International Journal of Computer Networks and Applications
Volume 1, Issue 1, November - December (2014)

RESEARCH ARTICLE

Design of a Monitor for Detecting Money Laundering and Terrorist Financing

Tamer Hossam Eldin Helmy
System and Computer Engineering, Military Technical Collage, Cairo, Egypt
Tamer_hossam2005@yahoo.com

Mohamed zaki Abd-ElMegied
Systems and Computer Department, Al Azhar University, Cairo, Egypt
azhar@eun.eg

Tarek S. Sobh
Information System Department, Egyptian Armed Forces, Cairo, Egypt
tarekbox2000@Yahoo.com

Khaled Mahmoud Shafea Badran
System and Computer Engineering, Military Technical Collage, Cairo, Egypt
KhaledBadran@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.

Index Terms – 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, 4], 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 TATF 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 butter 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’s decision making theory MLMDP’s framework identifies four different phases – Intelligence, Design, Choice, and Review. In [12], has 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 in to 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 [13] 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. PORPOSED MONITOR FRAMEWORK

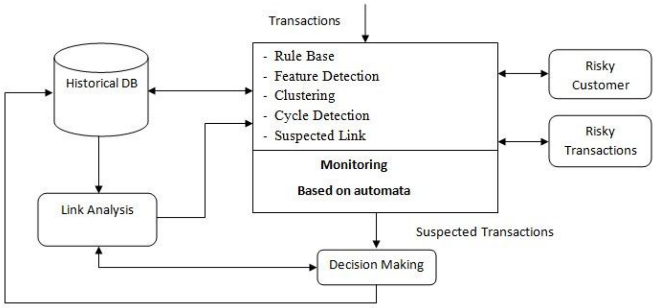


Figure 1 The Monitor Framework Architecture

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.

3.1. 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.

3.1.1.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.

3.1.2. 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 [14]. 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

		Risk Scale (severity)			
		Negligible	***Marginal***	***Critical***	***Catastrophic***
Likelihood	***Certain***	High	High	Extreme	Extreme
	Likely	Moderate	High	High	Extreme
	Possible	Low	Moderate	High	Extreme
	Rare	Low	Low	Moderate	High

Table 1 The Risk Score Table

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

Table 2 The Risk Matrix

3.2. 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.

3.2.1. 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 [15]. 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.

3.2.2. 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) [16].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 [17]. 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.

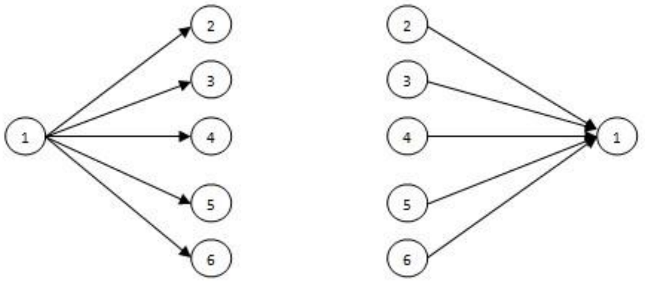


Figure 2 Smurfing Money Laundering Technique

C. Immediate Withdrawal

This module detects activities in an account when a large percentage of a deposit of cash is debited from the account in a short period of time and this transaction is not consistent with the customer's legitimate business needs. Thus, the system can record these transactions as suspected transactions.

D. Dormant Accounts

Dormant account means account that has no activity for a long time and suddenly, becomes active and carries out many transactions in a short time [18]. Dormant accounts have been used by terrorists to create a purported customer relationship, upon which additional frauds may be perpetrated. For example, a terrorist used a number of banks, holding an account in each of them contained a minimal sum, believed to be for two purposes: first, to keep the account open, and secondly, to ensure that undue attention was not drawn to it. At a strategic time, a transfer was received into the account, to enable the purchase of terrorist material. These transactions can be detected and recorded as suspected transactions.

3.2.3. 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 [19]. 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.

3.2.4. 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 [20]. 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.

