DESIGN OF A MONITOR FOR DETECTING MONEY LAUNDERING AND TERRORIST FINANCING

1TAMER HOSSAM HELMY, 2MOHAMED ZAKI, 2TAREK SALAH, 4KHALED BADRAN
1,3,4Computer Engineering Department, Military Technical College, Cairo, Egypt
2Systems and Computer Department Al Azhar University, Cairo, Egypt

E-mail: 1tamer_hossam2005@yahoo.com, 2azhar@eun.eg, 3tarekbox2000@gmail.com, 4khaledbadran@hotmail.com

ABSTRACT

Money laundering is a global problem that affects all countries to various degrees. Although, many countries take benefits from money laundering, by accepting the money from laundering but keeping the crime abroad, at the long run, “money laundering attracts crime”. Criminals come to know a country, create networks and eventually also locate their criminal activities there. Most financial institutions have been implementing anti-money laundering solutions (AML) to fight investment fraud. The key pillar of a strong Anti Money Laundering system for any financial institution depends mainly on a well designed and effective monitoring system. The main purpose of the Anti Money Laundering transactions monitoring system is to identify potential suspicious behaviors embedded in legitimate transactions. This paper presents a monitor framework that uses various techniques to enhance the monitoring capabilities. This framework is depending on rule base monitoring, behavior detection monitoring, cluster monitoring and link analysis based monitoring. The monitor detection processes are based on a money laundering deterministic finite automaton that has been obtained from their corresponding regular expressions.

Keywords: Anti Money Laundering system, Money laundering monitoring and detecting, Cycle detection monitoring, Suspected Link monitoring.

1. INTRODUCTION

Money is laundered to conceal criminal activities generated it. These crimes involved drug trafficking, illegal tax avoidance, terrorist activities, and any illegal crimes [1]. Money laundering is the process of illegal activities by which criminals and terrorists disguise the illegal origin and illegitimate ownership of property and assets that are results of their criminal activities. Such process means that “dirty money” is transformed into “clean money,” taking into account hiding the criminal origin and making it difficult to be traced. If money laundering processes are completed successfully, it allows criminals to represent their proceeds of their crime as having a legitimate source, and thereby maintain control over those proceeds and dispose of them without hindrance, which is the goal of money laundering crime [2].

Money laundering process involves three steps [3], first, “placement”, which is a process for transferring the proceeds from illegal activities into the financial institutions or converting cash into instruments. Second, “layering”, this is a process in which a launderer may conduct a series of financial transactions to distance the proceeds from illegal source. Third, “integration”, which is the final stage in the laundering process, illicit fund is integrated with monies from legitimate commercial activities as they enter the mainstream economy. The integration of illicit monies into a legitimate economy is very difficult to detect unless an audit trail had been established during the previous stages.

Many financial institutions deployed anti money laundering detection solutions. These solutions even belong to first generation solution (rule base approach) or second generation solution (risk base approach) [5]. The first generation solution depends mainly on establishing fixed rules based on monetary threshold and detecting specific transactions patterns. This approach cannot detect money laundering cases which include amount
less than the established threshold (cannot detect
smurfing scenarios under the established threshold).
Furthermore, rule base approach generates many
false positive alarms, as it alerts transactions over
threshold and marks them as suspicious while they
do not represent any existing risk [6].

Many financial institutions began to recognize
the need to overcome the ineffectiveness of the
rules-based regulations. The Financial Action Task
Force FATF recommendations distinguished risk as
an important feature to both international and
national levels of regulation. The European Union
(EU) shifted the regulatory framework from the
rules-based to a risk-based approach (RBA). This
approach is depending on four components [7].

- Client risk assessment, which is used to collect
  all the transactions and information gathered about
  all customers to investigate the customer risky.
- Transaction risk measurement, which is used to
detect accounts related to transactions that lead to
potential money laundering risk.
- Behavior detection technology, which is used to
detect suspicious pattern in the transactions that
leads to potential money laundering risk.
- Workflow and reporting tools, which are used
to alert the investigators about money laundering
risks and generate reports about these risks.

