A Secure Framework for Anti-Money-Laundering using Machine Learning and Secret Sharing

Arman Zand*, James Orwell†, Eckhard Pfluegel*
*Faculty of Science, Engineering & Computing, Kingston University, London, UK, {a.zand, e.pfluegel}@kingston.ac.uk
†Kubrick Group, United Kingdom, jamesorwell@kubrickgroup.com

***Abstract*—Nowadays, the scale of Money Laundering is difficult to estimate in the UK and elsewhere. Proceeds of crimes might be transferred using the available business infrastructure offered by banks, and this is a considerable problem. This paper outlines a novel scheme that allows banks to share information leading to Money Laundering (ML) detection all the while preserving confidentiality and integrity. The main contribution is the overall architecture that aims to improve ML detection by getting other banks to collaborate. In order to get other banks to co-operate, a primary directive of preserving privacy is enforced throughout the framework. The proposed scheme has two particular aspects, one of which is the application of encrypted data used in machine learning for ML detection. Another feature is using secret sharing as a collaborative element in this context. These aspects are found in the three phases of the framework: Signalling to the Auditor, ML Detection and finally Suspicious Activity Report (SAR) Feedback.**

***Index Terms*—anti-money-laundering, machine learning for security, secret sharing.**

I. Introduction

Money laundering activity is associated with various types of crime, and efficient detection of this activity strongly contributes to the prevention and prosecution of those crimes. The true scale of this activity is difficult to estimate accurately: one estimate [1] puts the proceeds at *£4.5B* annually for drug supply and *£5.9B* for fraud. There are several categories of money laundering (ML), including the use of property, gambling, and businesses to obfuscate the true source of funds. This paper concentrates on the use of proxy or 'mule' accounts for the swift transfer and 'cash out' of illegally obtained funds. The precise manner in which these funds are obtained is not the subject of this paper, but typical examples include deceiving the bank customer into transferring funds, theft by re-ordering and then intercepting bank authentication devices, or re-registering the phone number or postal address associated with the bank account. These funds are then typically transferred again, possibly multiple times and into smaller fragments, so that the parties involved in this organised crime can securely access them.

This paper concerns the design of Anti-Money Laundering (AML) systems to detect of that pattern of transactions and addresses mainly two aspects. The first aspect is the requirement for an AML system to respect certain confidentiality constraints, given that both banks and their clients have commercial and legal motivations for maintaining the privacy of transaction information. An innovative system is proposed that satisfies these constraints, and thereby provides the opportunity for banks to collectively use the system, by generating a cryptographic code for each transaction, that restricts certain transaction details to the authorised party (i.e. the originating bank). These codes are shared on a ledger using a suitably discrete protocol, for input into a detection process. Likewise, a secret sharing approach is proposed for the protocol to share the results of this detection process, which in turn enable the production of Suspicious Activity Reports (SARs). For any transaction included in these results or those reports, the transaction details are disclosed only to the authorised party, via the required cryptographic process, thus maintaining confidentiality.

The second aspect we consider is the context for the detection process that can operate on the set of contributed cryptographic codes. A suitable real-time detection architecture is proposed, with detailed consideration of data-set characteristics, engineering of input features and classification approaches for minimising the log entropy of the output probabilistic estimates. Hence, the proposed system is capable of producing, for each transaction, an estimate of the probability that it is associated with money laundering activity, while restricting the visibility of that estimate only to the related banks, and the associated transaction details, as determined by the parameters of the secret sharing protocol. More generally, the estimate for each transaction can be included alongside estimates derived from other approaches, outside the domain of transaction analysis, as part of an overall process for accurate and timely detection of money laundering activity.

The paper outlines a scheme that allows banks to share information leading to ML detection all the while preserving confidentiality and integrity as the overview of the framework (Figure 1) shows. Firstly, relevant literature review of money laundering detection and existing advancements highlight problem areas and research questions to gain a better stance in designing the scheme. More robust approaches in analysing data is found in machine learning techniques where requirements and related work is explored. Other state-of-the-art cryptographic technologies provide solutions for establishing authorisation to access information. These various aspects are put together into different phases that help banks collaborate in a naturally forming initiative of ML detection.