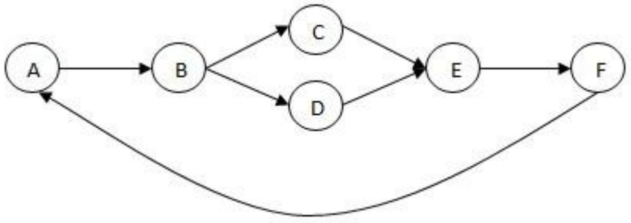


Figure 3 Money Flow Diagram Example

	A	B	C	D	E	F
A	0	1	0	0	0	0
B	0	0	1	1	0	0
C	0	0	0	0	1	0
D	0	0	0	0	1	0
E	0	0	0	0	0	1
F	1	0	0	0	0	0

Table 3 The Association Matrix

For convenience, association matrix, link analysis and cycle detection algorithms are presented as follow:

```
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)
            (recursive function)
        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
        Add edge from vertex acc1and to_account
        Set count = call
        returnAccountTrans(to_account)
        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
    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 $P_1$ which contains the current processing node and array $P_2$ 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 $P_1$, $P_2$.
    Put node $N_i$ into $P_1$.
    Set Path = null.
    Call reachability($n_i$)
    Set i = i + 1
Until i = N.length
```

End

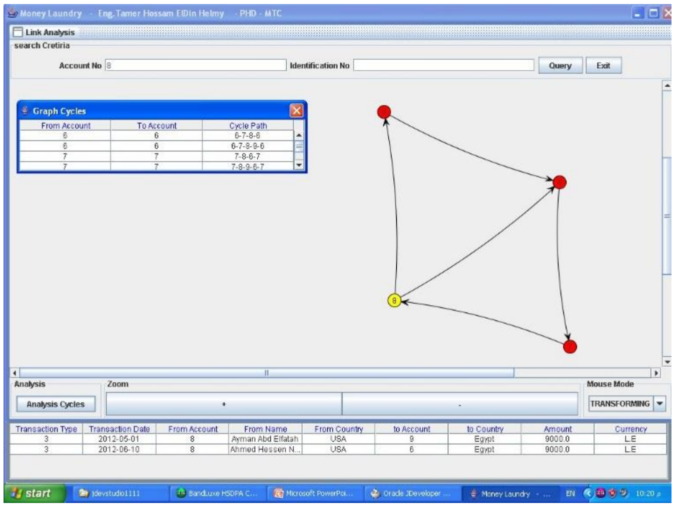
Function void reachability(n_i)

Begin

```
Insert into vector M Select to_account where from_account = $n_i$
Set j = 1
Repeat
    Set fromNode = $n_i$
    Set toNode = $m_j$
    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(*)count from matrix_table where from_account = $m_j$
    If count > 0 then
        If $m_j$ ¢ $P_1$ and $m_j$ ¢ $P_2$ then
            $P_1$.add ($m_j$).
            Call reachability ($m_j$) (recursive function)
        Else
            Insert into vector K Select to_account where to_account = $m_j$
            Set l = 1
            Repeat
                If $k_l$ = $n_i$then
```

```
                        Call reachability($k_l$)
                        (recursive)
                    End If
                    Set l = l + 1
                until l = K.length
                Set path = newPath and
                $P_2$.addNode($m_i$).
            End if
        Else
            path  = newPath
            $P_2$.addNode ($m_i$).
        End If
    Until j = M.length
    $P_1$.removeNode ($m_i$).
    $P_2$.addNode ($m_i$).
    remove last node from the path
```

End



Money Laundry - Eng. Tamer Hossam ElDin Helmy - PHD - MTC

Link Analysis

search Cretiria

Account No: 8

Identification No:

Graph Cycles

From Account	To Account	Cycle Path
6	6	6-7-8-6
6	6	6-7-8-9-6
7	7	7-8-6-7
7	7	7-8-9-6-7

Mouse Mode: TRANSFORMING

Transaction Type	Transaction Date	From Account	From Name	From Country	to Account	to Country	Amount	Currency
3	2012-05-01	8	Ayman Abd Elfatah	USA	9	Egypt	9000.0	L.E
3	2012-06-10	8	Ahmed Hessen N...	USA	6	Egypt	9000.0	L.E

Figure 4 Cycle Detection Case

3.2.5. 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.