In this paper we discuss a monitor framework that
can combine the two previous approaches (rule base
and risk base). It utilizes many detection techniques
to facilitate and enhance the detection process and
implemented customer due diligence requirements
to better understand who are your customers and
understand their transactions that let you precisely
define the customer risk. Detection techniques are
rule base, feature detection, clustering, and link
analysis based monitors (cycle detection and link
suspected). Cycle means that dirty money
transferred from the origin to many other accounts
through many transactions to disguise the money
origin and then the money returned back to that
origin. Link suspected means that there is a link
between a suspected person and other person. This
link can be detected even it is indirect link (many
accounts separate between the target accounts). This
module can be also used if financial institutions
declared list of suspected links. For example, links
between persons whom work in vital positions and
others whom work in commercial companies or
foreign people. The cycle and link suspected
patterns can be difficulty discovered manually by
the investigators. We represent an automatic way to
discover these patterns using the aid of link analysis

statistics. Moreover, the proposed framework
exploits finite state automata to enhance the monitor
detection capabilities and can be obtained from
corresponding regular expressions in its detection
processes.

This paper is organized as follows: Section 2
discusses the related work. Proposed monitor
framework is presented in section 3. Also, section 4
introduces suspicious pattern corresponding finite
state automata and its regular expression. Case
study is presented in section 5. The concluding
remarks are summarized in section 6.

2. RELATED WORK

There are many anti money laundering detection
approaches that have been published in the last
years, however building an efficient and powerful
anti money laundering system still a great challenge
that faces the financial institutions to analyze and
detect suspicious transactions among a huge
number of transactions carried out every day. In [8]
FinCEN Artificial Intelligence System (FAIS)
which integrates intelligence and a software agent
in a cooperative discovery task on a very large
data space. It implements AI technologies
including rule base reasoning and underlying
database which plays as a blackboard and uses for
analysis, interactive queries, and visualization. FAIS
integrates NetMap link analysis package for
visualization.

In [9] a new cross outlier detection model was
implemented. Its approach based on distance
definition incorporated with the financial
transactions data features. The author decides to
compare each transaction with its participant
accounts’ history and also compares it with a peer
group to detect any unusual behavior and minimize
the false positive rate. Peer group analysis concept
is largely depending on cross datasets outlier
detection model. An approximation algorithm
accompanied with the model to optimize the
computation of the deviation between the tested
data point to the reference dataset.

Also in [10] a decision tree method is used to
create the determination rules of the money
laundering risk by using four attributes from the
customer profiles. These attributes are business and
entity risk, location, business size, and product risk.
In order to demonstrate the decision tree learning,
the attributes are listed in three ranks, which are
‘low’, ‘middle’ and ‘high’. A decision tree can be
viewed as a partitioning of the instance space. Each partition, called a leaf, represents a number of similar instances that belong to the same class. The split points are chosen according to the most informative attributes of the data instances. Rules are thus can be extracted from the root to some leaf node.

In [11] Intelligent Money Laundering Monitoring and Detection Process system (MLMDP) is discussed. It provides intelligent agent oriented solution for money laundering monitoring and detection process based on Simons’ decision making theory. According to Simons’ decision making theory MLMDP’s framework identifies four different phases – Intelligence, Design, Choice, and Review. [A. Lekhac, et al., 2009] have defined a set of parameters for data mining and applied them to determine a relevant threshold and to detect suspicious transactions. The authors define two parameters Δ1 which is the proportion between the redemption value and the subscription value conditional on time, and Δ2 which is the proportional between a specific redemption value and the total value of the investors’ shares conditional on time. A clustering technique (K-Means family, for instance) is applied for each Δ1 and Δ2 at both two levels: fund and investor. These outputs will be then used to feed into a back propagation based neural network for training on suspicious and non suspicious cases. These results are then stored in a knowledge-base to help the investigators to take decision.

In [12] two main components: a detection component and a result evaluation component are implemented. The detection component provides the functionality to detect and retrieve hot spots in the data. The detection algorithm can be configured and adapted. An evaluation component, which relies heavily on visualization, was developed. Graphs and sub-graphs are identified to perform clustering. Pattern matching algorithms were proposed to detect weather grouped sub-groups are suspicious. The approach allowed detecting transaction chains and smurfing. The insights from evaluation can be used to reconfigure the detection component, allowing a refining and learning process.

