PROFORMA: Proactive Forensics with Message Analytics

Amarnath Gupta
San Diego Supercomputer Center
Univ. of California San Diego
La Jolla, USA
a1gupta@ucsd.edu
Subhasis Dasgupta
San Diego Supercomputer Center
Univ. of California San Diego
La Jolla, USA
sudasgupta@sdsc.edu
Aditya Bagchi
Computer Science/Data Science Dept
Ramakrishna Mission Vivekananda University
Belur, India
bagchi.aditya@gmail.com

Abstract

Digital forensics has traditionally been the study of methods to recover and investigate material found in digital devices that are examined to solve crimes involving the computer and the internet. However, we posit that there is a growing class of criminal activities, especially internet fraud, that can effectively utilize computational forensic techniques for *proactive prevention*. Prevention-minded analytics can be performed effectively through contextual trust analytics conducted atop modern data and knowledge technologies that can process and infer from a wide variety of information categories found in social media exchanges. We present the sketch of a system called PROFORMA that continuously evaluates the trustworthiness and risk of social communications. PROFORMA uses an underlying polystore-based data management system to store historical communications, observable social network of the victims, together with domain knowledge obtained from the social context of messages, which informs the computation of trust and risk, leading to preventive actions.

Index Terms

trust, risk, polystore, heterogeneous data, digital forensics, Internet scam

I. Introduction

Digital forensics is a branch of forensic science encompassing the recovery and investigation of material found in digital data. It ventures to search, collect and analyze evidences from electronic media – disk storage, mobile devices, computer networks as well as social media data that may be relevant for the investigation. However, a forensic investigation is a retrospective activity – it usually starts after the crime is committed. In real-life cases we know these investigations analyze the message history from Facebook and mobile phone and the call logs of the phones used in communication to triangulate the criminals. Could we, using the same investigative principles of digital forensics, develop automated techniques to *prevent* the crime before it actually happens? In this paper, we take the position that it is possible to develop technology to monitor and analyze traces of digital events and create a warning system that would caution a victim before a fiscal and psychological attack takes place. We use the term **proactive forensics** to refer to a prevention-minded digital forensic methodology.

Proactive forensics can be applied for online frauds, like inheritance schemes, lottery/prize/sweepstakes schemes, online sales schemes, bank and financial account schemes and romance schemes which are inherently different from traditional forms of fraud. These types of frauds capitalize on prolonged communication between a victim and an adversary – interactions that may spread over multiple channels (Internet, mobile phones, emails) and may be publicly visible or private between the parties. A crucial feature of these types of scam is that the victim directly partakes in the execution of the crime by progressively revealing sensitive information to the adversary, often through a protracted series of exchanges over time. As the victim feels a gradual growth of trust and "closeness" with the attacker, the disclosure of sensitive, private information seems "natural" in the context, and the victim never suspects a breach of privacy. The goal of proactive forensics is to make use of these salient characteristics of human communication toward crime prevention by issuing *early warnings* to a prospective victim.

We believe that the time for proactive forensics is *now*, because of three recent and converging advances in social media, data management and preventive technologies respectively:

1) Thanks to public APIs from social media companies like Facebook, it is possible to obtain traces of activity for user. Hence, with the right access privilege, both current and historical data (chat history, one's social circle, URLs, phone numbers . . .) can be analyzed.
2) While the digital evidence is heterogeneous in nature (e.g., social media, emails, SMS messages), and has different data models (e.g., text, JSON and networks), information integration techniques are now mature enough to integrate and correlate *multi-model* data efficiently within a single system.
3) Decision-making technologies now include the ability to estimate trust values of messages and individuals, measure the potential risk involved in transferring sensitive messages and develop a mechanisms to study trust and risk interplay between a potential victim and an adversary. Therefore, now it is feasible to design heterogeneous analytic schemes that would analyze heterogeneous data and produce a "suspicion" (i.e., compute an anomaly score), which can be used to trigger a legitimate warning.