Input	T				T				Ʈ			
Present State	T_1	T_2	….	T_m	t_1	t_2	….	t_k	$Ʈ_1$	$Ʈ_2$	….	$Ʈ_l$
$q_s\ =\ q_o$	$^{q_o}/_{nil}$	$^{q_1}/_{nil}$		$^{q_1}/_{P_1}$	$^{q_o}/_{nil}$	$^{q_2}/_{nil}$		$^{q_2}/_{P_2}$	$^{q_o}/_{nil}$	$^{q_3}/_{nil}$		$^{q_3}/_{P_3}$
q_1				$^{q_1}/_{P_1}$								
q_2								$^{q_2}/_{P_2}$				
q_3												$^{q_3}/_{P_3}$

Table 4 Mealy Representation of a 4-State Example

4. FINITE STATE AUTOMATA FOR MONEY LAUNDERING SCENARIOS

The proposed monitor works by making use of an automata approach [21]. Such automata are represented using Mealy notation [22]. Thus the monitor in its general form is represented by

S = <$\boldsymbol{Q, T, \delta, \Omega, P}$> in which Q is a set of states, where the start state q_s ϵ Q, T is a finite set of input transactions, $\boldsymbol{\delta}$ and $\boldsymbol{\Omega}$ are the next state and output mappings, respectively. The finite set P is the set of monitor outputs, {P_k ; k = 1,2,…K or nil}. For that automata a scenario can be represented by an inputT_i; i = 1,2,…,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 ∩ Q = Ø. 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.

q_o = q_s : Start state.

q_1: Monitor state after 1st monitored transaction.

q_2: Monitor state after 2nd monitored transaction.

q_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, Ʈ 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 = $T^{+}\ N^{*}\, T^{+}$

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 = $t^{+}\ N^{*}\ t^{+}$

Cycling = Ʈ$^{+}$ N^{*} Ʈ$^{+}$

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

Input Present State	T				T				Ʈ				Crime found
	$\boldsymbol{T_1}$	$\boldsymbol{T_2}$	….	$\boldsymbol{T_m}$	$\boldsymbol{t_1}$	$\boldsymbol{t_2}$	….	$\boldsymbol{t_k}$	Ʈ$_1$	Ʈ$_2$	….	Ʈ$_l$	
$0q_{11}q_{21}\dots q_{j1}$	q_{12}				q_{22}				q_{j2}				
q_{12}		$q_{1(n-1)}$											
$q_{1(n-1)}$				q_{1n}									
q_{1n}				q_{1n}									Smurfing
q_{22}						$q_{2(r-1)}$							
$q_{2(r-1)}$								q_{2r}					
q_{2r}								q_{2r}					Immediate withdrawal
q_{j2}										$q_{j(s-1)}$			
$q_{j(s-1)}$												q_{js}	
q_{js}												q_{js}	Cycling

Table 5 The Transition Table of the DFA

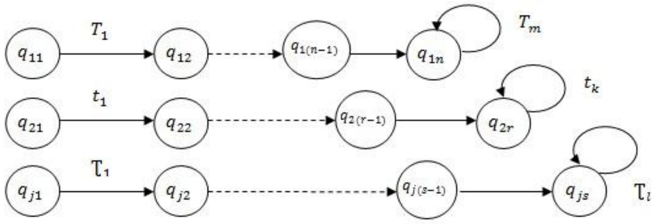


Figure 5 Money Laundering Cases Representations

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 [23]. 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.

