1.$$

Where W_{bh_k}, V_{bh_k}, and $Normal_{bh_k}$ are the weight, current value, and the normalized value of the behavior parameter bh_k respectively.

Observations:

In our authentication system an observation is the summarization of various behaviors exhibited by a mobile user during transaction execution. For example, in a mobile shares market application, combination of behaviors could produce observations like: *interests in shares, customer type, and spending habits.* The observation probability P_{Ob_i} is computed using the union of occurrence of defined set of behaviors which leads to that observation. Let BH_{Ob_i} is the set of behaviors considered for observation Ob_i.

$$P_{Ob_i} = P({Bh_a}^{new} \cup {Bh_c}^{new} \cup {Bh_k}^{new} \cup \cdots \cup {Bh_m}^{new}).$$

Where ${Bh_a}^{new}, {Bh_c}^{new}, {Bh_k}^{new}, \ldots, {Bh_m}^{new} \in BH_{Ob_i}$.

Beliefs:

Primarily the "beliefs represent information about the world or an entity, perceptions received from the external world, and execution of events update the beliefs [22]". For example, a customer is believed to be *low-risk taker*, if there are supporting observations over *investment options, investment volume, and investment types.* The probability of occurrence of a belief P_{Bl_i} is the union of those observations which will generate that particular belief. Let O_{Bl_i} is the observations set for belief Bl_i. A specific example for belief formulation is given in Figure 1.

$$P_{Bl_i} = P({Ob_c}^{new} \cup {Ob_f}^{new} \cup {Ob_l}^{new} \cup \cdots \cup {Ob_n}^{new}).$$

Where ${Ob_c}^{new}, {Ob_f}^{new}, {Ob_l}^{new}, \ldots, {Ob_n}^{new} \in O_{Bl_i}$.

Beliefs database:

The beliefs database at the SCA stores the probability values for the various beliefs w.r.t. a mobile user. For example, in a mobile shopping type of application, the system has beliefs such as genuine customer, competitor, casual visitor, mischievous visitor, hacker, etc., based on the observations generated using the history of transactions. These beliefs are represented using the probabilistic values. The contents from the belief database is used to estimate the belief deviation during user transactions. The frequency at which the beliefs database is updated is application dependent, since some applications like shopping will witness more frequent changes in behaviors compared to applications like banking, where the behaviors remains consistent over a long period of time.

Authentication database:

The authentication database is attached to the SCA, and it is used to formulate a required authentication challenge based on the sensitivity levels of transactions, and the belief deviation. The database includes various authentication related information for each transaction level.

Transaction Log:

This database is used to maintain detailed log of all the transactions conducted by the mobile users. The structure of the database is application specific, and the contents are used to construct transaction based challenges while authentication.

Observations storage:

This is a temporary storage available at the MCA, for storing generated observations during session. The content of this storage is used for belief generation, and analysis.

4 The TBAS Using CAs

In this section first we discuss the mobile transactions classification based on their sensitivity, followed by working of the proposed TBAS in a given network, and finally the system functioning in handoff based mobile communication scenarios.

4.1 Mobile Transaction Classification

The information sensitivity, and length of secrets are used as the key factors in proposing security levels for information systems [7, 33, 36]. As we observe, not all mobile transactions can have the same classified information. Therefore, we argue that the nature of transactions should be involved to classify the security levels. We propose a classification of mobile transactions into four types of classes based on the degree of severity of information that they are handling [4]. The degree of severity will be extracted from the policies laid out by service providers or organizations based on all direct, and indirect consequences that results due to misappropriation of transactions. One of the sample instance of classification is shown in the Table 1. In the TBAS, we use quantifiers: minor, significant, and substantial in the context of users likely to be effected by misuse of transactions. For example, if misappropriation of a transaction might result in a risk to user's personal safety, then the transaction will be allocated to level 3.

The level-0 transactions are un-authenticated transactions, there is no requirement for a MU identification to

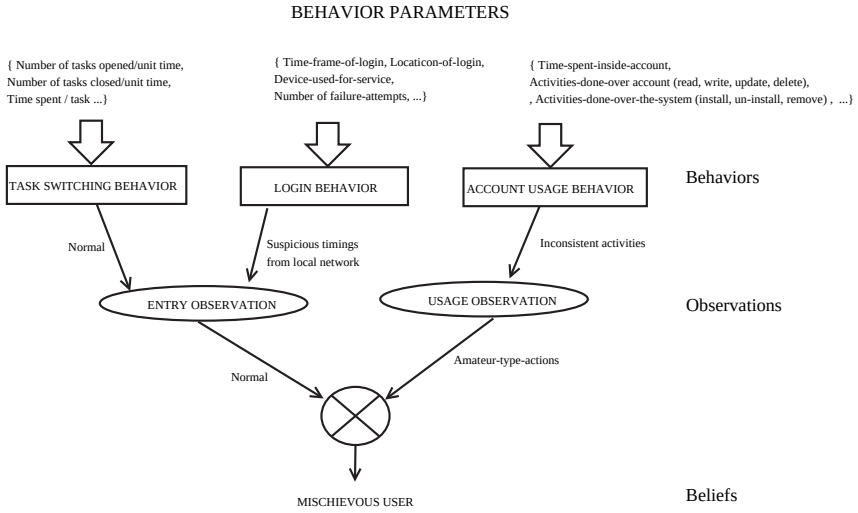


Figure 1: Belief generation example

execute these transactions. Level-1 transactions insist on some form of individualized data (which may be of type public, refer to Table 1), to build initial profile on a MU. At level-2, the system demands approved service identities for verification during authentication, through which non-repudiation situation could be handled. The level-3 sensitivity is used for high risk transactions, where confirmation of a MU identity is very much essential before the execution of a transaction. The level-2, and level-3 transactions are assumed to be secured with cryptographic techniques. The choice of security algorithms is based on transaction sensitivity levels. For instance, if the transaction level is 2, it would be enough to choose symmetric key based algorithms which work with stored keys on mobile devices, to encrypt/decrypt the authentication data or challenges. For level 3 transactions, the mutual authentication between two interacting parties is compulsory, which is efficiently achieved using public key based systems.

The effect of categorization of mobile transactions reduce the security cost in terms of execution of cryptographic algorithms, which is considered as major savings in a limited resource mobile environment. In case of a conventional authentication system all transactions should be cryptographically secured. As a result of this, there is a major reduction in key generation, encryption, and decryption delays in the TBAS. The authentication model of the TBAS uses rational approach towards attack detection, which is very useful in highly dynamic mobile environment.

4.2 The TBAS

The architecture of the TBAS is shown in the Figure 2, consists of both the cognitive agents, and their functional components. The TBAS can be hosted on base stations in case of cellular networks, authentication servers in wireless networks, and on the cluster heads in case of mobile ad hoc networks.

MCA:

The MCA migrates to a client mobile device along with the *belief formulator* logic during the service initiation request by a MU. The agent formulates beliefs using the *belief formulator*, and communicates them to the SCA along with transaction details. All the generated observations are stored in the *observation storage*. The MCA also provides the observations to the SCA, if required during belief analysis. The functioning of the MCA is given in Algorithm 1.