3. PROPOSED MONITOR FRAMEWORK

This framework, Figure 1 is responsible for monitoring the transactions and detecting any transaction that may violate one or more rules or may have suspected behavior hidden in other legitimate transactions. Moreover it can obtain the customer risky for each transaction and the transaction risky for suspected transaction. That framework can be explained as follows

- The transaction passes through the monitoring phase which contains many modules such as rule base, feature detection, cluster, cycle detection, and suspected link monitors.
- The proposed monitor checks the transaction participant risky from the customer risky and returns with a severity degree corresponding to the customer risky.
- The transaction is subjected to many monitoring modules as discussed. If any of these monitor’s modules detect that the underline transaction is suspected, the transaction marked as suspected and obtained a risky score from the transaction risky module.
- The suspected transactions are displayed in a tabular form according to their corresponding risky score.
- The investigator uses interactive queries and visual link analysis to decide which transactions are suspected and which are normal

A. Risky Mechanisms

The framework proposes two risky mechanisms, the customer risky and the transaction risky. The risky mechanism can be used to detect the customer risky degree and depends mainly on Customer Due Diligence. The transaction risky depends on likelihood approach that can be used to get a severity degree for each suspected transaction.
1) Risky Customer

The Money Laundering Regulations 2007 permit a risk based approach to compliance with Customer Due Diligence obligations. Regulation No 5 says that, customer due diligence means:

- **Know Your Customer (KYC):** “Understand who are your customers and what they do throughout the relationship with them”.

- **Know the Transactions of Your Customers (KTYC):** “Understand the transactions of the customer and have systems to spot suspicious activity”

- **Know the Customers of the Customer (KCYC):** “That allows for an extra level of the KYC process”

- **Know Your Business Partners (KYBP):** “Understanding those that institutions work with to avoid institutions that can be involved in unwanted activities”

- **Know Your Employees (KYE):** “Criminal organizations need employees in the financial institutions to support them with illegal activities”

A financial institution needs to obtain proof of identity of its customers, especially new customers opening accounts. The customer names and addresses are compared to sanctions lists of suspicious or designated persons that are regularly produced by government agencies. This knowledge effectively allows for developing risk profiles for customers. These profiles will facilitate the identification of any account activity that deviates from activity or behavior that would be considered “normal” and could be considered as potentially suspected or unusual.

2) Risky Transaction

The risk scoring module can rank suspected transactions that match a suspected behavior or violate some rule(s). Such rank is represented by a numeric value that simulates the severity degree for this suspiciousness. The risk scoring is measured in terms of the chance (likelihood) of this risk and the severity/amount of damage (impact) if this risk occurred. The risk scores are the product of the likelihood and the impact. They could be obtained from Table 1. For AML, Table 1 is exploited to obtain Table 2 that represents the money laundering risk scoring table, where Low =1, Moderate =2, High =3, and extreme =4.

<table>
<thead>
<tr>
<th>Risk Scale (severity)</th>
<th>Negligible</th>
<th>Marginal</th>
<th>Critical</th>
<th>Catastrophic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certain</td>
<td>High</td>
<td>High</td>
<td>Extreme</td>
<td>Extreme</td>
</tr>
<tr>
<td>Likely</td>
<td>Moderate</td>
<td>High</td>
<td>Extreme</td>
<td>Extreme</td>
</tr>
<tr>
<td>Possible</td>
<td>Low</td>
<td>Moderate</td>
<td>High</td>
<td>Extreme</td>
</tr>
<tr>
<td>Rare</td>
<td>Low</td>
<td>Low</td>
<td>Moderate</td>
<td>High</td>
</tr>
</tbody>
</table>