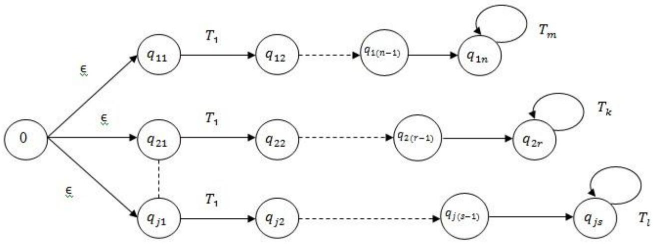


Figure 6 Non Deterministic Finite Automata for Money Laundering Cases

5. CASE STUDY: UNTRACEABLE MONEY SOURCE

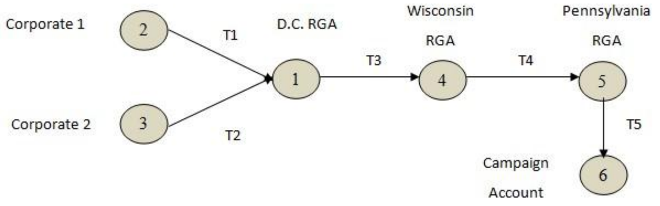


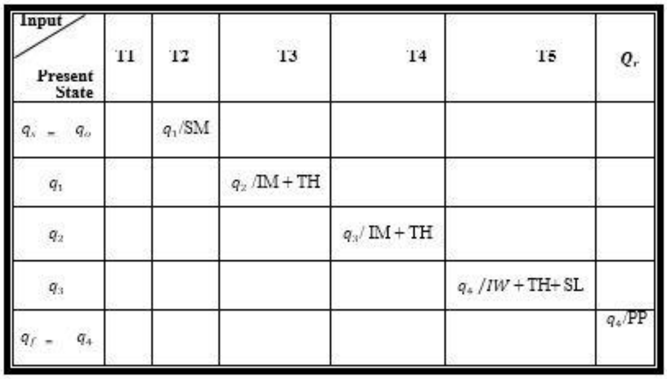
Figure 7 Case's Money Flow Diagram

This case describes untraceable money donation to the Pennsylvania governor election campaign [24]. 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.

5.1. 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.



Input / Present State	T1	T2	T3	T4	T5	Q_r
q_s = q_o		q_1/SM				
q_1			q_2 /IM + TH			
q_2				q_3/ IM + TH		
q_3					q_4 /IW + TH+ SL	
q_f = q_4						q_4/PP

Table 6 The Monitor Finite State Machine

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

[1] P. Lilley, "Dirty dealing, the untold truth about global money laundering, international crime and terrorism", Kogan page limited, 2000.

[2] P. Reuter, and E. Truman, "Chasing dirty money", Institute for international economics, Washington, USA, 2005.

[3] G. SHIJIA, X. Dongming, W. Huaiqing, W. Yingfeng, "Intelligent Anti Money Laundering System", Service Operations And Logistics, And Informatics, Ieee International Conference, Pp. 851-856, 2006.

[4] R. Molander, D. Mussington, and P. Wilson, "Cyberpayments and money laundering problems and promise", Financial crime enforcement network, RAND Critical technology institute, Washington, USA, 1998.

[5] F. Kwong, "Anti Money Laundering", Computer Assisted Auditing Techniques, Retrieved 1 July 2011 available at

http://uwcisa.uwaterloo.ca/Biblio2/Topic/ACC626%20Anti%20Money%20Laundering%20F%20Kwong.pdf

[6] Pellegrina, L. D, and Masciandaro, "The Risk Based Approach in the New European Anti-Money Laundering Legislatin",Retrieved May 29, 2011, available at http://www.bepress.com/cgi/viewcontent.cgi?article=1422&context=rle

[7] J. Wit, "A Risk-Based Approach to AML: A Controversy between Financial Institutions and Regulators", Journal of Financial Regulation and Compliance, Vol. 15, PP. 156-165, 2007.

[8] T. Senator, H. Goldberg, J. Wooton, M. Cottini, U. Khan, C. Klinger, W. Liamas, M. Marrone, R. Wong, "The FinCEN Artificial Intelligence System: Identifying Potential Money Laundering from Reports of large cash transactions", U.S. Department of the Treasury - Financial Crimes Enforcement Network (FinCEN), Association for the Advancement of Artificial Intelligence, USA, Vol. 16, No. 4, PP. 156-170, 1995.

[9] T. Jun. "A Peer Dataset Comparison Outlier Detection Model Applied to Financial Surveillance". Proceedings of the 18th International Conference on Pattern Recognition, IEEE Computer Society, Washington, USA, Vol. 4, PP. 900-903, 2006.

[10] S. Wang, J. Yang, "A money laundering risk evaluation method based on decision tree", Proceedings of the International Conference on Machine Learning and Cybernetics, PP. 283-286, Hong Kong, 2007.

[11] Y.Wang, H. Wang, S. Gao, D. Xu, "Intelligent Money Laundering Monitoring and Detecting System", European and Mediterranean Conference on Information Systems, Dubai, available at "http://www.iseing.org/emcis/emcis2008/Proceedings/Refereed%20Papers/Contributions/C%2045/Intelligent_Money_Laundering_Monitoring_and_Detecting_System.pdf ", 2008.