Algorithm 1 Working of the MCA

```
1: Begin
2: Initialize the observations storage.
3: Send service request to the SCA.
4: while Not end of user session do
5:    Accept transaction data T.
6:    Belief B ⇐ Call Belief Formulator (T).
7:    Send B and T to the SCA.
8:    if There is any request for observations from the
      SCA then
9:       Retrieve observations from the Observation Stor-
         age.
10:      Send observations to the SCA.
11:   end if
12:   Periodically refresh the observations storage
13: end while
14: End
```

Belief formulator:

The belief formulator is a component of the MCA collects various behavior parameters during the client transactions, generate behaviors of the client, and then observations. The beliefs over a client are deduced based on the new, and available observations from the *observation*

Table 1: Authentication levels and transactions categorization

Level	*Authentication type*	*Transaction sensitivity*	*Examples Transactions*
0	Not required	Minimal/No damage	Product general information browsing, downloading free samples, browsing other user feedbacks, etc.
1	**Individual authentication based on:** SSN, Driving License number, Employee ID, pseudonyms, e-mail address, device address, IP, etc.	Minor damage	Request for technical information, requesting comparative statements, requesting after sales service options, placing low volume orders, requesting feedbacks, requesting account statement, etc.
2	**Identity authentication based on:** login-name, password, PIN, TAN, OTP, etc.	Significant damage	Placing high volume orders, making macro payments, requesting purchase bills, requesting private information, making account transfers, etc.
3	**Attribute authentication based on:** Transaction history, Behavior biometrics, Physical biometrics, etc.	Substantial damage	Collecting health reports, making large advance payments, etc.

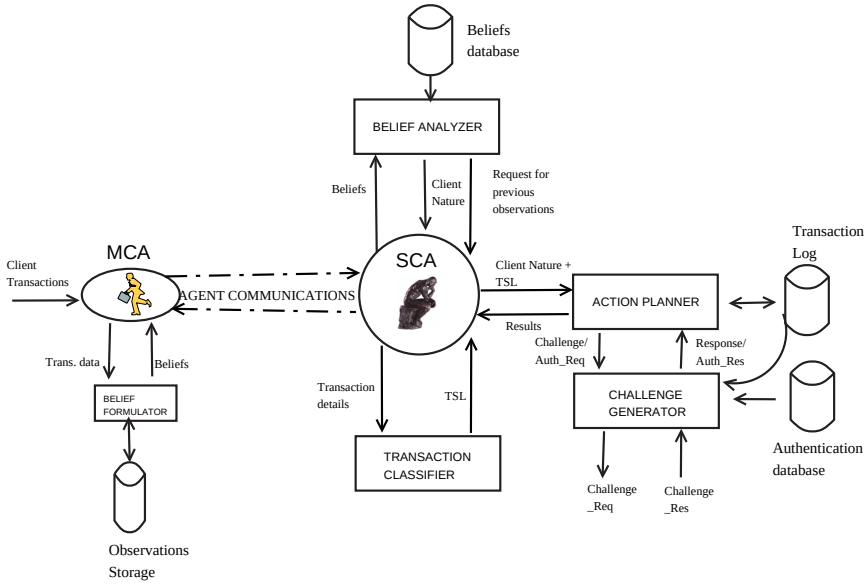


Figure 2: The TBAS architecture

storage. The belief data structure is created using sets of belief formulae, which is given by

$$(p\text{-}belief,\ t_1, \ldots, t_n).$$

Where *p-belief* is the predicate used to claim a value for a particular belief, and t_1 to t_n are terms, which are literals, and variables used to represent various observations on which the beliefs are reasoned. The given belief representation is compatible with cognitive agents created using the Agent Factory System (AFS) [18]. The working of the *belief formulator* is given in Algorithm 2.

Belief analyzer:

This module accepts newly generated beliefs on a MU from the SCA, and correlates them with established beliefs on a MU from the *beliefs database,* in order to identify the belief deviation after completing the transaction(s). The deviation function must required to be satisfy a distance property, where increased distance between two corresponding behavior values should produce higher deviations, and vice versa. The *belief analyzer* may also generate new beliefs to support belief analysis, especially when the user is turning out to be suspicious. Since the *observations storage*, is a part of the MCA, already generated observations on user transactions from the MCA could be obtained by sending a request to the SCA. Algorithm 3 is used by *belief analyzer.*

Transaction classifier:

The transaction classifier accepts transaction de-

Algorithm 2 Algorithm for the belief formulator

1: Begin
2: Accept T.
3: Begin
4: Initialize belief data structure of the MCA.
5: Let V= $\{v_1, v_2, \ldots, v_k\}$ is the set of values collected by the MCA for various temporal, and symptomatic behavior parameters from transaction T.
6: Generate the behaviors set BEH using V.
7: Generate the observations set O using BEH.
8: $\forall\ b_i \in$ belief data structure, select those beliefs which are triggered by O.
9: Let B is all selected beliefs.
10: Return B.
11: End

Algorithm 3 Algorithm for the belief analyzer

1: Begin
2: Accept B from the SCA.
3: $DF \Leftarrow 0$.
4: $B^{new} \Leftarrow B$.
5: Retrieve the established belief on a MU from beliefs database; say, $B^{established}$.
6: **if** the $B^{established}$ is not present **then**
7: Send request to the SCA to fetch from home network.
8: **end if**
9: **if** Observations are required to generate new beliefs for belief analysis **then**
10: Send request to the SCA to fetch from the MCA.
11: **end if**
12: $DF \Leftarrow |B^{established}, B^{new}|$
13: Return DF.
14: End

Algorithm 4 Logic for the transaction classifier

1: Begin
2: Accept transaction details T from the SCA.
3: Let **OP** is the operation requested by transaction T.
4: **if OP** is "RETRIEVE" **then**
5: Let **INFO** is the requested information to read.
6: **if INFO** is public **then**
7: TSL = 0.
8: **else if INFO** is personal data **then**
9: **if** personal public data **then**
10: TSL = 0.
11: **else if** personal private data **then**
12: TSL = 1.
13: /* More analysis on personal data – follows*/
14: **end if**
15: **else if INFO** is financial data **then**
16: TSL = 2.
17: /* More analysis on other readable items – follows*/
18: **end if**
19: **else if OP** is "WRITE" **then**
20: Let **D** is the target database for writing.
21: **if D** is public **then**
22: TSL = 0.
23: **else if D** is personal record **then**
24: TSL = 1.
25: **else if D** is transaction Log. **then**
26: TSL = 2.
27: /* More analysis on other writable items – follows*/
28: **end if**
29: /* More Analysis on other transaction type – follows */
30: **end if**
31: Pass TSL to the SCA.
32: End

tails submitted by a MU from the SCA, and finds the Transaction Sensitivity Level (TSL). The TSL is generated by analyzing various transaction parameters, like, *type of operation; time of operation; type of data; sensitivity of data; volume of data; etc.* This analysis produces a TSL ranging from level 0 to 3. The sample logic for the *transaction classifier* is given in Algorithm 4.

