International Journal of Marketing and Technology
Vol. 15 Issue 02, February 2025
ISSN: 2249-1058 Impact Factor: 6.559
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gate as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Real Time Scam Detection for end to end encrypted messaging using Machine Learning

Anuja More*, Sumit Shyamsukha, Rupesh Kumar*****

Abstract (12pt)

The recent rise and proliferation of messages-based mass marketing through enterprise cloud services (e.g. Google RCS, WhatsApp API) has empowered malicious actors too. Malicious actors often abuse systems of scale (e.g. API) to send fraudulent (aka scam) messages to a large number of people. These messages may be encrypted by the sender, which introduces additional challenges to detect abuse. This needs a real time scam detection system that can conduct intermediate scanning by using advanced machine learning models to detect and enforce in a timely manner to prevent propagation of user harm. This scanning is architecturally viable specifically in enterprise messaging settings where part of the message is generated on the client side while part of it is populated in the cloud.

Keywords: Scam, Fraud, API, Fraud Detection, ML model, real-time, end to end encryption

A well-prepared abstract enables the reader to identify the basic content of a document quickly and accurately, to determine its relevance to their interests, and thus to decide whether to read the document in its entirety.The Abstract should be informative and completely self-explanatory, provide a clear statement of the problem, the proposed approach or solution, and point out major findings and conclusions. The Abstract should be 100 to 200 words in length. The abstract should be written in the past tense. Standard nomenclature should be used and abbreviations should be avoided. No literature should be cited.The keyword list provides the opportunity to add keywords, used by the indexing and abstracting services, in addition to those already present in the title. Judicious use of keywords may increase the ease with which interested parties can locate our article (9 pt).

Author correspondence:

Rupesh Kumar,
Product Manager
Email: Rupesh.kumar25@gmail.com

1. Introduction (12pt)

Have you received a text on your favorite messaging app saying your USPS delivery is stuck and you need to pay using an unknown web link Or that you could make $1000s of $$s in days by trading crypto Or an enticing offer of an amazon job?

These seem trivial for the educated and tech savvy recipient but to the unbeknownst user its trouble, more commonly known as scam. Receipts of these messages end up losing their life savings! In 2022 alone, Americans lost over $48 Billion dollars to online scams [2]. A recent report by the Indian Cybercrime Coordination Centre revealed that digital financial frauds accounted for a staggering ₹1.25 lakh crore over the last three years [3]

A scam is a dishonest plan to trick someone into something which usually involves money.[1]

The discovery and rise of automated messaging platforms (APIs) that enable businesses to send 100s of millions of email, sms, whatsapp messages etc sent with low efforts for marketing, one time codes and other use cases has exacerbated spam and scam potential. Traditional methods for detecting and preventing such activities often involve analyzing message content for patterns indicative of fraud. However, the advent of end-to-end encryption has posed significant challenges to these conventional approaches. Here's how end to end encryption works in enterprise settings where part of the message generated on the client side is not encrypted while part of it populated on the cloud is:

A. **Message on client side that is reviewed for scam proactively:**
Your package has been shipped. It will be delivered in {{1}} business days.

B. **Message that could be sent by scammer through cloud encryption by replacing the *{{1}}* parameter above during send time:**
Your package has been shipped. It will be delivered in 😀😀😀😀👇 👇 👇 👇Dear Friend, as the market starts to recover, we invite you to join the internal discussion group of the professional investment team. The group will post daily trading signals and teach you how to make great profits in the cryptocurrency market, If you join this group, we have a great gift for you and a chance to win 1000USD!click the link to enter 👇 👇 👇 👇 👇 👇 👇 👇 👇 business days.

End-to-end encryption ensures that only the recipient user can read the messages, thereby preserving privacy. Consequently, senders (cloud services) have limited capability to monitor and mitigate scam messages, as they cannot access the encrypted content of the messages being transmitted through their systems. This limitation has led to a need for innovative solutions that can effectively detect and prevent scams in an end-to-end encrypted environment without compromising user privacy.

2. Research Method (12pt)

This paper proposes a novel approach to scam detection given limited research and analysis in the specific field of mass messaging and end to end encrypted enterprise cloud system. The proposal herein describes a system that can proactively and in real time scan and block fraudulent messages that are end to end encrypted in the enterprise cloud system described above, thereby protecting users from potential scams while ensuring that legitimate messages are delivered promptly and securely.

The proposed scam detection processor(s) apply a machine learning model trained on anonymized data to detect potential scam content. The processor(s) is configured to, in response to the machine learning model identifying the message as containing scam content, drop the message from the queuing system to prevent delivery to the user. It uses a non-transient computer-readable storage medium having instructions embodied thereon, the instructions being executable by one or more processors to perform a method for detecting scams in encrypted messaging in enterprise cloud system in real time. The method holds the message in a queuing system and applies a machine learning model trained on anonymized data to eventually prevent delivery. The method, in response to the machine learning model, identifies the message as containing scam content, dropping the message from the queuing system to prevent delivery to the user.

Another aspect of the system configured for detecting scams in encrypted messaging in enterprise cloud system is its- capacity to act in real time. The system includes means for holding the message in a queuing system and means for, in response to the machine learning model identifying the message as containing scam content, dropping the message from the queuing system to prevent delivery to the user.