### Table 1: The Risk Score Table

<table>
<thead>
<tr>
<th>Risk domain</th>
<th>Risk</th>
<th>Impact</th>
<th>Likelihood</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Risk</td>
<td>More than threshold amount</td>
<td>Marginal</td>
<td>Likely</td>
<td>High</td>
</tr>
<tr>
<td>Suspected sender/receiver</td>
<td>Critical</td>
<td>Possible</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Suspected account sender/receiver</td>
<td>Critical</td>
<td>Possible</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Unusual transactions</td>
<td>Critical</td>
<td>Possible</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Sender Smurfing</td>
<td>Catastrophic</td>
<td>possible</td>
<td>Extreme</td>
<td></td>
</tr>
<tr>
<td>Receiver Smurfing</td>
<td>Catastrophic</td>
<td>possible</td>
<td>Extreme</td>
<td></td>
</tr>
<tr>
<td>Divide transactions between multiple related accounts</td>
<td>Catastrophic</td>
<td>Possible</td>
<td>Extreme</td>
<td></td>
</tr>
<tr>
<td>Immediate withdrawal</td>
<td>Critical</td>
<td>Likely</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Dormant account suddenly activated</td>
<td>Critical</td>
<td>possible</td>
<td>High</td>
<td></td>
</tr>
<tr>
<td>Link suspected person</td>
<td>Catastrophic</td>
<td>Possible</td>
<td>Extreme</td>
<td></td>
</tr>
<tr>
<td>Link suspected account</td>
<td>Catastrophic</td>
<td>Possible</td>
<td>Extreme</td>
<td></td>
</tr>
<tr>
<td>Cycling</td>
<td>Catastrophic</td>
<td>Possible</td>
<td>Extreme</td>
<td></td>
</tr>
<tr>
<td>Country Risk</td>
<td>Transactions from suspected country</td>
<td>Critical</td>
<td>likely</td>
<td>High</td>
</tr>
</tbody>
</table>

### Table 2: The Risk Matrix

B. Monitoring Modules

The proposed monitor will be discussed and many monitoring modules such as rule base, feature extraction, clustering, cycle detection, and suspected link will be explained.

1) Rule Base

Rule base monitor: This monitor consists of a rule (fact) base and an interpreter. When the interpreter receives a transaction, it examines it to determine whether that transaction may match a rule antecedent. If it matched then the rules consequent fires ( executes a corresponding action). It is obvious that this monitor relies upon direct search in the underlying database. As the monitor rules increases, its ability to discover suspected transactions increases. Rules are designed to monitor customer transactions. The rule base monitoring captures the business logic in the form of rules from AML practice, such as from the FATF 40 Recommendations [13]. The system uses rules of the IF-Then form such as:

- IF a customer OR a person publicly associated with the customer is on any of the Sanctions
Lists, THEN this customer and associated transactions are recorded as possible money laundering.

- IF a customer’s account has wire transfers to or from a country identified as a high money laundering risk, THEN this customer and associated transactions are recorded as possible money laundering.

- IF a customer has wire transfer to or from any other account with amount exceeds the threshold amount which is defined by $10000, THEN this transaction is recorded as possible money laundering.

- IF a customer has wire transfer to or from account marked as suspected account, THEN this customer and associated transactions are recorded as possible money laundering.

2) Feature Detection
This monitor aims at detecting hidden feature(s) in customers' transactions. Such features could not be directly obtained. The corresponding extraction method searches various resources such as historical database, special purpose files and/or relevant financial documents to collect the required features. Since these features are invariant, then their correspondent patterns characterize the underlying banking process. The monitoring efficiency increases by detecting a feature repeatedly in similar situations. The more scenarios are implemented in the solution, the more detection power is gained. In fact, the feature detection monitoring implemented many scenarios that are pointed out in the following:

a) Nominees
Customers try to avoid writing a CTR report, so they divide a placement exceeding the threshold into many transactions to their related accounts (their wives or sons) [14]. In this case the link analysis technique is used to discover the related accounts for each customer and deals with these accounts as they are belonging to one person. It detects the relationship (link) by analyzing linking and matching the customers’ information (address and telephone number).

b) Smurfing (Structure)
Smurfing, or structure, is one of the most common money laundering methods, because it focuses on making funds untraceable through diversification [15]. Person tries to evade scrutiny from government agencies by breaking up a transaction involving a large amount of money into smaller transactions that are below the reporting threshold as shown in Figure 2. The smurfing module can monitor the customers’ transactions and combines transactions for each customer around the day, and then detects if there exist any transactions belongs to or send to dedicated account and these transactions amount exceeded the threshold.

3) Cluster Based Monitoring
Clustering technique is used to develop methods of looking for closely related groups of objects. Cluster analysis can be used to determine underlying groupings that are not apparent in the data. For example, cluster analysis of wire transfers could be based on the frequency and amount of each transfer, as well as the type of beneficiary and
his monthly income. Such analysis could reveal groups of transfers whose originators are highly similar (e.g., brokerage houses, industrial firms, or money transmitters).