SCA:

The SCA co-ordinates the functions of all the components at the authentication server. It is responsible for migrating the MCA to a MU, and carrying out communications with the MCA. Upon receiving the beliefs, and transaction details from the MCA, the SCA submits them to the *belief analyzer*, and the *transaction classifier* respectively. Based on the value of cumulative deviation factor, the SCA produces one of the following three types of opinions on a MU: *NORMAL-USER, SUSPICIOUS-USER, ABNORMAL-USER.* The results obtained from these modules are passed onto the *action planner* for suitable authentication actions. The SCA also fetches the observations from the MCA, on request from the *belief analyzer*. The reason is that the *belief analyzer* could generate additional beliefs in order to substantiate the belief sent by the MCA. The functioning of the SCA is described in Algorithm 5.

Action planner:

Based on the values of TSL, and opinions on a MU, the *action planner* perform the following. All the transactions with sensitivity (TSL=0) are executed without any authentication by the system. The higher level transactions appearing for the first time, the *challenge generator* create an initial authentication challenge for that sensitivity level. Otherwise, the future actions are decided based on the value of a belief deviation factor. The algorithm 6 explains the working of the *action planner*.

Challenge generator:

This module is responsible for generating authentica-

Algorithm 5 Working of the SCA

```
Begin
Initialize cumulative deviation factor (CDF) to zero.
while Not end of user session do
    Accept B & T from the MCA.
    DF ⇐ BeliefAnalyzer(B).
    TSL ⇐ TransactionClassifier(T).
    if There is any request from the belief analyzer for
    observations then
        Fetch them from the MCA.
    end if
    if There is any request from the belief analyzer for
    new beliefs then
        Fetch the beliefs from the SCA of a MU's home
        network.
    end if
    Pass transaction details to the transaction classi-
    fier.
    Add DF to CDF.
    if CDF < Th_suspicious then
        User-nature ⇐ NORMAL-USER.
    else if CDF ≥ Th_suspicious and CDF < Th_abnormal
    then
        Generates additional beliefs (if required for con-
        firmation).
        User-nature ⇐ SUSPICIOUS-USER.
    else if CDF ≥ Th_abnormal then
        Generates additional beliefs (if required for con-
        firmation).
        User-nature ⇐ ABNORMAL-USER.
    end if
    Authentication-result ⇐ ActionPlanner(User-
    nature, TSL).
    if Authentication-result is Failure then
        Disconnect the client session.
        Deallocate the MCA from a mobile node.
    else
        CDF = CDF - DF.
    end if
end while
End
```

tion challenges, and attacks counteracting challenges during transaction execution. The challenge generator uses the information stored in the *authentication database, the beliefs database, and the transaction Log,* to reason out the challenge question. In order to safeguard the challenge system from phishing attacks, the challenges are encrypted using the security algorithms of the corresponding TSLs. The MCA decrypts challenge, obtains response from a MU, and sends the encrypted response to challenge/response module. Algorithm 7, shows the working of challenge generator, and the Table 2 shows some of the sample challenges.

Algorithm 6 Algorithm for action planner

```
Begin
for Each transaction T do
    Accept TSL and User-nature from the SCA.
    if TSL is 0 then
        Pass Authentication success message to
        the SCA.
        Execute transaction T.
    else if TSL is not encountered before then
        Instruct challenge generator to perform initial
        authentication of that TSL.
    else if User-nature is NORMAL-USER then
        Pass Authentication success message to
        the SCA.
        Execute transaction T.
    else if User-nature is SUSPICIOUS-USER then
        Instruct challenge generator to get the next au-
        thentication data of that TSL.
    else if User-nature is ABNORMAL-USER then
        Instruct challenge generator to create
        transaction-based challenges.
    end if
    if The response from challenge generator is "Suc-
    cess" then
        Pass Authentication success message to
        the SCA.
        Execute transaction T.
    else
        Roll-back transactions of that session.
        Pass Authentication failure message to the
        SCA.
    end if
end for
End
```

4.3 The TBAS in Handoff Scenario

To explain the working of the TBAS during handoffs, we assume every host in the wireless network runs the TBAS. When a MU at home/foreign network wish to use some service, he/she contacts the TBAS situated at corresponding home/foreign agent respectively. The TBAS migrate an instance of the MCA to the mobile device of a MU.

In a without handoff scenario, a MU is assumed to perform all its transactions with respect to a single TBAS situated either in a foreign network or in a home network. In case of a MU at a foreign network, mobile node perform initial registration using AAA (Authentication-Authorization-Accounting) resolution via the home network. During transactions the TBAS at foreign network may request established beliefs on a MU, and any authentication information from home network.

In a with handoff scenario, after the initial registration, a MU can either perform intra-domain or inter-domain handoffs. The intra-domain handoffs does not involve the AAA resolution via MU's home network. From the TBAS

Algorithm 7 Algorithm for challenge generator

```
Begin
if Transaction appearing first time and TSL > 1 then
    Create encrypted challenge to perform initial authentication of TSL.
else if User-nature is SUSPICIOUS then
    Create encrypted challenge for that TSL over Next data from authentication data set.
else if User-nature is ABNORMAL then
    Create encrypted challenge for that TSL over Transaction Log.
end if
Decrypt and validate the response obtained from the user.
if Response is correct then
    Send "Success" to the action planner.
else
    Send "Failure" to the action planner.
end if
End
```

point of view, when a handoff occurs, i.e., either intra-domain or inter-domain, the SCA of current (before handoff) network need to transfer authentication status to the SCA of next (after handoff) network securely, so that the authentication process will continue without any discontinuity. The status information essentially includes belief deviation factor of a MU, session key in use, challenge information, and beliefs received from the home network. With this approach of continuous authentication there is no need to restart the authentication process at foreign host, the TBAS has a better knowledge of the situation to do further authentication. The exchange of session key information will avoid key generation latency.

Table 2: Sample challenges

TSL	***User Nature***	***Example Challenge***
0	Any thing	No challenges.
1	NORMAL and First time	Please enter your mail-id?
	NORMAL but not First time	No Challenge
	SUSPICIOUS	Please enter your SSN?
2	NORMAL and First time	Please enter your login-name?
	NORMAL but not First time	No Challenge
	SUSPICIOUS	Please enter your Customer PIN?
	ABNORMAL	Please enter your favorite day of purchase?

5 Analytical Modeling

In this section we have provided the analytical model for the proposed system. The model is used to find out belief deviations, computing authentication delay at different sensitivity levels of transactions and computing corresponding security costs.