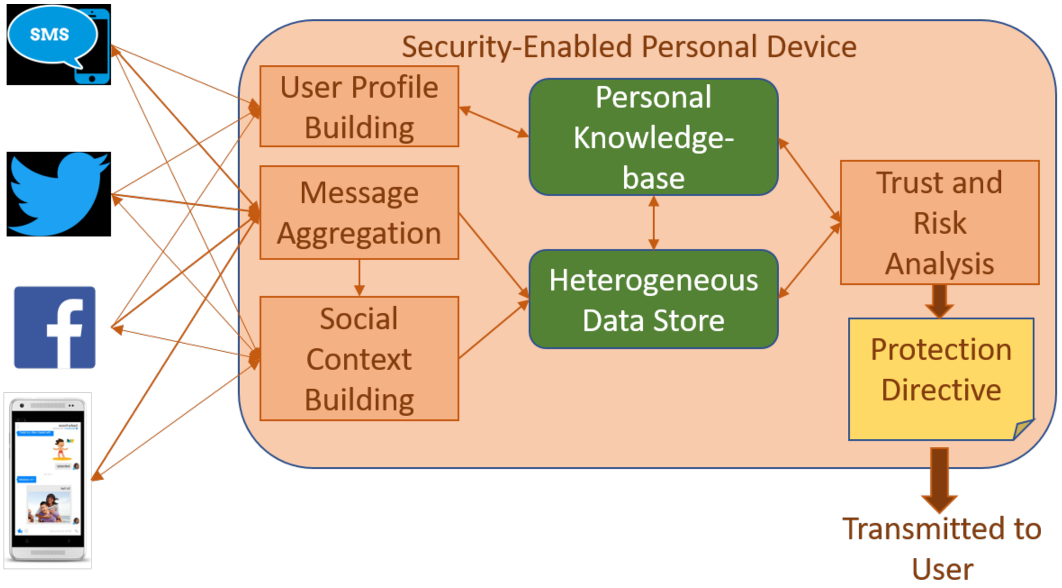

Fig. 1. A high-level architecture of a proactive forensics system

II. Bird's Eye View of a Proactive Forensics System

The basic processes behind a proactive forensics system is depicted in Figure 1. The system resides in a secured personal device as shown here, or alternatively, is split where the green components are connected to a remote server through a secured connection, and the light brown components are located in the personal device. A subject (i.e., user) must give the system explicit permission to access relevant social channels like SMS, Facebook and Twitter so that the actions of the software are not considered a breach of privacy and access control policies of the social channels.

Once commissioned, the system uses the API of these sources to construct a combined profile of the subject by drawing his/her individual profile information from all connected sources. The *Social Context Builder* attempts to reconstruct the visible (based on the subject's permissions) part of her social network over all media channels, and store this information in a personal knowledge base. The *Message Aggregator* scans messages from different sources, places them in a *Heterogeneous Data Store* (e.g., the AWESOME polystore [1]) for analytical operations occurring downstream. The *Trust and Risk Analysis Module* is the heart of the prevention mechanism. The risk analysis involves computing a trust score for each message (or a set of messages depending on the configuration). More importantly, it monitors the responses written by the user and assesses the risk associated with the user's message based on the content of the message, the trust of the receiver and the history of trust and risk computed over the lifetime of exchanges between them over all message channels. For example, divulging the security code of a locked gate to a suddenly-turned-romantic friend of a friend may be very risky. If the message is evaluated to have high-risk, a *Protection Directive* – a statement that says which part of the subject's response is high-risk, along with a link that explains why the system made the assessment – is immediately created for the user. Clearly, these directives for every message can be a significant burden both on the system and the subject, and to be practical, it would allow the user to create a "safe list"(often found in email systems like Microsoft Outlook), as well as a risk threshold (like safety thresholds in web browsers) below which the directive will not be issued.

III. Building User Profiles

A user profile is a collection of verified or verifiable facts, i.e., data records, about the user (i.e., the person being protected). These facts are collected using the respective APIs of the social networks accessible to the system, and are stored in a secure personal knowledge base. Examples include birthdays, workplace details and academic qualifications. In a more aggressive scenario, it can also include other publicly available facts, such as the properties bought and sold, current and prior addresses and names of current and former spouses. The facts can sometimes be related – one can have a credit card account that is related to a job function. We use the term knowledge-base (instead of database) for the user profile because we can perform a number of reasoning operations on the collection. *Consistency checking* is a common form of reasoning. If some collected fact logically conflicts with another, the system will try to assert them both and fail due to logical contradiction. It will then first try to verify them by asking the user directly. This style of direct verification has become more common lately – Researchgate (`www.researchgate.net`) verifies if a member is indeed the author of a publication, and some credit card systems ask users to verify a previous address not directly supplied by the user to the system. If however a contradiction is derived but remains unexplained, it is considered a violation. In this case the conflicted facts will internally retained with a conflict flag. In the proof-of-concept PROFORMA system, these collected facts are stored as instances of knowledge encoded in RDF/OWL (`https://www.w3.org/OWL/`), a World Wide Web standard for representing semantic information. RDF/OWL has a well-defined formalism with a specified set of *logical entailments*, i.e., derivation rules, which can be used to detect contradictions to easily-verifiable rules. In addition, one can specify application-specific "business rules" that can specify derived facts that may (or may not) be computed in the system. We witnessed a real-world instance of an insurance scam, where the victim was incorrectly convinced that he had a lapsed insurance policy, and he would get money back as soon as he paid a "small" lapsing penalty. In this case, the fraudulent party asked the potential victim to "verify" his address on record. While they correctly presented the would-be victim's current address, they did not recognize that the person had moved and the current address could not have been valid when the alleged policy was issued. Thus, if the personal knowledge-base has the fact livesAt(Joe, ``123 Elm St, Modesto, CA 99999'', [3/1/1995-10/14/2006]) and also knows the Joe does not have to travel for his job. If the knowledge-base has the rule