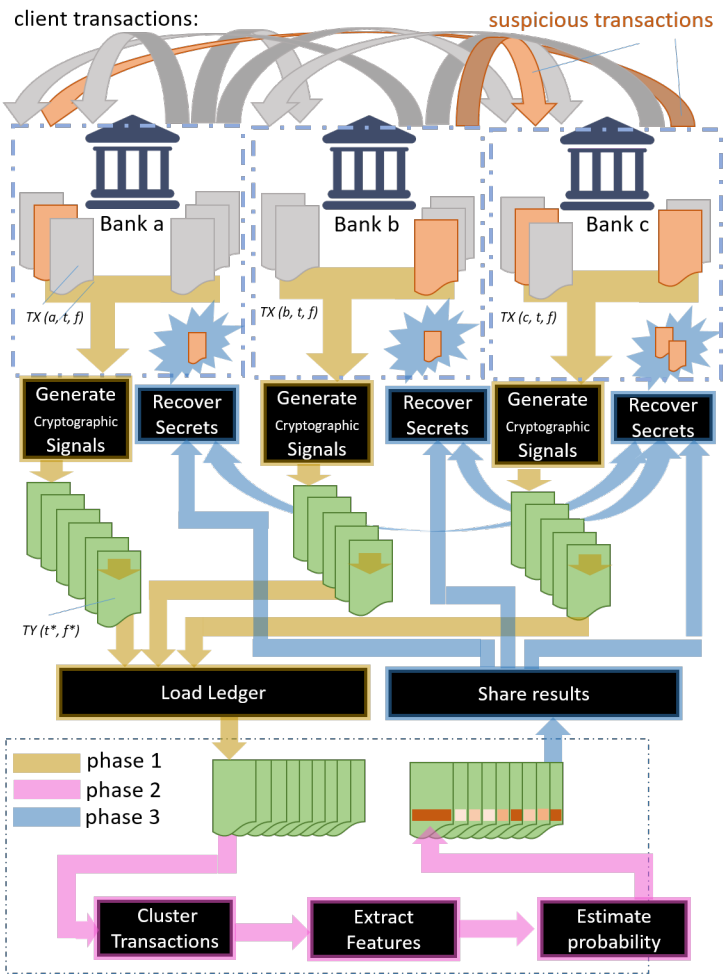

Fig. 1. The general framework of the scheme covered in this paper which highlights the three phases. Firstly banks convert their transactions into encrypted signals i.e. the green pages and subsequently processed by the Auditor. Secondly, the Auditor processes this input of data with a machine learning classification algorithm. The Auditor determines which transactions are suspicious despite not being able to infer sensitive information. Finally the Auditor splits the results into shares where the banks will have to communicate with each other in order to recover their results.

II. Related Work

A. Categories of Detection Approach

A review paper on AML [2] describes five categories of advances that have been made in this domain. The first category is the detection of suspicious *transactions*. The second category is the detection of *entities* that may be associated with money laundering, through analysis of patterns, groups and anomalies. Thirdly, there have been advances in the analysis and assessment of *risk* factors. Fourth, there are increasingly sophisticated tools for controlling the structure and governance of *assets* and *processes* that might be used in ML activities. Finally, there have advances in the *visualisation* of threats and activities to facilitate further analysis. Investigation into the first and second category are relevant for the framework in this paper.

B. Machine Learning Technologies

There are several issues associated with the deployment of machine learning techniques in this application domain. The first issue is data quality: there may be missing values in the data input to the analysis. The causes of such missing data include variations in the technical specification of the systems used by the various institutions, and non-participation of some institutions for reasons of privacy or commercial confidentiality. Moreover, some input data values may be corrupted or subjected to alternative definitions (e.g. currency and time-date formats) that add noise. Various strategies are proposed to identity and mitigate the effect of missing or corrupted data, including filtering algorithms such as mean substitution, cold deck imputation and hot deck imputation methods [3].

Secondly, a criticism of solutions that include machine-learned components is that there may be a risk that laws or ethical codes are breached. One such set of laws concerns privacy, the disclosure of data to third parties, the active consent of the data subjects, and the principle that data is only used for the stated purpose. Another set of laws relates to fair treatment, and the risk that machine-learned algorithms are effectively discriminating against certain population groups, via the statistical learning method used in their creation. researchers are not entirely mindful on conflicting laws, particularly in privacy and discrimination.