5.1 Belief Analysis

When the transactions are initiated by a MU, new values for behavior parameters are captured. Based on these values, the MCA computes probabilities of occurrence of various behaviors, observations, and beliefs for the current session (which are suffixed by *new*). The SCA calculates the deviation factor between the probability values of beliefs received from the MCA, i.e., ${P_{Bl}}^{new}$, with the corresponding established probability values of beliefs in the *beliefs database,* i.e., ${P_{Bl}}^{old}$.

$$DF(Bl^{new}, Bl^{old}) = |{P_{Bl}}^{new}, {P_{Bl}}^{old}|.$$

Exponentially moving averages are used to accumulate deviation factors of beliefs generated during various transaction instances. The weights for each transaction decreases exponentially, giving much more importance to current deviation while still not discarding older deviations entirely. The smoothing factor α is given by,

$$\alpha = \frac{2}{Number\ of\ Transactions + 1}.$$

The cumulative deviation factor (CDF) for beliefs at time t is given by,

$${CDF^t}_{Bl} = \alpha * DF(Bl^{new}, Bl^{old}) + (1 - \alpha) * {CDF^{t-1}}_{Bl}.$$

Thresholds have been established in order to take security actions, namely $Th_{suspicious}$, and $Th_{abnormal}$. The CDF within $Th_{suspicious}$ refers to transactions are normal. If the CDF is between $Th_{suspicious}$, and $Th_{abnormal}$, then the transactions are suspicious. When the CDF exceeds $Th_{abnormal}$ then the transactions are bizarre. Values for thresholds are computed using statistical deviation $SDev$ over the set of newly generated beliefs Bl^{new}, the weight W_{Bl} assigned to various beliefs based on history, and the step function γ provides distance between $Th_{suspicious}$, and $Th_{abnormal}$.

$$\begin{aligned}
SDev_{Bl_i} &= \sum_{i \in Bl^{new}} \sum_{j \in Bl^{new}} P_{Bl_i} * D(Bl_i, Bl_j) \\
Th_{suspicious} &= \sum_{i \in Bl^{new}} W_{Bl_i} * SDev_{Bl_i} \\
\sum_{i \in Bl^{new}} W_{Bl_i} &= 1 \\
Th_{abnormal} &= Th_{suspicious} + \gamma \\
\gamma &= \frac{\sum_i^n ({Th_{suspicious}}^i - \mu)^2}{n}.
\end{aligned}$$

Where μ is the mean of $Th_{suspicious}$ computed so far, and n is the number of times thresholds are computed.

5.2 Average Authentication Delay

We use the message transition diagrams shown in Figure 3(a) - (c), to derive the authentication delay , and cost in different sensitivity levels of transactions. A visiting MU sends a start of service request to a mobile application service provider(MASP) through wireless gateway(WG), and the TBAS. The WG is either an access point or base station. The SCA migrates an instance of the MCA to a MU for beliefs generation. The MCA at a MU collects behavioral parameters, and formulates beliefs, these beliefs along with transaction request $Tran.\_Req$ is sent to the SCA. The SCA performs belief analysis, and transaction analysis in order to decide the TSL, and the CDF. During level-0, the $Tran.\_Req$ is simply passed onto MASP without any authentication actions, and the response $Tran.\_Res$ from the MASP is sent back to a MU. At level-1, and level-2/3, the SCA generates authentication challenge requests i.e., $Challenge\_Req$ based on the value of the CDF, and the TSL. After validating the response given by the user i.e., $Challenge\_Res$, the $Tran.\_Req$ is forwarded to the MASP for execution. The major difference between the level-1, and level-2/3 is all the messages of level-2/3 are secured with suitable encryption techniques.

The delay in authentication of a transaction is defined as the time taken for a MU to receive the authentication reply for its request. The average authentication delay, T_{avg}, is defined as the sum of the authentication delay over a number, and type of transactions in a unit time.

$$T_{avg} = \sum_{l=0}^{3} \lambda_l {T_l}^i.$$

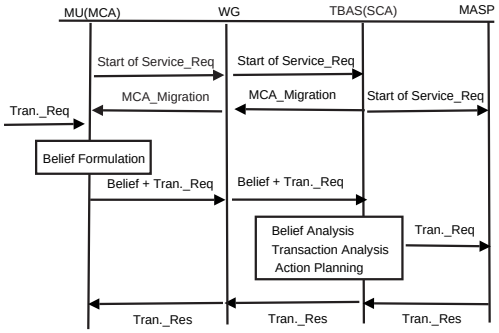
Where λ_l is the arrival rate of transactions of type-l, and ${T_l}^i$ is the authentication delay per transaction of the l type, with the number of occurrence as i.

The Table 3 shows set of time parameters used in computing authentication delay. The, ${T_l}^i$ can be expressed as

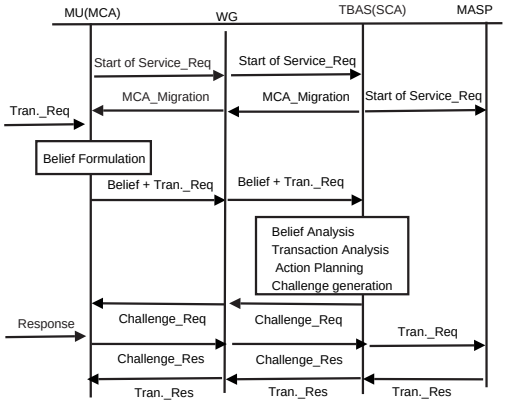
$$ {T_l}^i = cT_x.$$

Table 3: Authentication time parameters

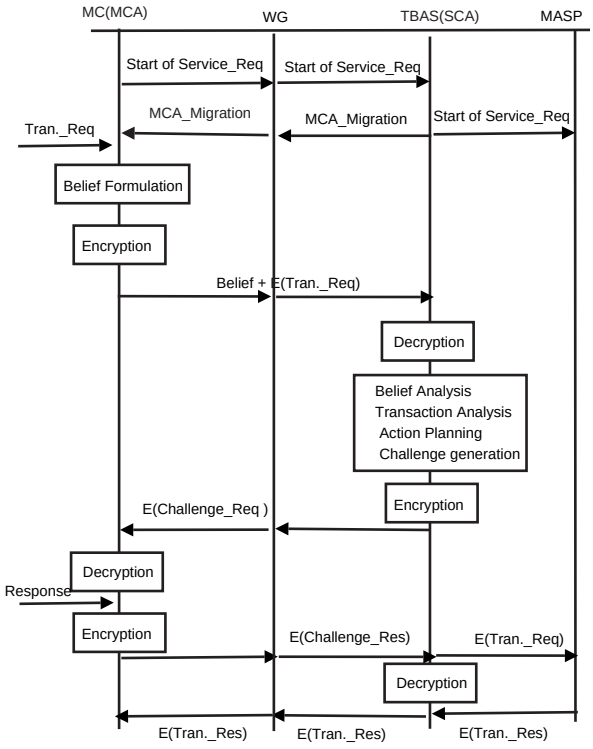
Symbol	Description
T_{pr}	Message propagation time on one hop
T_{tr}	Message transmission time on one hop
T_{bf}	Belief formulation time by the MCA
T_{ba}	Belief analysis time by the SCA
T_{ta}	Transaction analysis time by the SCA
T_{ap}	Action planning time by the SCA
T_{cg}	Challenge generation time by the SCA
T_{enc}	Time for encryption
T_{dec}	Time for decryption



(a) For level-0 transactions



(b) For level-1 transactions



(c) For level-2/3 transactions

Figure 3: Message transition diagrams

Where c is the coefficient of T_x, denotes the number of such time parameters required for level l authentication. If the number of hops between a MU, and the TBAS is N_h, then for various transaction sensitivity levels, the authentication delay per transaction are listed below.
For level-0 transactions, the T_{ap} is negligible, since no authentication actions are planned for these transactions, irrespective of the behaviors exhibited by a MU.

$$T_0{}^i = \ 4N_h(T_{pr} + T_{tr}) + T_{bf} + T_{ba} + T_{ta} + T_{ap} \quad i \geq 1$$

For level-1 transactions, additional hops are required for challenge, and response transmissions. These are generated during the first occurrence of these transactions, and when a MU shows the suspicious behaviors. Otherwise level-1 transactions authentication delay is same as level-0 transactions.