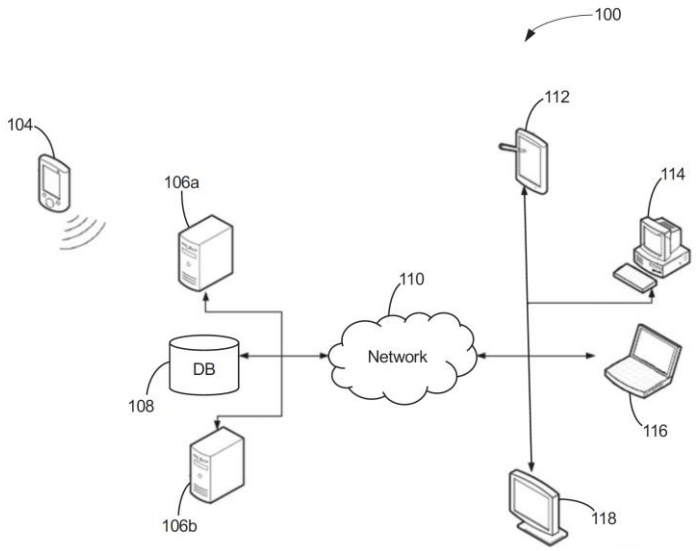

FIG. 1

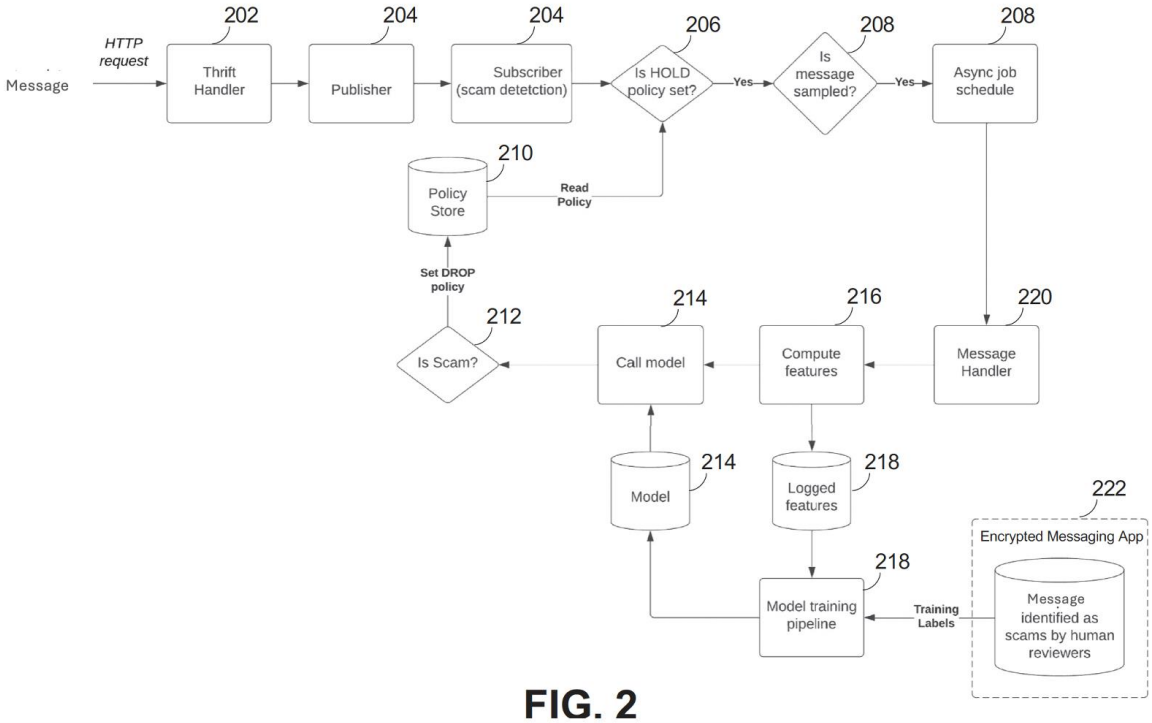

FIG. 2

FIG. 1 is a block diagram illustrating an overview of an environment 100 in which some implementations of the proposed processing can operate. The environment 100 can include one or more client computing devices, mobile device 104, tablet 112, personal computer 114, laptop 116, desktop 118, and/or the like. Client devices communicate wirelessly via the network 110. The client computing devices can operate

in a networked environment using logical connections through network 110 to one or more remote computers, such as server computing devices. The server computing devices 106a-106b is configured to show (e.g., make encrypted content in cloud visible) content to one or more of the client computing devices for those client computing devices that presented a correct public key. As an example, the server computing devices 106a-106b can include a database (e.g., database 108) that tracks which users of the client computing devices have granted access to their encrypted content (e.g., encrypted by corresponding privately held private keys in the enterprise cloud deployed by the sender) to other client users.