A third issue is the performance of ML detection systems; there are several aspects to this performance. Firstly is the system *accuracy*, typically measured with reference to the precision and recall of the detection results, *i.e.* what proportion of the transactions labelled as 'high probability' are indeed ML transactions, and what proportion of ML transactions are labelled as high probability, respectively. The decision to act on an alarm is not to be taken lightly, since it incurs operational cost, and also a risk of reputational damage, if incorrect. A further factor for system accuracy is the dynamic nature of the signals to be detected: ML actors will change tactics to avoid detection, and exploit new systems and opportunities as they become available. The challenge for any AML system is to react to this changing environment: since machine learning algorithms rely on training datasets, this challenge is particularly acute.

The *computational* performance is an important consideration and has several factors: whether the proposed system is a real-time system (or an off-line batch processing design); how the proposed system will scale with increasing rate of transactions and also with respect to the increasing size of historical data.

A real time capability has the significant benefit of enabling intervention in the ML activity, whereas batch processing can only provide evidence for forensic investigations or research. Two machine-learning approaches have been proposed in this domain: clustering algorithms [4], to identify groups of related transactions, and then classification algorithms [5], to estimate the 'degree of suspicion' associated with any given group.

Some advantages to aim for include scalability for any amount of data, rate of precision, speed of training and automatic parameter tuning.

C. Confidentiality Preserving Systems

The detection of money-laundering activity through the analysis of multiple transactions requires some domain in which these multiple transaction can be subjected to this analysis. If the transactions are all under a single authority, or if there is agreement between multiple authorities to share this information in plain-text form, then this requirement can be satisfied without subsequent cryptographic processing.

However, if neither of these conditions are satisfied, then there exists a confidentiality constraint. One solution [6] has been proposed in this domain, using decision trees to classify transactions in a two party problem. In addition to working within this confidentiality constraint, it also addressed several topics noted above: non-stationary patterns, scalability and redundancy.

Shamir secret sharing (k, n) [7] is a privacy preserving but collaborative scheme that splits up a secret into n shares where k of them are enough to recover it. Having less than this threshold value will not yield any information about the secret. This scheme is amalgamated in the AML framework which is explained further in Section VI.

From the five categories listed in section II-A, this paper will focus on systems for the detection of suspicious transactions. This has been previously investigated [8], through use of specific software tools such as FTK (Forensic Toolkit). The confidentiality constraints are also discussed. There are some recurring terminologies in the literature such as KYC (Know Your Customer) which employs policies or procedures that assess and verify the identity of customers. In addition to SARs (Suspicious Activity Reports) [8], [9].

III. System Design

In this section, the system requirements are stated for a AML framework, and then an architecture is proposed that can deliver a system that satisfies these requirements.

A. Requirements

We consider an MLDS (money laundering detection system), used by two or more *banks*, which have records of transactions that reference 'to' and 'from' client account numbers, each held in a given bank. Hence, a given transaction will generate two records, from both 'from' and 'to' banks (assuming they are both participating in the MLDS). The transaction will also include timestamp and quantity attributes.

The MLDS also includes one or more *auditors*, whose role is to analyse these transactions, albeit via some confidentiality-preserving cryptographic protocol. This analysis results in each record being assigned a 'degree of suspicion'. It is supposed that the banks trust to the auditors to some limited extent, i.e. the auditor is contractually bound to confidentiality, but for commercial and regulatory reasons, banks wish to limit the information shared with the auditors. The framework consists of some important non-functional requirements:

- Banks that share a transaction (e.g. Money sent from 'Bank A' to 'Bank B') will mean both banks will have to send their encrypted format at the same time to the Auditor.
- The Auditor processes pairs of transactions in order to create a directed acyclic graph and be able to calculate certain properties about it in order to feed the Machine Learning algorithm.
- The client account details in each record shall remain confidential to the banks throughout the MLDS process. This includes 'to' and 'from' bank identifiers. (This information may be subsequently revealed in some limited capacity, at the discretion of the bank authorities, in compliance with the relevant regulations.)
- The timestamp and quantity details in each record shall be disclosed to auditors, but without disclosing 'from' and 'to' bank identifiers.
- The additional 'probability of suspicion' attribute (which may be assigned to records by auditors) shall remain confidential to the originator of the record, i.e. the 'to' or 'from' bank. (One more, this information may be revealed by the originator of the record, at their discretion.)