[12] N. Le-Khac, S. Markos, M. Kechadi, "A Heuristics Approach for Fast Detecting Suspicious Money Laundering Cases in an Investment Bank", International Scholarly and Scientific Research AND Innovation, Engineering and Technology, Vol. 3, No. 12, PP. 70-74, 2009.

[13] J. Luell, "Employee Fraud Detection under Real World Conditions", Ph.D. thesis, University of Zurich, 2010.

[14] Anti-Money Laundering Countering Financing of Terrorism, "National Risk Assessment 2010", Financial Intelligence Unit, New Zealand Police, New Zealand, available at http://www.justice.govt.NZ/policy/criminal-justice/AML-CFT/publications-AND-consultation/national-risk-assessment-2010, 2010.

[15] Financial Action Task Force, "International Standards on Combating Money Laundering and the Financing of Terrorism and Proliferation – The FATF Recommendations", available at "www.fatf-gafi.org/recommendations ", 2012.

[16] Financial Transactions and Reports Analysis Center, Typologies and Trends, "Money Laundering and Terrorist Financing (ML/TF) Typologies and Trends for Canadian Money Services Businesses (MSBs)", FINTRAC Reports of Canada, Canada, 2010.

[17] The Canadian Institute of Chartered Accountants, "Canada's Anti-Money Laundering and Anti-Terrorist Financing Requirements", available at www.cica.ca, 2008.

[18] Belgian Financial Intelligence Processing Unit, "Money Laundering Indicators", available at http://www.ctif-cfi.be/website/images/EN/typo_ctifcfi/NL1175eENG.pdf, 2007.

[19] E. Garcia, P. Regan, J. Stern, W. Johnson, R. Macallister, J. Reidenberg, D. vogt, R. Serino, B. Proter, S. Welling, "Information Technologies for the Control of Money Laundering", OTA-ITC-630, Washington, USA, 1995.

[20] Australian Transaction Reports and Analysis Center, "Money Laundering in Australia, Australia Government, available at http://www.austrac.gov.au/files/money_laundering_in_australia_2011.pdf, 2011.

[21] R. Fuhrer, "Sequential Optimization of Asynchronous and Synchronous Finite-State Machine: Algorithms and Tools", Springer, 2001.

[22] D. Lee, "Principles and Methods of Testing Finite State Machine survey", Proceedings of the IEEE, Vol. 84, No. 8, PP. 1089-1123, 1996.

[23] A. Aho, R. Sethi, J. Ullman, "Compilers: Principles, Techniques, and Tools Second Edition", Addison Wesley, 2006.

[24] The Center for Public Integrity, available at http://www.publicintegrity.org/2012/10/18/11498/pennsylvania-governor-benefited-untraceable-15-million-donation, 2012.

Authors

Tamer Hossam Eldin Helmy received his BSc and MSc in Computer Engineering from Military Technical Collage and Al-Azhar University, Egypt in 1996 and 2010 respectively. He received his PhD in Computer Engineering from Military Technical Collage, Egypt on 2014. He is interesting in artificial intelligence, Knowledge discovery, ontology and Web technology.

Mohamed Zaki is a Professor of Software Engineering in the Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University at Cairo. He received his BSc and MSc in Electrical Engineering from Cairo University in 1968 and 1973 respectively. He received his PhD in Computer Engineering from Warsaw Technical University, Poland in 1977. His fields of interest include artificial intelligence, soft computing, and distributed systems.

Tarek Salah Sobh received his B.Sc. degree in computer engineering from Military Technical College, Cairo, Egypt in 1987. Both M.Sc. and Ph.D. degrees from Computer and System Engineering Department, Faculty of Engineering, Al-Azhar University, Cairo, Egypt. He has managed, designed and developed several package for business applications and security systems. He has authored/co-authored of many refereed journal/conference papers and booklet. Some of the articles are available in the ScienceDirect Top 25 hottest articles. His research of interest includes computer networks, security systems, distributed systems, knowledge discovery, data mining, and software engineering.

Khaled Mahmoud Shafea received his BSc and MSc in Computer Engineering from Military Technical Collage, Egypt on 1995 and 2001 respectively. He received his PhD in Computer Engineering from Electrical and Electronic Engineering Department, University of Sheffield, UK in 2009.