$$error \longleftarrow livesAt(p : Person, a1 : Address, T1 : TimeInterval) \ \wedge \ not(p : TravelingPerson) \wedge \\ livesAt(p : Person, a2 : Address, T2 : TimeInterval) \ \wedge not(equal(a1, a2)) \wedge overlaps(T1, T2)$$

whereby a person who does not travel cannot live at two different addresses at overlapping time periods, then, the addition of a new fact
livesAt(Joe, ``456 Palm St, Modesto, CA 99969'', [8/10/2005-9/19/2014]) will lead to a contradiction.

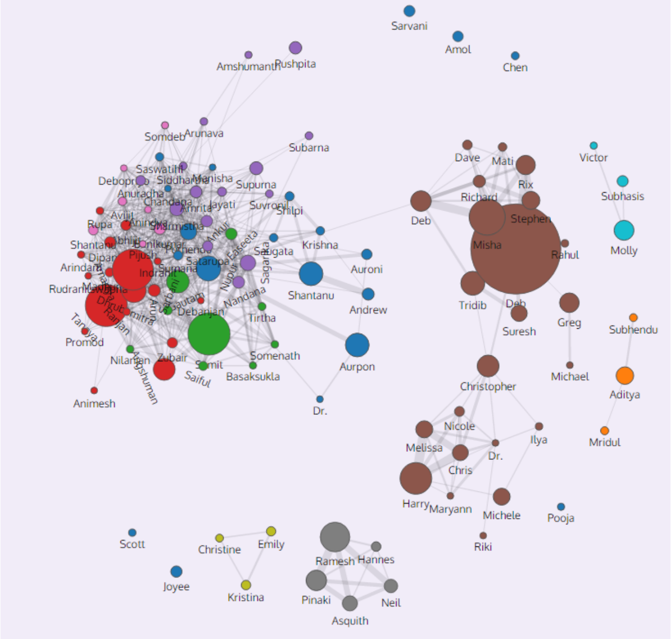

Fig. 2. The gmail-based ego network of author Gupta for the last three years. The node size represents the volume of emails received by contact person. Nodes of the same color represent a cluster.

IV. Building Social Context

The ego network of a user constructed from her friends and followers on Facebook (or any other social channel like LinkedIn or Whatsapp) forms a social context. In our design, we consider the graph to be a *property graph*, where the nodes and the edges of the graph are typed and have a set of properties ({attribute, value}) pairs, associated with them. In these cases, the nodes of the network can be *people, organizations, places, job positions* etc., and the edges represent relationships like *friend, child, colleague, work-institution* and so forth. Figure 2 shows the gmail ego network of author Gupta over the last three years. The nodes may have further properties including the names, IP addresses, network IDs etc. while the edges may include properties like measures of closeness with the user, derived from the average number of messages exchanged per week; other properties may include the duration of the relationship, the histogram of messages exchanged over time, and so on. Social graph of a user is constructed when he/she registers with the system for the first time. The social graph construction process performs a merger of the social networks of the new user available from every channel. The social context graph is then stored in the heterogeneous data store which contains a component for efficient storage of graphs. Next, the "bare" network is enriched by estimating an *initial trust value* for every *Person* and *Organization* node. The trust-value for a node is assigned by combining (i) the extent of communication between the user and the node, where greater communication (or a close family link) implies a higher trust, (ii) the link strength, which computes the relative importance of the node with respect to the user based on a measure of common nodes between the neighborhoods of the user and the node in question, and (iii) the quality of message exchange between them, as measured by topic similarity and consistency in communication threads between the user and the node. The individual scores from these component measures will be combined and normalized to a [0, 1] scale. Additional factors can be computed by matching the static properties (public profile details) of the node with the profile of the user employing the software. Aside from initial trust assignment, the composite trust measure computed from these factors can also be used for screening. Initial trust assignment may depend on subjective judgment of a user. For a close friend or a relative, user may like to assign a high initial trust value even without computing against any static parameters. For a ”friend request” from an unknown person, however, needs to compute a trust value against matching of profile parameters and also to consider the number of mutual friends. Usually, for matching profiles of two members, a parameter is considered to be matching only when the values at both ends are semantically same [2]. On the other hand, since in our case the profile information is maintained in an ontology structure, partial matching is possible. Figure 3 shows the comparison of qualification of 3 member nodes A, B and C where A and B are both engineers with degree in Civil and Mechanical engineering respectively and C has a degree in English Literature. If the qualification of A is compared with those of B and C with a matching value lying between [0, 1], usually it would show 0 in both the cases. However in our system, since both A and B are engineers, there will be a partially matching score between 0 and 1. Score against mutual friends is computed as a ratio N/T where N is the number of mutual friends and T is the total number of friends. So, for a new “friend request” from an individual having very few mutual friends and insufficient public profile data may automatically have a very low initial trust value and may be recommended for rejection from the proactive forensics system. We are considering of assigning threshold values for both profile matching score and N/T ratio so that anyone having values below threshold in both the cases will be automatically rejected and anyone having such score in anyone of of the two cases will be left to the user for decision.