B. Architecture

Comprehension of the scheme revolves around a number of objectives. The redacted/encrypted transactions produced by banks are sent to the auditor. The Auditor is able to categorise these transactions by their relationships with each other. This is proceeded with detection of money laundering, a group of transactions are converted into a SAR. This SAR is split up by the Auditor and communicated to the banks where in order to recover the SAR, co-operation and collective permission is required. Finally, throughout the scheme, security objectives have to be met i.e. prevent information from being inferred by unauthorised entities and maintaining integrity. The scheme involves two types of entities: Banks and an Auditor. Banks can arrange themselves in a peer-to-peer network, satisfying certain conditions which shall be detailed in the next section. The Auditor receives encrypted/partially redacted transactions from the banks and determines their "suspiciousness". The Auditor will then also allow the banks to co-operate in recovering SARs with a specific cryptographic techniques, secret sharing.

C. AML Phases

In Figure 1, the general framework depicts the three phases (colour coded) and the flow of information. In order to understand the diagram better, it may be necessary to see Table I that describes the notation and Figure 2 which shows how the Auditor receives transactions. These phases have their own requirement that are elicited:

Phase 1 - **SIGNALLING** to the Auditor

1) Each bank converts their respective transactions into a redacted/encrypted format.
2) The Auditor and any other party is not authorised to read plaintext transaction data. Amount and timestamp is not redacted.
3) The banks involved in the same transaction send a "pairwise" signal to the auditor.

Phase 2 - ML **DETECTION**

1) The Auditor organises relationship with received information as trees of transactions.
2) The Auditor processes each transaction into the neural network as it accesses them.
3) Once the neural network determines money laundering activity the relevant transactions are put into an SAR.
4) Auditor bundles together the transactions and respective probability of ML into the SAR. This information may only be authorised to the source banks.

Phase 3 - SAR **FEEDBACK**

1) The Auditor splits up the SAR into shares and sends them to the banks.
2) The banks co-operate to recover the SAR by putting together their shares.
3) Only source banks will recognise their transactions which will be able to attribute the probability attached.

TABLE I
TABLE OF MISCELLANEOUS NOTATIONS.

Notation	Description
$TX_Z{}_T^F$	Plaintext transaction TX produced by bank Z where T is the 'To' bank account and F is the 'From' bank account.
$TY_Z{}_T^F$	Ciphertext transaction TY produced by bank Z where T is the "To" bank account and F is the 'From' bank account.

D. Network Security Setup

Banks and auditor setup a Public Key Infrastructure (PKI) using certificates [10] which prevents Man-In-The-Middle attacks where trust and public keys are established. This allows entities to perform a key exchange and key agreement as with the TLS protocol [11]. Commonly used public key encryptions are RSA [12] and Elliptic Curves [13] which is negotiated in TLS as well as the symmetric encryption algorithm such as AES [14]. In combination, there is a strong confidence of confidentiality and integrity of data when two entities communicate. The banks have the freedom to arrange themselves in a peer-to-peer network because of the pairwise signals that are sent which therefore means the Auditor must be a central authority for all the banks.

IV. PHASE 1: SIGNALLING

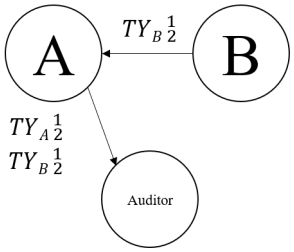

Fig. 2. Communication example of the first phase. Bank A and B have a transaction object $TX_A{}_2^1$ and $TX_B{}_2^1$ which are encrypted with their own secret key to produce $TY_A{}_2^1$ and $TY_B{}_2^1$. Bank B sends this encrypted transaction to Bank A so that Bank A may attach its encrypted version together and send both of them together to the Auditor.

For this section, the objective is for the Auditor to be unable to discern which transactions belong to which bank. Transactions between two bank accounts (say bank account 1 sent money to bank account 2) coming from two different banks: A and B respectively would both produce a transaction signal for the auditor represented as $TX_A{}_2^1$ and $TX_B{}_2^1$ which would be communicated as such in Figure 2 where the encrypted versions are denoted by TY. With the use of cryptographically secure pseudo-random number generators to fill out the parameters of AES [14], the banks encrypt the properties of the transaction such as the 'To' and 'From'. One important property is a unique code to represent this transaction for later recognition for which a secure hash function: SHA512 on large pseudo-randomly generated number would be appropriate. Both related banks produce this redacted form of the transaction they share. The recipient of the transaction as with the example in Figure 2 sends this to the counterpart. Then, once this bank has both transactions (that is, Bank A receives Bank B's version of this transaction encrypted with Bank B's private key), it will forward it off to the auditor together which is the pairwise signal. When forwarding this transaction, the network messages are encrypted with the individual bank-to-auditor key established while setting up the network.