Computational techniques for clustering can be found in [17]. In financial data, clusters might reveal similar types of accounts, individuals, or organizations. For example, customers’ accounts could be divided into personal account and companies’ accounts. Customers whose income is nearly equal might cluster closely together in comparison to other customers. Similarly, companies which are similar in financial type and has nearly equal income might resemble each other closely in terms of their financial transactions. These clusters might allow investigators to identify manufacturing firms whose financial transactions are typical and examine them more closely to determine whether the customer transaction is a usual transaction (lies in the amount criteria dedicated for his category) or unusual transaction (out of the boundaries of his category).

In the monitoring phase, the cluster based monitor returns the transaction amount and the transaction participants, and detects the categories which they are belonging to. If the transaction amount is more than the maximum transaction amount for the sender category or the receiver category, the monitor fires the unusual transaction alarm.

4) Cycle Detection Monitoring
Cycle means that the money transfer began at the source account and transfer through many other accounts (layering process) before it returns back again to the initiator after conceal the illegal origin and illegitimate ownership of property and assets that are the proceeds or results of their criminal activities [18].

Cycle detection is difficult to be discovered manually, so we implemented cycle detection algorithm by making use of link analysis to discover cycles in addition to firing cycle detection alarm to warn the investigator about this threat. Link analysis is used to identify relations between objects (accounts, customers). It uses transactions data between objects to restructure the data into a relational association matrix that can represent the relation between accounts and aids in constructing the link analysis Visualization interface.

Figure 3, represents the money flow diagram for six customers’ accounts. The association matrix for these customers can be represented in table 3. In that matrix, rows represent the senders of the money transferred while, columns represent the receivers. The element cells provide information about the relation between each pair of objects. The cell value is 1 if there is an association between the raw name and the column header and 0 otherwise.
### Function void createAssociationMatrix(acc1):

Begin

Select transactions T from transactions table Where from_account = acc1 or the receiver_account = acc1

T = 1, 2, 3, ......N where N is the number of returned transactions

Set i = 1

Repeat

If $T_i$ is not exist in the matrix table then

Insert transaction number, sender account, and receiver account into matrix table.

End if.

Set i = i + 1

Until i > N

Set i = 1

Initialize array V to zeros.

Repeat

If $T_i$ Sender account <> acc1 then

Put $T_i$ Sender account into V

End if

If $T_i$ receiver account <> acc1 then

Put $T_i$ receiver account into V

End if

Set i = i + 1

until i > N

Repeat

Get acc2 from V

Select count(*) X from matrix_table where sender account = acc2 or receiver account = acc2

Select count(*) R from transactions_table where sender account = acc2 or receiver account = acc2

If X < R then

call createAssociationMatrix(acc2)

End if

until empty V

End

### Function vector returnVertexData(account)

Begin

Select from_account, to_account, trans_ser from matrix_table where from_account = account or to_account = account.

Put the data into vector V.

Return V

End

### Function void linkAnalysis (account number)

Begin

Set acc1 = account number

Call createAssociationMatrix(acc1)

Initialize graph G to null

Add vertex acc1 to G

Vector V = Call returnVertexData(acc1)

While V is not null Do

Get value to_account from V

Add vertex to_account to G

If count > 0 then

Call graphCreation(to_account)

End if

Get another value to_account from V

End While

End

### Function void graphCreation(account)

Begin

Initialize vector V to zeros

V = returnVertexData(account)

While V is not null do

Get value to_account from V

If vertex to_account ¢ G then

Add vertex to_account to G

Add edge from_account and to_account

Set count = call returnAccountTrans(to_account)

If count > 0 then

Call graphCreation(to_account)  // (recursive)

End if

Get another value to_account from V

End if

Get value from_account from V

If vertex from_account ¢ G then

Add vertex from_account to G

Add edge from account and account

Set count = call returnAccountTrans(from_account)

If count > 0 then

Call graphCreation(from_account)  // (recursive)

End if

Else

Add edge from account and account

End if

End while

End

### Function int returnAccountTrans(account)

Begin

Select Count(*) count from matrix_table where from_account = account.