$$T_1{}^i = \begin{cases} T_0{}^i + 2N_h(T_{pr} + T_{tr}) + T_{cg} & \text{if } i = 1 \text{ or Suspicious;} \\ T_0{}^i & \text{Otherwise.} \end{cases}$$

For level-2/3 transactions, the number of hops remains same as that of level-1 transactions. But there is an additional delay of three pairs of encryption, and decryption operations in case of first appearance of level-2/3 transactions or when a MU is suspicious. Otherwise the authentication delay remains same as level-0 transactions with an additional delay of one pair of encryption, and decryption.

$$T_{2/3}{}^i = \begin{cases} T_1{}^i + 3(T_{enc} + T_{dec}) & \text{if } i = 1 \text{ or Suspicious;} \\ T_0{}^i + T_{enc} + T_{dec} & \text{Otherwise.} \end{cases}$$

The arrival rate of level-l transactions, i.e., λ_l, is given by,

$$\lambda_l = \lambda_u P_l.$$

The arrival of transactions from a MU is considered as a Poisson process with average rate λ_u, with the PDF of the transactions inter-arrival time as

$$f_A(t) = \lambda_u e^{-\lambda_u t}.$$

The P_l is the probability of occurrence of level-l transactions. By considering a particular time interval *(t, t+Δt)*, the number of level l transactions appearing in this interval is given by, *I(t,t+Δt)*. Since we assume the transaction arrival rate as a Poisson process, the P_l is given by,

$$P_l = \int_0^\infty P[I(t, t + \Delta t) = 1] = \int_0^\infty \lambda_u \Delta t e^{-\lambda_u \Delta t}.$$

5.3 Average Authentication Cost

The authentication cost is defined as the sum of signaling load, and processing load for cryptographic techniques during each authentication operation. The average authentication cost C_l, is defined as the sum of the authentication cost over a number of authentication requests per unit time at transaction level-l, which is given by,

$$C_l = \sum_\beta \lambda_\beta \Big[C_\beta{}^{(s)}(l) + C_\beta{}^{(p)}(l) \Big].$$

Where β takes the value based on the CDF generated during the belief analysis. The $\beta = 1$ if the CDF is $< Th_{suspicious}$. The $\beta = 2$ if the CDF is between $Th_{suspicious}$, and $Th_{abnormal}$. The $\beta = 3$ in case of CDF is $> Th_{abnormal}$. The signaling load, and processing load of cryptographic techniques are given by $C_\beta{}^{(s)}(l)$, and $C_\beta{}^{(p)}(l)$ respectively. The values of these parameters are dependent on β, and l. The arrival rate of transactions from the user type β is defined as λ_β.

For convenience of analysis, we define a set of cost parameters as shown in the Table 4.

Table 4: Authentication cost parameters

Symbol	Description
c_s	Transmission cost on one hop
c_p	Encryption/decryption cost on one hop
c_v	Verification cost at an authentication server
c_{us}	A pair of encryption and decryption cost for a value
c_g	Key generation cost
c_{ts}	Transmission cost for a key to other communication identities

The transmission costs $C_\beta{}^{(s)}(l)$, can be derived using the message transition diagrams in Figures 3(a) - (c), as follows

$$C_\beta{}^{(s)}(l) = m_{\beta,l} c_s.$$

Where $m_{\beta,l}$ is the number of hops by which the entire authentication process passes for a particular type of user β, and the particular transaction sensitivity level-l. When l=0, all the type of users transactions requires $4N_h$ hops, and when $l > 0$ additional $k * 2N_h$ hops are required for transmitting the challenge, and receiving the response, where k is the number of times the challenge is generated.

Similar to the analysis of $C_\beta{}^{(s)}(l)$, by using message transition diagrams in Figures 3(a) - (c), the $C_\beta{}^{(p)}(l)$ is

$$C_\beta{}^{(p)}(l) = \vec{n}_{\beta,l}.\vec{x}_p.$$

Where, $\vec{x}_p$ is a vector defined as; $\vec{x}_p^T = [c_p, c_v, c_{us}, c_g, c_{ts}]$ and $\vec{n}_{\beta,l}$ is the vector denoting the corresponding number of costs to be considered during one authentication. The vectors $\vec{n}_{1,0} = \vec{n}_{2,0} = \vec{n}_{3,0} =$ [0,0,0,0,0], indicates for level-0 transactions there is no additional processing cost. The vectors $\vec{n}_{1,1} = \vec{n}_{2,1} = \vec{n}_{3,1} =$ [0,1,0,0,0], one verification cost at the TBAS is involved for level-1 transactions if a MU is *NORMAL;* otherwise, [0,k,0,0,0], k number of verification costs are involved, where k is the number of times the challenge is generated. The vectors $\vec{n}_{1,2} = \vec{n}_{2,2} = \vec{n}_{3,2} = \vec{n}_{1,3} = \vec{n}_{2,3} = \vec{n}_{3,3} = [4N_h,1,3,1,1]$, for level-2, and level-3 transactions if a MU is *NORMAL;* otherwise, it is $[k * 4N_h, k, 3k, 1, 1]$.

6 Simulation and Results

6.1 Simulation Environment

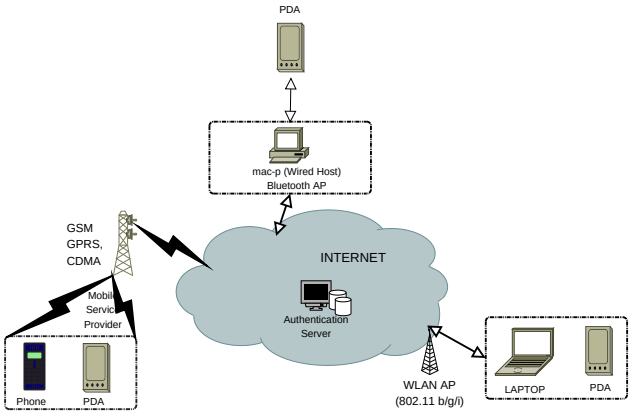


Figure 4: Hybrid wireless network testbed

The proposed authentication scheme has been tested on hybrid wireless testbed shown in Figure 4. Different types of mobile devices used in testbed includes Samsung X10 Laptop with 802.11b/g WiFi connectivity, HP iPAQ rx3715 PDA with Bluetooth, IEEE 802.11b, and IrDA connectivity, HP iPAQ h6365 PDA with IEEE 802.11b, Bluetooth, and GSM/GPRS connectivity, and CDMA enabled mobile phone. The Cisco Access Point AiroNet 1200 series gateway is used for wireless networks, and one of the local CDMA/GSM mobile service for cellular networks. The proposed system is very much consistent with many wireless networks such as Mobile IP, and WLAN's, which guarantees that our scheme is applicable to a realistic mobile environments.