Fundamentally, the auditor would receive the same transaction but produced and encrypted by two different entities. The information between them, although is the same in plaintext, may not infer or reveal to unauthorised entities i.e. entities that do not have the relevant secret key. To specifically perform this in a cryptographic manner, AES in mode "Synthetic Initialisation Vector" [15] allows reusing nonces in a way that doesn't risk the security of the algorithm. This means the encryption of data is deterministic in a sense that a certain plaintext always produces the same ciphertext. Due to this encryption method, the Auditor is able to match together different transactions without knowing which accounts they

actually are albeit that banks still control the confidentiality of this information which they can later decrypt.

V. Phase 2: Detection

In the preceding section, it was shown how the sensitive information from each transaction can be non-reversibly substituted, so that the set of transactions can be subjected to analysis, without disclosure of that information. In this section, the main contribution is adding privacy to the machine learning processing by using cryptographic algorithms in a money laundering context.

A. Output Requirement and Cost Function

We argue that the appropriate goal for this analysis is to assign to each transaction, an estimate of the probability that it is associated with Money-Laundering Activity, in other words the probability that it is 'suspicious'. This can be written $p_s(TY)$. There is some ambiguity about the definition of this term, for suspicion may rightly fall on transactions that transpire to be innocent. However, the motivation is to obtain records for inclusion in the 'Suspicious Activity Report', hence the use of this term. Thus, in this paper, the label 'suspicious' refers only to those transactions that are actually connected with some money-laundering activity, rather than the broader sense of those that have a raised likelihood of such a connection.

The labels $l_s(TY)$ that are used for training are drawn from a discrete binary support: either $l_s(TY) = 1$ (suspicious) or $l_s(TY) = 0$ (not suspicious). The aim is to predict these values as accurately as possible, using the output from the classifier $p_s(TY)$, for which the action space is in the range between 0 and 1.

The cost (or loss) function C() that is used to evaluate the accuracy of the classifier is referred to as the log-loss or cross-entropy loss function.

$$C_s = \sum_i l(TY_i) \log p(TY_i) + \sum_i (1 - l(TY_i)) \log(1 - p(TY_i)).$$

B. Selection of Machine Learning Architecture

While it may be possible to devise a rule-based system for the detection of anomalies, the standard approach is to cast this task as a machine-learning problem, and devise a framework in which a detector can be configured to 'learn' from a set of training data.

The main categories of machine learning framework are: supervised, unsupervised, reinforced and recurrent (or LSTM) learning. Of these, we argue that the supervised paradigm is the most suitable for this case. This approach uses labelled training data to learn a mapping between input and output, which can then be used when the output labels are withheld or unavailable (i.e. for testing and deployment, respectively). The input transactions can either be considered as a stream (for real-time analysis) or as a set (for offline analysis). In either case, they can be arranged into a directed graph network, in which the bank accounts are the nodes and the transactions are the edges, the direction of the transaction determining the direction of the edge.

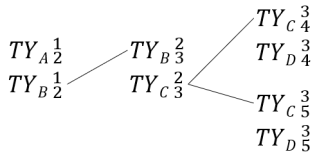

Fig. 3. How the Auditor may link different pairs of transactions together by finding mutual values in the 'To' or 'From' property.

The Auditor may be able to produce these graphs with pairs of transactions that can be directly linked with these mutual values and can form a tree like Figure 3. Features of this arrangement of transactions are extracted such as the delay between different children, the number of grandchild transactions, etc. are inserted into a vector. The certain features extracted may be explored further to increase the number of factors to input into the machine learning algorithm and potentially improving the estimated probability. Once, the neural network has determined the probability of ML for the transactions and has detected ML, the auditor creates a SAR from the collection of suspicious transactions. To reiterate, the auditor still does not have access to the encrypted information originally sent by the banks in the first phase.