Return count.

End
Function void FindCycle(transaction T)
Begin
Set acc1 = T.receiverAccount
Call createAssociationMatrix(acc1)
Initialize array F2 which contains the current processing node and array F1 which contains the processed node to zeros.
Insert into Vector N values Select from_account From matrix_table.
Set i = 1
Repeat
  Clear all values in F2, P2
  Put node Ni into F2
  Set Path = null.
  Call reachability(Ni)
  Set i = i + 1
Until i = N.length
End

Function void reachability(Ni)
Begin
Insert into vector M Select to_account where from_account = Ni
Set j = 1
Repeat
  Set fromNode = Ni
  Set toNode = Mj
  If path = null then
    path = path + fromNode + "+" + toNode
  Else
    path = path + "+" + toNode.
  End if.
  Set newPath = path
  Insert into reachability table (from node, to node, path) values (fromNode, toNode, path).
  Select count(*) from matrix_table where from_account = Mj
  If count > 0 then
    If Mj \in F1 then
      F1.add(Mj)
      Call reachability(Mj) //recursive
    Else
      Insert into vector K Select to_account where from_account = Mj
      Set l = 1
      Repeat
        If Kj = Mj then
          Call reachability(Kj) //recursive
        End if.
        Set I = I + 1
        Until I = K.length
        Set path = newPath and
        F2.addNode(Mj).
      End if.
      If path = newPaths then
        F1.addNode(Mj).
      Else
        path = newPath
        F2.addNode(Mj).
      End if.
      Remove last node from the path
  Else
    End if.
  End if.
Until j = M.length
End

4) Suspected Link

Suspected link monitoring is concerned with uncovering hidden relationships between customers participating in the underlying transaction or layering transactions. Suspected link monitor can detect relations between customers and any other suspected person recorded in the sanctions list or informed by the government. This monitor can detect the relations between persons even if there are many people's accounts acting as pass-through accounts between these persons (connected through many other persons). Suspected link monitoring mechanism can also detect hidden feature represented by a relation between two customers that leads to high degree of suspicious. For example, transactions carried out between corporate owner and governmental leader. This monitor facilitates the financial institutions to add many pre-defined relations present suspected link relation, and provides the capability to detect these suspected relations at the run time.

These hidden relations are difficult to be discovered manually or by querying the database, so we implemented the link suspected monitoring mechanism which detects this link automatically using the aid of link analysis technique and fires suspected link threat. Suspected link module uses the link analysis and the reachability algorithms to detect links between transactions’ participants. The module monitors the transactions and returns the link path for each transaction from the reachability algorithm, and then returns the customer data for each account. After that the module compares the customer data obtained from the transaction with the suspected pre-defined relations described by the financial institution and fires suspected link detection alarm in case of match detected.

4. FINITE STATE AUTOMATA FOR MONEY LAUNDERING SCENARIOS

The proposed monitor works by making use of an automata approach [19]. Such automata are represented using Mealy notation [20]. Thus the monitor in its general form is represented by

\[ S = \langle Q, T, \delta, \eta, P \rangle \]

in which Q is a set of states, where the start state \( q_0 \in Q \). T is a finite set of input transactions, \( \delta \) and \( \eta \) are the next state and output mappings, respectively. The finite set P is the set of monitor outputs, \{\( P_k ; k = 1,2,\ldots,K \) or nil\}. For that automata a scenario can be represented by an input \( T_i^j; i = 1,2,\ldots,I \) where I is the number of
input transactions, and a monitor state is represented by $q_j$, $j = 1,2,…,J$ where $J$ is the number of states and $T \cap Q = \emptyset$. Table 4 provides a general 4-state example where its rows represents the states and the columns express the input. Any entry is denoted by: next state/identified problem, where the identified problem belongs to the monitor present output.

$\mathcal{A}_0 = q_0$: Start state.

$\mathcal{A}_1$: Monitor state after 1st monitored transaction.

$\mathcal{A}_2$: Monitor state after 2nd monitored transaction.

$\mathcal{A}_3$: Monitor state after 3rd monitored transaction.