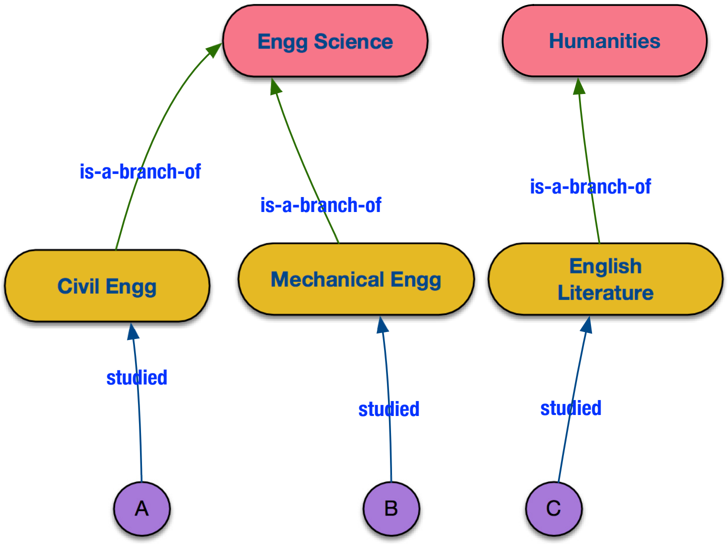

Fig. 3. An ontological representation of user profiles may assist in partial profile matching.

An important goal of the system is its ability to identify potential “friends” that might have a fake or deceptive online profile. The problem of fake profile detection on online social networks have been extensively studied since 2006 (see survey by Fire et al [3]). Most of these techniques rely on anomalous topological properties of the network [4] and how it grows [5]. A “friend” who creates an increasing number of friends from the user’s own ego network within a short period of time creates an *infiltration attack* or *Sybill attack* [6] – such an attack can be analyzed from the graph store and the initial trust value of someone with this behavior can be sharply reduced. A different but related measure is to compare the growth of one individuals contact formation rate with respect to the rest of the nodes in the same network.

Yet a third pattern of establishing contact by a predatory user is to rapidly cross channels, e.g., from a social network to an instant messaging channel. These patterns are computationally detected by Social Context Understanding process.

V. The Message Aggregation Process

The ability to automatically detect problematic messages is a fundamental capability of a proactive forensics system. A message, regardless of whether it is an email, FB post, SMS message etc., has a core information structure that consists of:

1) metadata about the message including type of data, character encoding, date and size of message
2) the sender and receivers of the message
3) whether the method belongs to a thread and if so, the prior message it refers to
4) the body of the message, either plain or rich text object, that contains semantic references (e.g., syntactic tokens like hashtags and user references, lexical structures like URLs, lexico-semantic structures like phone numbers and date of birth, as well as semantic entities like the names of people, organizations and locations.
5) auxiliary entities like images, audio and video files

The message aggregator translates individual message formats from different sources into a common internal form that it stores in a component store specialized for semistructured[1] data (like JSON), that has a strong text content.

Next, the message is processed through an analysis workflow. The goal for these analysis is not message understanding, but identifying whether it is a trust modifier. We contend that although full-scale NLP processing that identifies the phrasal structures and named entities of text is ideally a better option, parsers like Stanford's CoreNLP works well when the textual content of a message is relatively error-free. Since correct spellings and grammar cannot be guaranteed for most casual messages, we take recourse to a *structure-spotting approach* instead. Accordingly, as messages get collected, the message aggregator uses a number of pattern recognizers to identify potentially important structures contained in the data (e.g., phone number, names), as well as potential trust-violating sentence and word patterns (e.g., "send . . . Moneygram . . ."). Towards this task, the system maintains a *fraudulent pattern dictionary*, that is regularly updated by the system based on fraudulent phrases encountered by a global service, similar to virus signature updates used in virus protection software. In the simplest case, the pattern dictionary may be expressed as a regular expression pattern. A similar approach has been adopted in a recent research effort [7] to identify *Indicators of Compromise* (IOC) in messages where candidate IOC terms are initially identified by text processing and a dependency graph of the IOC-bearing sentences is analyzed and classified to determine the significant IOC terms and the relevant text context in which they occur.