VI. PHASE 3: SAR Feedback

This section describes the final phase of the framework. In the previous phase, the Auditor has detected suspicious ML and has to proceed to give feedback. The contribution here is the application of secret sharing to combine collaborative and privacy preserving elements in a money laundering detection paradigm. The literature review has not found similar solutions. [2]. This phase involves the co-operating element where the Auditor splits up a SAR and gives the shares out among the banks. In terms of security it requires an eavesdropper for every channel a share is sent across. In addition, even after recovering the results, only the source bank may be able to decrypt the original transaction sent to the auditor. Therefore, the secret sharing adds a layer of security and consensus for the banks to see the results.

A preview of the results is given by taking the unique codes attached to the transactions. This preview is given to the banks along with a share of the SAR. When a bank receives this preview and are able to recognise a code, it means that it is involved in the SAR and may decide to abdicate. None-the-less, the banks put together their shares to recover the SAR containing the probability of ML. Subsequently, the banks will be able to decrypt their own transactions originally sent in the first phase and therefore are able to attribute the probability value attached. Finally, the banks may then be able to share the results. With Shamir secret sharing [7], no knowledge may be gained when obtaining any less shares than the threshold amount k.

Referring this section so far to Table II, recalling basic Secret Sharing notations, the SAR would be substituted into

TABLE II
TABLE OF SECRET SHARING NOTATIONS.

Notation	Description
Input Parameters	
k	Threshold number of shares to recover s.
n	Overall number of nodes/shares to produce.
Classic Shamir 'Dealing'	
i	Unique node index $(1 \leq i \leq n)$.
$\mathbb{F}_q$	Finite Field of characteristic q where $q > n$.
s	Secret to distribute.
r_a	Vector of size $((k-1) \times 1)$ of random values.
Ψ	$\begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ i^0 & \dots & i^k \end{bmatrix}$ Encoding Vandermonde matrix which are the selected n shares to produce.
α	$\begin{bmatrix} s \\ r_a \end{bmatrix}$ Matrix containing secret to split.
t_i	Element of vector from the result of $\alpha \cdot \Psi$.
Classic Shamir 'Recovering'	
Ψ^{-1}	Inverse of Vandermonde matrix with respect to selected share indexes in ω.
ω	Vector containing k shares $\begin{bmatrix} t_i \\ \vdots \\ t_k \end{bmatrix}$
α	Recovered information from the result of $\Psi^{-1} \cdot \omega$

s and depending on the number of banks participating, the (k, n) parameters are adjusted. A share t_i is produced when the Dealer, which in this case, the Auditor computes $\alpha \cdot \Psi$ which is a vector containing n shares and taking the ith share in this vector, hence t_i would be given to the respective bank. The second part involving the recovering of the contents requires k banks to co-operate and send each other their share. Once k shares are obtained for example t_1, t_2, t_4 then the bank would then produce the inverse of Vandermonde matrix arranged as so:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 4 & 16 \end{bmatrix}^{-1} \cdot \begin{bmatrix} t_1 \\ t_2 \\ t_4 \end{bmatrix} = \begin{bmatrix} s \\ r_{a_1} \\ r_{a_2} \end{bmatrix}$$

One of the challenges in this scenario is the fact that the banks may not necessarily trust each other, as a number of potential attacks could arise. In a peer-to-peer network, it is required to address the Byzantine Agreement Problem which would also add constraints to the arrangement of the network and protocol design. Colluding banks may decide to use the results to gain some competitive advantage such as damaging the reputation of other banks. A wider attack surface is given in this way giving more opportunities for performing Man-in-the-Middle attacks especially that of bank-to-bank communication rather than bank-to-auditor. A more centralised approach means the banks will send encrypted transactions directly to the Auditor using the appropriate keys agreed in the network setup and also directly receive the output.

VII. IMPLEMENTATION

A prototype implementation of the AML framework has been made, comprising of the different phases: signalling, ML detection and SAR feedback.

The popular programming language in the cyber security community, Python, has been used for the implementation. It is fast, lightweight, works cross-platform and thus simplifying development and deployment. The implementation covers the beginnings of the communication network as a bottom-up approach where the subsequent stage of implementation focuses on the ML detection engine. A preliminary evaluation, based on an implementation of the three framework phases in a simulated network has been carried out. No significant performance issues have been detected for networks comprising of a small number of banks. A systematic study for larger network of banks will be subject of further work. The cryptographic algorithms described in the multiple phases have been applied in the implementation, that is for example the AES SIV method that banks use to encrypt bank details. Synthetic data was produced in the simulation with labelled transactions which has a positive detection rate of 98% using a Fast Tree binary classification algorithm. Dataset was produced with various derivatives of transaction properties such as time, relationships (e.g. transaction children) and amount.