6.2 Simulation Procedure

A mobile service having 30 different types of transactions which are distributed among various authentication levels are simulated, some of the example transactions are: *Requesting for a product technical information; Requesting for customers feedback; Requesting for discount policy; Filling up the purchase order; Submitting card details; and so on.* The belief database is established for chosen 100 mobile users out of population of customers, who are using the mobile service. This choice is made such a way that each 10 customers represents a big group which commonly exhibit similar type of behaviors, therefore we have 10 such groups who have heterogeneous behaviors, and observations. The authentication database is created with all the necessary attributes. The normal mobile transaction scenario between mobile user, and the authentication server has been simulated first, in which the mobile user connects to the SCA through the MCA for transaction executions. In the normal scenario the authentication challenges, and transaction based challenges have been generated over the changes in sensitivity levels of transactions, and user behaviors. In our attack model, we have injected the attack traffic into the stream of mobile transactions, by interrupting the session, intercepting, and modifying the transactions; by varying the values of temporal, and symptomatic parameters of transactions.

6.3 Results and Discussion

The effects of traffic pattern on the authentication delays at different sensitivity levels of transactions are demonstrated in Figure 5. It is observed that the delays are proportional to the transactions arrival rate λ_u, since the variables λ_l (l =0,1,2,3) are proportional to λ_u. Higher the sensitivity of transactions more is the authentication delay.

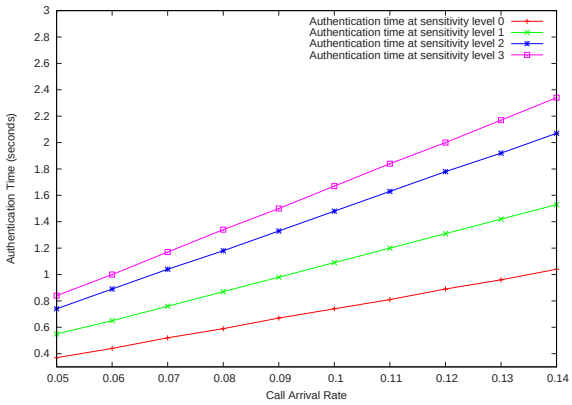


Figure 5: Authentication time vs. call arrival rate

With marginal addition of authentication delay, the TBAS detects many of the application level attacks which goes undetected under the regular mobile IP (MIP) schemes. Some of the simulated attack scenarios, and the corresponding results from the TBAS, and the MIP schemes are given as follows.

- **Scenario 1:** The attacker has stolen authentication identifiers of a MU by successfully executing identity theft attacks, and using them to obtain the service.

  MIP: Successfully authenticates the attacker.

  TBAS: Authenticates the attacker, until his/her transactions become suspicious, then authentication challenges are dynamically created based on changes in sensitivity level of the transactions, and beliefs.

- **Scenario 2:** The attacker is executing modification attack, by changing the contents of the transactions.

  MIP: No means to analyse transaction sensitivity levels, successfully authenticates the attacker.

  TBAS: Changes in transaction sensitivity levels are analyzed, and corresponding authentication challenges are created dynamically before committing the transaction.

- **Scenario 3:** Changes in attacker behaviors, for example, change in behaviors from normalcy to urgency.

  MIP: No means to recognize the changes in user behaviors, therefore attack becomes successful.

  TBAS: The user behavior analysis produces belief on urgency, which leads to high belief deviation factor, as a result authentication challenges are created dynamically.

The effects of traffic pattern on the authentication cost at different sensitivity levels of the transactions are demonstrated in Figure 6. It is observed that the cost is proportional to the transactions arrival rate λ_u, since the variables λ_l (l =0,1,2,3) are proportional to λ_u. The encryption/decryption cost on one hop C_p, and key generation cost C_g are assumed to be the lightest load compared to other costs. The values of other costs are determined by comparing to C_p, and C_g with the time to finish the operation.

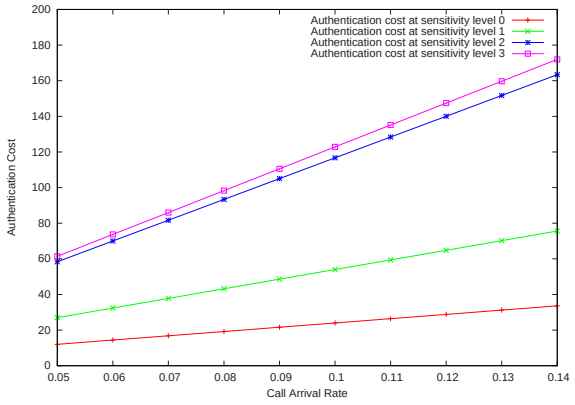


Figure 6: Authentication cost vs. call arrival rate

The transactions of various sensitivity levels are simulated randomly to test, the generation of required authentication levels by the proposed system. The generated authentication levels by the system is shown in Figure 7, for a stream of 50 transactions conducted in a particular session. For the experimentation purpose, we have varied the data, device, and location sensitivities. The Table 5 shows some of input instances used for simulation experiment.

The plot given in Figure 8, shows three sample cases of variations in deviation factors computed by the SCA. All the transactions of user1 are inside the normal range. The user2 transactions enters into suspicion range, and return back to normal range on successful answering of authentication challenge. This effect is due to decrementing the CDF by the value of current belief deviation factor (DF) generated by the *belief analyzer*. For example, if the current CDF is 0.45, and new DF is 0.1, then the new CDF become 0.55 (i.e.,0.45 + 0.1); as a result of this transactions become suspicious, an authentication challenge is generated by the *challenge generator*. On successful answering of the challenge the CDF return back

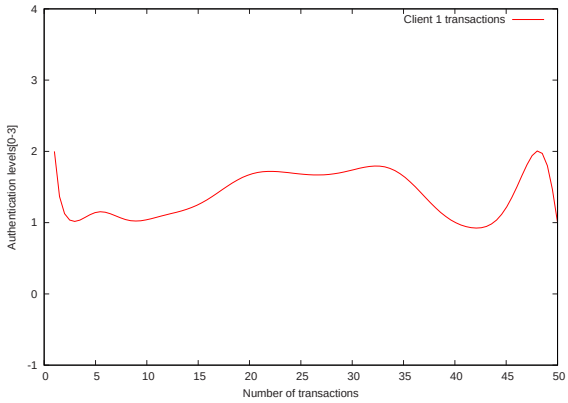


Figure 7: User transactions vs. authentication levels

to 0.45 (i.e., 0.55 - 0.1). Whereas the user3 transactions first enters into suspicion range, and further into abnormal range. This is due to failure in answering authentication challenge by a MU, as a result of this the CDF further increases, and cross threshold of suspiciousness.

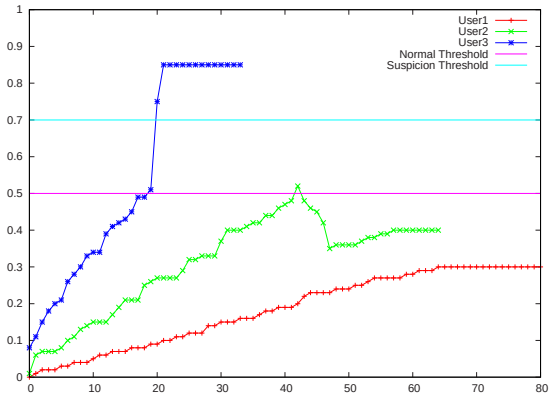


Figure 8: User transactions vs. belief deviation factor

Another result of our simulation experiment is shown in Figure 9, which indicates for the population of users, the authentication system has least number of false negatives, during classifying a MU as legitimate or illegitimate. This result is based on belief creation, and analysis by CAs. The false negatives indicates number of genuine MU's who have been identified as fake due to variations in client behaviors. The consequence of reduced number of false negatives is the genuine user will face less number of authentication challenges while performing transactions.