When these structures are recognized, the aggregator computes a trust value for the current message from a cache of the prior messages exchanged between the user and the message sender in the current and if needed, prior sessions that have been stored in the system. If the message contains an *identifiable reference* to known entities and events, this information may be additionally used to update the computed trust value. For example, if a message refers to an insurance company, and claims that it provides auto insurance, the system can look up the insurance company on the web. If the insurance company does not exist, the unverifiable information reduces trust. The problem becomes more complex if the insurance company exists in real life, but does not provide auto insurance. In this case, the system may either determine it automatically by searching for the key phrase "auto insurance" on the web site of the company – if it succeeds, the trust value remains unaltered. If it fails, the system seeks the user's help in locating the information on the web site. If the user cannot find it, the system marks the conflict in its knowledge-base and reduces the trust of the message sender. A different situation warrants a *aspect-level sentiment analysis* [8] of messages, where the goal is to find and aggregate sentiment on entities mentioned within

[1] A data collection is considered to be *semistructured* if it can be modeled as a tree or an acyclic graph such that one data element can occur "under" another and the number of branches of a tree (or DAG) node can vary across data items.

documents (i.e., messages) or aspects of them. For example, the history of messages may indicate that the "friend" may have an excessive eagerness towards the user's financial assets or valuable possessions. Such personality traits can be measured using a combination of sentiment identification methods [9], and then be scored with respect to the user's full social context to determine excessiveness. Other models of trust computation and update have been proposed in literature based on social engagement as well as popularity of an individual. But we consider interaction-based trust computation to be more important for social media frauds. In this category of trust models, Švec *et al* [10] provide an update rule for engagement trust based on interaction time span, number of interactions, interaction regularity, number of characters in each communication, common interest etc., and developed a trust aggregation function to compute a composite trust from these factors. This approach is somewhat similar to our formulation where we also use different factors like *interaction time span*, *number of interactions*, *frequency of message transfer* etc. for computing trust. Certain anomaly in communication is also detectable during computation. For example, we borrow the trust measure for "number of interactions" and present it as:

$$A = \frac{1}{n} \times \sum_{i=1}^{n} I_x \tag{1}$$

$$T_x = \frac{I_x}{(A + \frac{1}{n} \times \sum_{i=1}^{n} |A - I_x|)} \tag{2}$$

where I_x represents the number of interactions my user is having with another member x and A is the average number of interactions among n such members. Computation of trust T_x shows that a sudden surge of interactions over the average A will reduce the trust value suspecting anomaly in interaction on the part of member x. However, our system will only generate warning messages for the user which the user can ignore if required.

VI. Progressive Analysis of Trust and Risk

In the Message Aggregation process, computation and re-assessment of trust of messages and message senders have been proposed as a function of current message and the messages already exchanged. As explained earlier, the proposed proactive forensic effort normally expects a long exchange of messages between a possible adversary and a potential victim (user of our system) covering a large time period. In the process, the adversary gradually earns increased trust from the possible victim. This increase in trust may cause the victim to divulge sensitive information to the adversary inadvertently. So as the trust on the adversary increases, the possible victim runs into the risk of revealing sensitive information. Unlike trust, *risk* refers to the likelihood that a specific information contained in a message m about to go out from the user (the possible victim to-be-protected) will reduce her level of security (or increase a material damage to her) with respect to the adversary receiving the message. In this sense, it roughly corresponds to a conditional probability of security loss where the conditioning factors are the history of personal information already available to the adversary and the *current* level of trust of the adversary. The proposed system needs to compute both the progress on trust, risk and their inter-play to generate appropriate warning message for the user of our system, the potential victim in this case.

A machine learning based risk assessment technique was developed in [11] for user actions like adding a friend, and sending messages in a social media setting. In this formulation, risk indicators are associated with an information item (i.e., the user action), and can be evaluated based on four factors: (i) *criticality level*, measuring how important the action is, (ii) *likelihood level*, measuring how expected the action is under similar situations, (iii) *impact*, which refers to the impact of security risks, and (iv) *information requestor reputation*, which roughly corresponds to trust of the possible adversary in our framework. Based on these factors, [11] classifies the information item into k risk levels. The first two factors, namely criticality of action, which is an estimate of how important it is for the user to provide the information contained in the message, and likelihood level, which requires the system to have a compendium of all situations and likely user responses, are hard metrics to compute. It is evident from the discussion so far that trust computation has to be done from the parameters like, duration of friendship, frequency of communication, quality of communication and possible agreement in different posts etc. On the other hand, risk involved in the relationship will primarily be reflected from the content of the different messages sent by a potential adversary. Types of trust and risk functions considered here are mainly to analyze the presence of a possible adversary so that any risky communication revealing sensitive information can be detected and the possible victim can be warned accordingly. In general, however, social media carry many useless relationships with either very few communication or exchange of messages with gross disagreements on several issues. Any trust or risk functions designed for such cases may be different from the issues considered in this paper. Here we assume the trust between a potential victim and a possible adversary is built up slowly and gradually through long chain of exchange of messages. Considering the communications in congenial mood here, we have taken the number of messages exchanged as the primary parameter for increase in trust. Gross disagreement in issues as reflected in the content of the messages exchanged or possibility of divulging sensitive information through messages will be parameters for computing risk. Accordingly, we have considered the change in trust in the continuous domain as an exponential function bounded by (0,1). This approach conforms to a recent work that has provided a comprehensive survey on trust evaluation in social network [12]. it has been shown that a popular method of modeling trust is as a *subjective probability* by which one user expects another user to perform a given action. It conforms to the idea of building social context proposed earlier where all components related to trust are combined and normalized to a [0,1] scale.