VIII. CONCLUSION

In this paper, a proposed ML detection scheme establishes requirements and actions take to have multiple banks collaborate in an AML initiative. Cryptographic autonomy is given to the discretion of the banks, performing real-time analysis processed by a machine learning algorithm to finally have the results fed back to the banks in a co-operative manner. Substituted and encrypted information simulated in the preliminary implementation demonstrates the existence of promising AML framework with a reasonable detection rate. However, it is necessary to highlight that transactions that depart to non-participating banks prevents the scheme from drawing the full picture of a plausible money laundering string. Further work involves exploring sharing of the work-load in detecting money laundering where banks may interact in a peer-to-peer network without a central authority. This type of network could be investigated further with distributed ledgers where each node performs some neural network processing as a contribution. It could also be mentioned that changing what actions that are performed in one phase may have ripple effects such as doing a pairwise signal to the auditor, if replaced it would mean to change how the auditor finds relationships and what information banks can decrypt. Therefore, it is detrimental to be mindful of changes and carry out improvements in such fashion that the framework would not require a great amount of consolidation.

REFERENCES

[1] E. Fell, O. James, H. Dienes, N. Shah, and J. Grimshaw, "Understanding organised crime 2015/16," Feb. 2019, pp. 11,18,31, 38.

[2] G. Leite, A. Albuquerque, and P. Pinheiro, "Application of Technological Solutions in the Fight Against Money Laundering — A Systematic

Literature Review." Multidisciplinary Digital Publishing Institute, Nov. 2019.

[3] M. Brown and J. Kros, "Data mining and the impact of missing data," *Industrial Management and Data Systems*, vol. 103, pp. 611–621, 11 2003.

[4] S. Haider, "Clustering based anomalous transaction reporting," *Procedia CS*, vol. 3, pp. 606–610, 12 2011.

[5] L.-T. Lv, N. Ji, and J.-L. Zhang, "A rbf neural network model for anti-money laundering," in *2008 International Conference on Wavelet Analysis and Pattern Recognition*, vol. 1. IEEE, 2008, pp. 209–215.

[6] C. Ju and L. Zheng, "Research on suspicious financial transactions recognition based on privacy-preserving of classification algorithm," in *2009 First International Workshop on Education Technology and Computer Science*, vol. 2. IEEE, 2009, pp. 525–528.

[7] A. Shamir, "How to share a secret." New York, NY, USA: Association for Computing Machinery, Nov. 1979, vol. 22, no. 11, p. 612–613. [Online]. Available: https://doi.org/10.1145/359168.359176

[8] D. Flores, O. Angelopoulou, and R. Self, "Combining digital forensic practices and database analysis as an anti-money laundering strategy for financial institutions," 09 2012, pp. 218–224.

[9] Z. Chen, L. Khoa, E. Teoh, A. Nazir, E. Karuppiah, and K. Lam, "Machine learning techniques for anti-money laundering (aml) solutions in suspicious transaction detection: a review," Feb. 2018.

[10] M. Cooper, Y. Dzambasow, P. Hesse, S. Joseph, and R. Nicholas, "Internet X.509 Public Key Infrastructure: Certification Path Building." Internet Engineering Task Force, 2005. [Online]. Available: https://tools.ietf.org/html/rfc4158

[11] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3." Internet Engineering Task Force, 2018. [Online]. Available: https://tools.ietf.org/html/rfc8446

[12] R. Rivest, A. Shamir, and L. Adleman, "Cryptographic communications system and method." Google Patents, 1977. [Online]. Available: https://patents.google.com/patent/US4405829

[13] "Sec 1: Elliptic curve cryptography." Certicom Corp., 2009. [Online]. Available: https://www.secg.org/sec1-v2.pdf

[14] J. Daemon and V. Rijmen, "AES Proposal: Rijndael." National Institute of Standards and Technology, 1999. [Online]. Available: https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf

[15] D. Harkins, "Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES)." Internet Engineering Task Force, 2008. [Online]. Available: https://tools.ietf.org/html/rfc5297