The following example will be given to describe more specifically anti-money laundering key transactions. The example describes a finite state machine which is illustrated by Mealy transition diagrams, Figure 5, instead of Mealy transition tables. If $T$, $t$, $\bar{T}$ denote transactions of smurfing small amounts, immediate withdrawal, and cycling respectively. Smurfing for example, can be defined as one transaction or more followed by normal transactions or not, followed by one transaction or more under condition that the transactions carry the smurfing characteristics (multiple transactions produced from one account to multiple account or transactions from many accounts to one account). Then one can write the smurfing regular expressions as follow:

Smurfing = \begin{align*} T^+ N^* T^+ \end{align*}

Where $N$ denotes normal transaction, “$*$” means zero or any number of repetitions and “$+$” means one or more repetitions.

Similarly, other money laundering scenarios can be expressed as follows:

Immediate withdrawal = \begin{align*} T^+ N^* T^+ \end{align*}

Cycling = \begin{align*} T^+ N^* T^+ \end{align*}

The corresponding automata diagrams are shown in Figure 5, in which closure “$*$” is taken to be zero, for simplicity.

Figure 5: Money Laundering Cases Representations

Figure 6, represents the combined automata for the scenarios of money laundering under consideration. From that figure the generalized deterministic finite automata DFA, Table 5, can be obtained by making use of the subset construction algorithm given in [21]. In complicated practical situations, if that DFA recognized more than one attack at the same time, then the attacks should be prioritized in order to provide a means for conflict resolution.

Figure 6: Non Deterministic Finite Automata for Money Laundering Cases

5. CASE STUDY: UNTRACEABLE MONEY SOURCE

This case describes untraceable money donation to the Pennsylvania governor election campaign [22]. In 2010, a total amount of $1.5 million contribution arrived from Wisconsin to the campaign account in Pennsylvania. This amount of money is due to the fact that two big businesses “corporate” contribute $1.5 million to the campaign which is forbidden according to Pennsylvania law which prevents candidates from accepting corporate money, so the money could not be directly translated to the campaign account. The money translated from the corporate to the Republican Governors Association (RGA), then in the same day the money is translated to Wisconsin PAC, consequently, to Pennsylvania PAC and finally to the campaign account. Figure 7, describes the money flow diagram for that case.
The Monitor detection:

For convenience since the monitor is changeable with respect to time it will be represented by a sequential machine, Table 6.

From the monitor’s modules, it has been found out that transactions T2, T3, T4, and T5 are suspicious. Therefore, the detected suspicious transactions are listed according to their risky score and subjected to the investigator for more investigation.

6. CONCLUSION

This work proposes a monitor framework for anti money laundering systems. It provides an effective system for AML. It monitors transactions relying on various monitoring techniques, followed by a visualized link analysis that uses to strengthen the analyst belief. The proposed monitor framework combines the rule base and risk base approaches. It achieves the risk base approach by using customer profile and transaction risky score. It also has many monitoring modules that have the ability to detect many money laundering scenarios. Using Cluster module in the monitoring framework leads to decrease the false positive alarms that may exhaust money laundering investigators. The underlying monitor has been implemented successfully on several practical cases, one of them has been discussed here for illustration.

REFERENCES


### Table 4: Mealy Representation of a 4-State Example

<table>
<thead>
<tr>
<th>Input</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present State</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
</tr>
<tr>
<td>q1</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
</tr>
<tr>
<td>q2</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
</tr>
<tr>
<td>q3</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
</tr>
<tr>
<td>q4</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
</tr>
</tbody>
</table>

### Table 5: The Transition Table of the DFA

<table>
<thead>
<tr>
<th>Input</th>
<th>T</th>
<th>T</th>
<th>T</th>
<th>Crime found</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present State</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>q1</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
</tr>
<tr>
<td>q2</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
</tr>
<tr>
<td>q3</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
</tr>
<tr>
<td>q4</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
</tr>
</tbody>
</table>

### Table 6: The Monitor Finite State Machine

<table>
<thead>
<tr>
<th>Input</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present State</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>q1</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
</tr>
<tr>
<td>q2</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
</tr>
<tr>
<td>q3</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
</tr>
<tr>
<td>q4</td>
<td>T1</td>
<td>T2</td>
<td>T3</td>
<td>T4</td>
<td>T5</td>
</tr>
</tbody>
</table>