However, value of trust in a new friendship may not start from 0 but from the initial trust value computed at the time of accepting a friendship request. Nature of the trust function for two different persons is shown in Figure 4. The trust function has been considered as:

$$T = max(I, (1 - \alpha^n))$$

where, T = Current trust value, n = number of messages, α = learning rate.

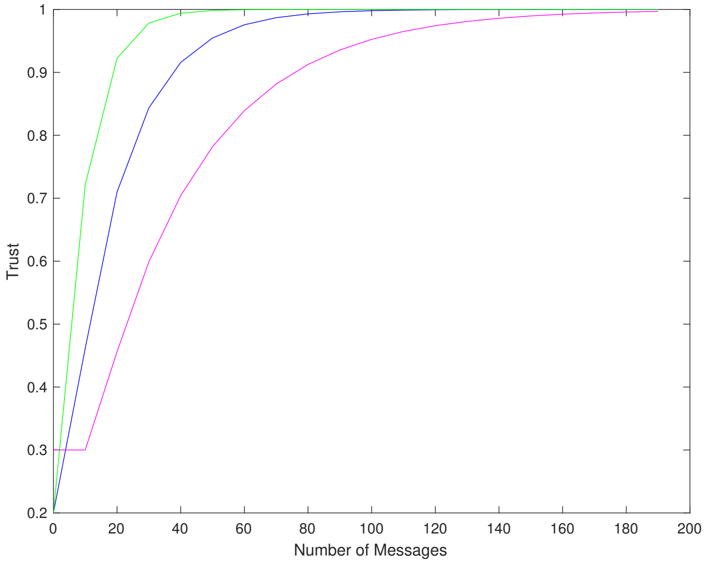

Fig. 4. Rising Trend of Trust with Different Learning Rate.

As shown in Figure 4, two persons having same initial trust value may not have the same rising trend in trust for different learning rate α. At present we have considered initial trust value be computed as a composite score resulting from Profile Matching and Social context graph matching. In addition, person accepting a friend request may add a personal bias as a subjective judgment depending on how much he/she knows the requester. As a result, friend request received from a close relative or close friend would carry a significantly high initial trust value. However, initial trust value once computed would be considered constant. Change in social context and quality of communication (degree of agreement in different issues or sentiment matching) taken together will constitute the learning rate. Therefore, though in Figure 4, the value of α controlling the rising trend in trust has been shown as constant, in actual practice, it may vary. Paucity of space prevents us from providing detailed computation of trust function along with all its parameters.