6.4 Attacks Detection

We explain how the TBAS addresses some of the possible attacks aimed at client transactions. The parameters used for generating observations, and beliefs by the MCA are listed below.

- $D_{\{A,T_1\}}$: The time delay between end of initial authentication, and beginning of first transaction of the session.

Table 5: Sample instance of transaction sensitivity variations

Data Sensitivity	Device familiarity	Location of operation	Auth. level produced
Personal: Public	Frequently used device	Home Network	0
personal: Private	Known device	Foreign Network	1
Personal: Public	New device	Known Foreign Network	1
Personal: Private	Frequently used device	Home Network	0
Personal: Financial	Known device	Home Network	1
Personal: Financial	New device	Unknown Foreign Network	2

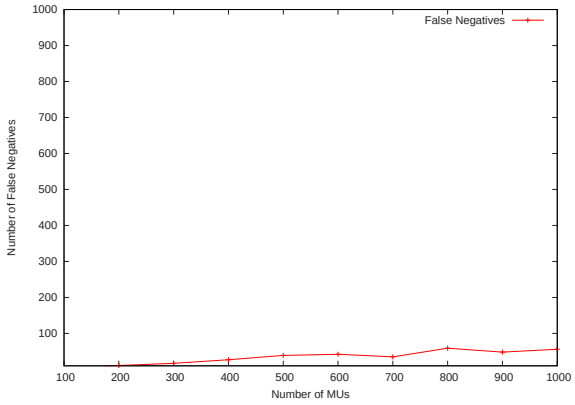


Figure 9: False negatives

- $D_{\{T_i,T_{i+1}\}}$: The time delay between two successive transactions of the session, namely T_i, and T_{i+1}.
- $W_{initial}$: This is the maximum idle time of the MCA before receiving the first transaction of the session.
- W_{trans}: This is the maximum idle time of the MCA between two successive transactions of the session.
- $E[D_{\{T_i,T_{i+1}\}}]$: The expected delay between two successive transactions.
- $\sigma(D_{\{T_i,T_{i+1}\}})$: The standard deviation of delay between two successive transactions.
- T_{type}: The type of current transaction, e.g., {*0:Read, 1:Write, 2:Delete, 3:Modify, ...*}.

Client transaction interruption:

The MCA generate belief on a MU transactions as *sluggish*, when $D_{\{A,T_1\}} > W_{initial}$ or $D_{\{T_i,T_{i+1}\}} > W_{trans}$. The SCA finds amount of deviation between existing belief from beliefs database, and a new belief. As a proactive measure the MCA sends an alert to a MU requesting for re-initiation of a transaction. If there is no response from a MU, and the deviation cross threshold. The SCA declares a MU interruption attack.

Client transaction modification:

We shown here how the modification attack is identified by the SCA. The MCA generate belief on a MU transactions as *passive*, when $D_{\{A,T_1\}} > W_{initial}$ or $D_{\{T_i,T_{i+1}\}} > W_{trans}$, and $(D_{\{T_i,T_{i+1}\}} > E[D_{\{T_i,T_{i+1}\}}])$. If the established belief on the client is different from new belief then the SCA temporarily buffers all the successive transactions of the session as a proactive measure. If the CDF crosses $Th_{suspicious}$ then the agent constructs a challenge based on the information from suspected transaction, and available challenge/response data. When an attacker receives challenge, he has to modify the challenge, and send that to client for response. This process requires considerably a long time which leads to detection of interruption attack. If the attacker sends that challenge without modification, then the client will come to know about suspicion, and he/she aborts transaction. Otherwise, if the attacker answers the challenge wrongly, then a modification attack is identified, the SCA sends modification attack identified communication to server. If the challenge was answered properly then all the buffered transactions are sent to the server for commitment. A modification attack detection result on transactions is simulated on two MU's named as *User-1*, and *User-5*, is shown in Figure 10.

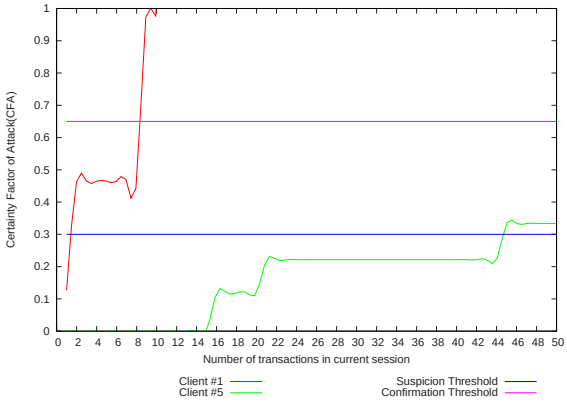


Figure 10: Modification attack detection results

Client transaction fabrication:

One of the method of identifying the fabrication attack is by using transaction type parameter. These at-

tacks could be detected by observing orthogonal changes in values of parameters which are representing type of transaction. The MCA generates belief on a MU as *casual*, if ($D_{\{A,T_1\}} < W_{initial}$ or $D_{\{T_i,T_{i+1}\}} < W_{trans}$, and ($\psi({T_{type}}^i, {T_{type}}^{i+1}) >$ Allowed-value). Where $\psi()$ is the deviation function to identify an orthogonal changes in type of transactions going on. If the established belief on a MU is different from new belief; then the agent temporarily buffers all the successive transactions of the session as a proactive measure. If the CDF crosses $Th_{suspicious}$ in the later stage of the session then the agent generates a challenge to detect fabrication attack.

7 Conclusion

The TBAS using cognitive agents is the new thinking towards authenticating the mobile user, and his/her transactions. The scheme is *intelligent* due to employing cognitive science approach, and *dynamic* using changing authentication requirements based on the sensitivity of transactions. We strongly feel that the rational approach towards authentication will address many existing weaknesses of conventional approaches of authentication. We could able to effectively identify those transaction-based attacks which are difficult to determine in conventional authentication schemes. The TBAS model could be further extended by incorporating in detail various handoff scenarios, and by performing detailed analysis on performace.

References

[1] N. G. Aghaee, and T. I. Oren, "Effects of cognitive complexity in agent simulation: Basics," *The SCSC 2004-Summer Computer. Simulation Conference*, pp. 15-19, 2004.

[2] P. S. Alvirado, C. L. Roncancio, and M. Adiba, "Analyzing mobile transactions support for DBMS," *The 12th International Workshop on Database and Expert Systems Applications*, pp. 595-600, 2001.

[3] P. G. Argyroudis, R. Verma, H. Tewari, and D. O. Mahony, *Performance Analysis of Cryptographic Protocols in Handheld Devices*, The Technical Report TCD-CS-2003-46, University of Dublin, 2003.

[4] B. S. Babu, and P. Venkataram, "Transaction based authentication scheme for mobile Communication: A cognitive agent based Approach," *The 3rd International Workshop on Security in Systems and Networks (SSN 2007), Conjunction with IPDPS 2007*, pp. 1-8, 2007.

[5] M. Baudet, "Security of protocols against guessing attacks," *SECSI '05*, pp. 16-25, 2005.