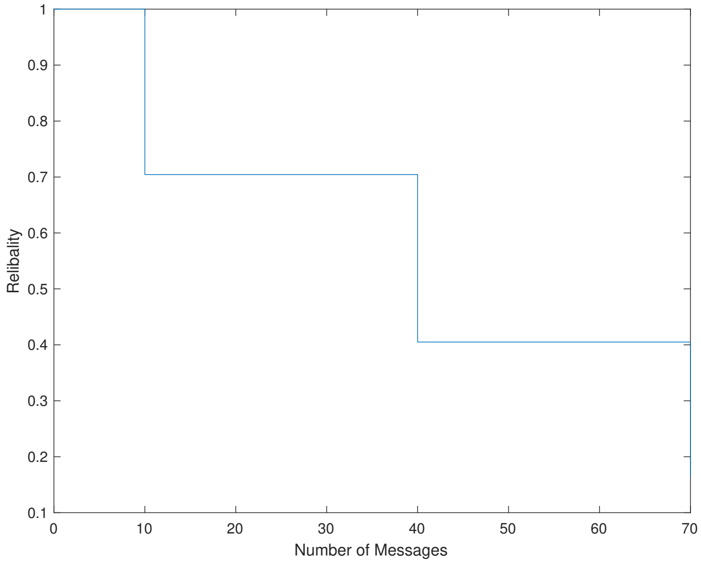

Fig. 5. Change in Reliability with Appearance of Sensitive Terms

In our system, change in risk has been modeled as inverse of change in reliability. In other words, in a domain of (0,1), Reliability = 1- Risk. Moreover, similar to the effort made in [7] to identify *Indicators of Compromise* (IOC), we also create a *Dictionary of Sensitive Terms* where each such term is associated with a cost, once again in the domain of (0,1). So each such appearance of a sensitive term in the message sent by a possible adversary would reduce the reliability by the cost of the term appeared, as available in the *Dictionary of Sensitive Terms*. So, the change in reliability as modeled here would be discrete, depending on the appearance of sensitive terms in the message received. The situation is explained in Figure 5. Initially, risk has been considered to be 0, hence the reliability is 1. As soon as a sensitive term i appears, the reliability is reduced by C_i, the cost of the term i. Figure 5 shows the gradual reduction in reliability and thereby increase in risk with the appearance of different sensitive terms of different costs. Figure 6 shows the composite effect of risk and trust, effectively modeled as reduction in reliability with increase in trust. So at the time instant, t_i appearance of a sensitive term of cost C_i reduces the current trust value T_i as:

$$T_i = T_i - C_i$$

Value of trust then increases again following the earlier function. This Risk and Trust Interplay is represented in Figure 6. If this interplay brings down the trust below a certain threshold, system generates a warning message. For our experiment, we have considered the threshold as the initial trust value. Initially, our system was modeled to generate this warning only when the predefined threshold is crossed. However, later we found that a modification is required. For a close relative or friend, asking for e-mail address, phone no.or home address may be normal even when they are considered to be sensitive information and will reduce reliability or increase risk with respect to the requester. So appearance of each sensitive term

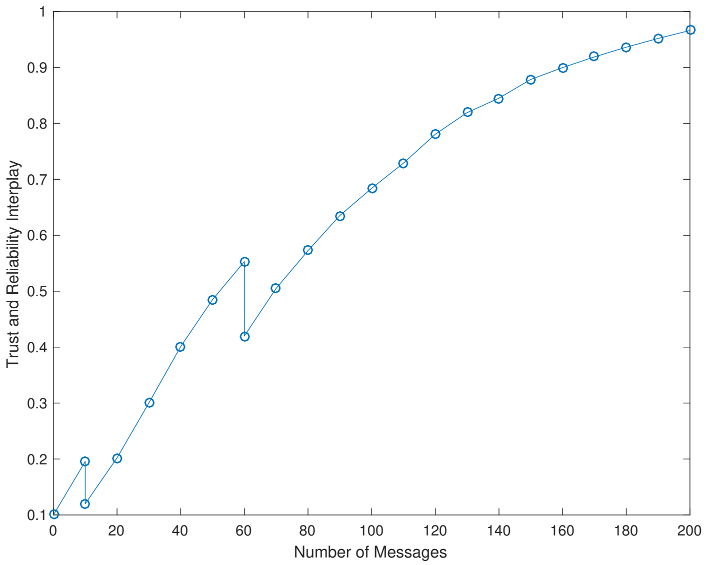

Fig. 6. Reliability and Trust Interplay

will be notified to the receiver with option to ignore it so that trust value remains unaltered. However, warning message will advise the receiver to stop further communication with the requester declaring him/her as a possible adversary.

VII. The Privacy Issue

Though the proposed system is for reducing cyber-crime and to generate an early warning against any such possibility, it offers a great challenge to avoid breach of privacy in communication between two persons on social media. Two important survey papers covering different aspects of security and privacy in social network have provided details of the privacy issues that may be encountered in communication over social media [13], [14]. Out of the different attacks discussed in the two surveys, the relevant issues for the present system are: *Attack from other users* and *Crawling attack*. Attack from other users is the phenomenon discussed so far in this paper. As a matter of fact, proactive forensic system has been proposed only to avoid such attacks and to generate an early warning system against the possibility of such attack. On the other hand, for accepting a friendship request profile matching and social context building methods described so far constitute crawling attack. Profiling of a requester can be done only by exploiting publicly available $APIs$ of different social media. So crawling through the web sources and consolidating data available from those sources, it may be possible to build a good amount of information about a person requesting for friendship. Most of those who are sending friendship requests are not potential adversaries but the profile matching and social context building will be done for all requesters. In other words, from privacy point of view, the proposed system creates a situation where in order to prevent *Attack from other users* the system will create a *Crawling attack*.

However, initial screening of friendship requests is essential to avoid increasing cyber scams. So as a compromise, the proposed system plans to offer a message to all friendship requesters that the receiving side has a proactive forensic system installed. It would imply that a possible crawling on social media will be done for profile matching and social context building. A requester on the other hand can always decide how much personal information will be kept in public domain to be exploited by web crawling. Even some social network system by default, controls the extent of public information to be maintained. For example, LinkedIn discloses very little profile information of a requester till the other side accepts a friendship request. On the other hand, Facebook leaves it to user's discretion. It is necessary that users of different social media should also be aware of these privacy related issues. Researchers like [15] have developed questionnaires and predictive models to assess an individual's degree of privacy concern, level of privacy awareness and the proclivity toward self-disclosure.