[6] D. P. Benjamin, R. S. Iyer, and Archana Perumal, "VMSoar: a cognitive agent for network security," *SPIE*, vol. 5812, pp. 72-80, 2005.

[7] E. Bertino, S. Jajodia, L. Mancini, and I. Ray, "Advanced transaction processing in multilevel secure file stores," *IEEE Transactions on Knowledge and Data Engineering*, vol. 10, no. 1, pp. 120-135, 1998.

[8] K. Boudaoud, H. Lubiod, R. Boutaba, and Z. Guessoum, "Network security management with intelligent agents," *The Network Operations and Management Symposium NOMS 2000*, pp. 579-592, 2000.

[9] A. Boukerche, and S. M. Mirela, "Behavior-based intrusion detection in mobile phone systems," *Journal of Parallel and Distributed Computing*, vol. 62, no. 9, pp. 1476-1490, 2002.

[10] J. Burke, B. Hartselle, B. Kneuven, and Bradley Morgan, "Wireless security attacks and defenses," *Window Security*, pp. 1-12, 2006.

[11] S. S. Chan, X. Fang, K. Brzezinski, Y. Zhow, S. Xu, and J. Lam, "Usability for mobile commerce across multiple form factors," *Journal of Electronic Commerce Research*, vol. 3, pp. 187-199, 2002.

[12] H. Chen, and T. V. L. N. Sivakumar, "New authentication method for mobile centric communications," *The IEEE 61st conference on Vehicular Technology*, pp. 2780-2784, 2005.

[13] Deloitte, *2006 Global Security Survey*, *Deloitte Inc*, 2006.

[14] A. I. Gardezi, *Security in Wireless Cellular Networks*, 2006.

[15] V. Gupta, S. Gupta, and S. Chang, "Performance analysis of elliptic curve cryptography for SSL," *The ACM Workshop on Wireless Security*, pp. 87-94, 2002.

[16] L. S. He, and N. Zhang, "An Asymmetric Authentication Protocol for M-Commerce Applications," *The 8th IEEE International Symposium on Computers and Communication (ISCC'03)*, vol. 1, pp. 244-250, 2003.

[17] J. Ho, and I. Akyildiz, "Mobile user location update and paging under delay constraints," *Wireless Networks*, vol. 1, no. 4, pp. 413-425, 1995.

[18] (http://sourceforge.net/projects/agentfactory)

[19] M. N. Huhns, and M. P. Singh, "Cognitive agents," *IEEE Internet computing*, pp. 87-89, 1998.

[20] K. Hwang, "Wireless PKI and distributed IDS for securing intranets and M-commerce," *The IEEE Third International Conference on Parallel and Distributed Computing, Applications, and Technologies (PDCAT 2002)*, pp. 1-16, 2002.

[21] W. Jansen, and T. Karygiannis, *NIST Special Publication 800-19-Mobile Agent Security.*

[22] C. M. Jonker, J. Treur, and W. Vries, "Temporal analysis of the dynamics of beliefs, desires, and intentions," *The Cognitive Science Quarterly (Special Issue on Desires, Goals, Intentions, and Values: Computational Architectures)*, vol. 2, pp. 471-494, 2002.

[23] S. T. Kent, and L. I. Millett, *Who Goes There?: Authentication Through the Lens of Privacy*, The National Academies Press, 2003.

[24] H. Kim, and H. Afifi, "Improving mobile authentication with new AAA protocols," *The IEEE International Conference on Communications*, vol. 1, pp. 497-501, 2003.

[25] H. N. Le, and M. Nygaard, "Mobile transaction system for supporting mobile work." *The 16th Database and Expert Systems Applications*, pp. 1090-1094, 2005.
[26] B. Lee, T. Kim, and S. Kang, "Ticket-based authentication and payment protocol for mobile telecommunications systems," *The International Symposium on Dependable Computing*, pp. 218-221, 2001.
[27] E. Nielsen, and S. Jacobs, *A security Model Supporting the Legacy UserID: Passphrase the Authentication Model that Won't Go Away!*, 2002.
[28] S. N. Panduranga, "Simplifying mobile commerce through a trusted transaction broker," *The IEEE International Conference on Personal Wireless Communications*, pp. 267-271, 2005.
[29] C. Perkins, and P. Calhoun, *Mobile IPv4 Challenge/Response Extensions*, RFC3012, 2000.
[30] A. S. Rao, and M. P. Georgeff, "Modeling rational agents within a BDI-architecture," *The Knowledge Representation and Reasoning*, pp. 473-484, 1991.
[31] A. S. Rao, and M. P. Georgeff, "An abstract architecture for rational agents," *The Knowledge Representation and Reasoning*, pp. 439-449, 1992.
[32] A. S. Rao and M. P. Georgeff, "BDI agents: From theory to practice," *The First International Conference on Multi-Agent Systems (ICMAS-95)*, pp. 312-319, 1995.
[33] D. Rosenthal, and F. Fung, "A test for non-disclosure in security level translations," *The IEEE Symposium on Security and Privacy*, pp. 196-206, 1999.
[34] S. Shieh, F. Ho, and Y. Huang, "An efficient authentication protocol for mobile networks," *Journal of Information Science and Engineering*, vol. 15, pp. 505-520, 1999.
[35] T. Shimoda, *A Theory Belief Model for Cognitive Agents*, Colorado State University.
[36] S. Sutikno, and A. Surya, "An architecture of F(22N) multiplier for elliptic curves cryptosystem," *The ISCAS 2000 on Circuits and Systems*, vol. 1, pp. 196-206, 2000.
[37] U. Varshney, "A framework for managing transactions in group-oriented mobile commerce services," *The 39th Annual Hawaii International Conference on System Sciences*, pp. 22.2, 2006.
[38] J. Veijalainen, V. Terziyan, and H. Tirri, "Transaction management for m-commerce at a mobile terminal," *The 36th Annual Hawaii International Conference on System Sciences*, pp. 89.1, 2003.
[39] P. Waters, and A. Walter, "Trusted transactions in a mobile environment," *The 4th International Conference on 3G Mobile Communication Technologies*, pp. 359-363, 2003.

Sathish Babu B. received his B.E., and M.E. degrees in Computer Science Engineering from Bangalore university, India. He has more than 10 years of teaching experience in engineering. From 2005 he is a research scholar in PET unit, department of Electrical Communication Engineering in Indian Institute of Science, Bangalore, India. His research interests includes intrusion, and fraud detection systems for mobile commerce environment, mobile communication security, and application of cognitive agents in dynamic transaction-based authentication, and security system for mobile communications. He has five international conference papers, and three journal papers in his credit.

Pallpa Venkataram received his Ph.D degree in Information Sciences from the University of Sheffield, U.K.in 1986. He is currently a Professor of Electrical Communication Engineering with Indian Institute of Science, Bangalore, India. Prof. Pallapa's research interests includes protocol engineering, wireless networks, network management, computational intelligence applications in communication, mobile computing security, and multimedia systems. He is a Fellow of IEE (England), Fellow of IETE(India), and a Senior member of IEEE Computer Society. Dr. Pallapa is the holder of a distinguished visitor diploma from the Orrego University, Trujillo, Peru. He has authored three books, and published over 150 papers in International/national Journals/conferences.