One possible solution to mitigate the *Crawling attack* issue can be to accept a trusted third party for both requester and receiver and such an agency can take the responsibility of profile matching and social context building. However, more detailed study on this *Crawling attack* issue is yet to be done.

VIII. PROFORMA: A Proof-of-concept System

We are developing PROFORMA, an initial functional prototype that incorporates the ideas described above. PROFORMA is being implemented as an application above the AWESOME system [1], where tasks like building the user's profile, social context and message aggregation are handled by the application layer, and the heterogeneous data store is implemented through AWESOME. AWESOME can accept real-time data from social media APIs, as well as "static data" like dictionaries and metadata about the user. All data ingested into the system are transformed and placed in an appropriate data store. The PROFORMA application uses AWESOME as its data layer and uses ADIL, AWESOME's scripting language to define the dataflow for each process. For instance, the trust computation needs to use the history of the user with a set of "friends", the application uses an API call to retrieve a friend's messages, reverse sorted by time. The system can be configured to run in an in-memory mode as well as a full-scale data management system over distributed cluster of machines. These modes of operation are important for the PROFORMA application – which can run either in a standalone, single user mode, or as a multitenant service mode. The personal knowledge-base is developed with Scigraph (https://github.com/SciGraph/SciGraph), an ontology management engine. Scigraph accepts OWL knowledge-bases and internally represents it as a graph database in a separate Neo4J instance, also stored in AWESOME. Ontological queries are expressed as graph queries in SciGraph using a REST API.

References

[1] S. Dasgupta, K. Coakley, and A. Gupta, "Analytics-driven data ingestion and derivation in the AWESOME polystore," in *Proc. of the IEEE Int. Conf. on Big Data*, pp. 2555–2564, IEEE, Dec. 2016.

[2] J. Mcauley and J. Leskovec, "Discovering social circles in ego networks," *ACM Trans. on Knowledge Discovery from Data (TKDD)*, vol. 8, no. 1, p. 4, 2014.

[3] M. Fire, R. Goldschmidt, and Y. Elovici, "Online social networks: threats and solutions," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2019–2036, 2014.

[4] M. Fire, G. Katz, and Y. Elovici, "Strangers intrusion detection-detecting spammers and fake profiles in social networks based on topology anomalies," *Human Journal*, vol. 1, no. 1, pp. 26–39, 2012.

[5] C. Xiao, D. M. Freeman, and T. Hwa, "Detecting clusters of fake accounts in online social networks," in *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, pp. 91–101, ACM, 2015.

[6] S. Y. Bhat and M. Abulaish, "Using communities against deception in online social networks," *Computer Fraud & Security*, vol. 2014, no. 2, pp. 8–16, 2014.

[7] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. Beyah, "Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence," in *Proc. of the ACM SIGSAC Conf. on Computer and Communications Security*, pp. 755–766, ACM, 2016.

[8] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 813–830, 2016.

[9] S. Poria, A. Gelbukh, B. Agarwal, E. Cambria, and N. Howard, "Common sense knowledge based personality recognition from text," in *Mexican International Conference on Artificial Intelligence*, pp. 484–496, Springer, 2013.

[10] T. Švec and J. Samek, "Trust evaluation on facebook using multiple contexts," in *21st Conference on User Modeling, Adaptation, and Personalization*, pp. 1–10, 2013.

[11] A. Ali-Eldin, J. van den Berg, and H. A. Ali, "A risk evaluation approach for authorization decisions in social pervasive applications," *Computers & Electrical Engineering*, vol. 55, pp. 59–72, 2016.

[12] W. Jiang, G. Wang, M. Z. A. Bhuiyan, and J. Wu, "Understanding graph-based trust evaluation in online social networks: methodologies and challenges," *ACM Computing Surveys (CSUR)*, vol. 49, no. 1, p. 10, 2016.

[13] E. Novak and Q. Li, "Security and privacy in online social networks - a survey," Tech. Rep. WM-CS-2012-02, Department of Computer Science, The College of William and Mary, 2012.

[14] I. Kayes and A. Iamnitchi, "A survey on privacy and security in online social networks," *arXiv preprint arXiv:1504.03342*, 2015.

[15] H. Krasnova and N. F. Veltri, "Privacy calculus on social networking sites: Explorative evidence from germany and usa," in *Proc. of 43rd Hawaii Int. Conf. on System sciences (HICSS)*, pp. 1–10, IEEE, 2010.