The environment 100 includes a server such as an edge server which receives client requests and coordinates fulfillment of those requests through other servers. The server includes the server computing devices 106a-106b, which logically form a single server. Alternatively, the server computing devices 106a-106b can each be a distributed computing environment encompassing multiple computing devices located at the same or at geographically disparate physical locations. The client computing devices and server computing devices 106a-106b each act as a server or client to other server/client device(s). The server computing devices 106a-106b connect to a database 108 or can comprise its own memory. Each server computing device 106a-106b corresponds to a group of servers, and each of these servers share a database 108 or can have their own database 108. The database 108 logically forms a single unit or be part of a distributed computing environment encompassing multiple computing devices that are located within their corresponding server, located at the same, or located at geographically disparate physical locations. The database 108 store data indicative of keys or access granted by a given user to other users of the given user's encrypted content in enterprise cloud and/or shared messaging platform content that are subscribed to by other users. For example, the database 108 is used to facilitate key rotation in a one-to-many cloud enterprise encryption architecture by causing issue of new keys when a copy of a shared key becomes compromised.

The network 110 can be a local area network (LAN), a wide area network (WAN), a mesh network, a hybrid network, or other wired or wireless networks. The network 110 may be the Internet or some other public or private network. Client computing devices are connected to network 110 through a network interface, such as by wired or wireless communication. In some implementations, the server computing devices 106a-106b are used as part of a messaging platform such as implemented via the network 110. The messaging platform can host content and protect access to the content, such as via the database 108, although the server computing devices 106a-106b of the messaging platform do not have access to private keys and can be remote/separate from the application(s) that perform key generation and content encryption in the enterprise cloud. So, the messages can only be decrypted at user devices of the sending and/or receiving user. A private key is required to decrypt the enterprise cloud encrypted messages stored at the user devices. The message can be any digital data such as text, images, audio, video, links, webpages, minutia (e.g., indicia provided from a client device such as emotion indicators, status text snippets, location indictors, etc.), or other multi-media.

End-to-end encryption in enterprise cloud setting creates a secure environment for users to communicate privately. However, this also prevents service providers from accessing message content, which is necessary for the detection of scams and fraudulent messages. Scammers exploit this privacy feature by sending scam messages which are not revealed until the message is sent. As a result, these scam messages

bypass traditional detection systems that rely on content analysis. The inability to scrutinize message content due to enterprise cloud encryption protocols leads to a significant increase in scam messages reaching users, undermining the integrity of messaging platforms and exposing users to potential harm.

Implementations described herein address the shortcomings by providing a real-time scam detection system that operates on cloud-based application programming interfaces to address the challenge of detecting scam messages in an environment with end-to-end encryption as configured on the enterprise cloud. The system utilizes a machine learning model trained on anonymized data to identify potential scam messages. Integrated into a queuing system, the system holds messages from accounts exhibiting suspicious behavior and analyze messages in real time. If a message is detected as a scam, the system prevents its delivery to the end user, thereby blocking the scam attempt. The process occurs with minimal latency to ensure that legitimate messages are not significantly delayed.

Despite this cloud enterprise encryption, the system is still able to analyze a message at the time the message is sent to detect fraudulent content. For example, fraudulent messages may state, “If you are a stock investor and want to choose high-quality and stable stock recommendations, please click on the link below, add our assistant and send a message. We will recommend accurate stocks to you at 2PM every trading day to ensure that you can maximize your returns in the investment market” or “BTC-Alpha currently has more than 300 senior analysts from all over the world, which is a large scale. So far, we have helped more than 200,000 investors realize more than 25 times of wealth appreciation through trading Bitcoin contracts. Alpha Exchange has been selected by the local government as an excellent foreign currency financial investment institution and is also the best choice for cryptocurrency investors.” The system is configured to analyze the message verbiage and syntax to determine if such aforementioned examples are indeed fraudulent without restricting data flow or decrypting the message.

FIG. 2 illustrates an example flow diagram (e.g., process 200) for detecting scams on cloud enterprise with encrypted messages, in real time. The steps of the example process 200 are described herein as occurring in serial, or linearly. However, multiple instances of the example process 200 occurs in parallel. At 202, the thrift handler receives an HTTP request containing a template message. For example, the thrift handler processes incoming HTTP requests that include various messages from businesses using the cloud API. The thrift handler then extracts the relevant data from the HTTP request for further processing by other components in the process 200.

At 204, the publisher sends the message to the system. For example, the publisher acts as a distribution point, disseminating the message to the appropriate channels within the system. This includes routing the message to the subscriber, which is tasked with evaluating the message for potential scam content.

At 206, the subscriber checks if a HOLD policy is set for the message. For example, the subscriber queries the policy store to determine if the incoming message matches any templates that have been flagged for additional scrutiny. If a HOLD policy is in place, the subscriber temporarily withholds the message from being delivered to end-users until a scam assessment is completed.

At 208, the subscriber schedules an asynchronous job if the message is sampled for scam detection. For example, the subscriber utilizes a random sampling technique to select certain messages for in-depth analysis. This involves scheduling an asynchronous job that allows the scam detection process to occur without delaying the delivery of other messages that are not under suspicion.

The policy store may be accessed to read the current policy for the message. When the subscriber identifies a message for review, it retrieves the relevant policy from the policy store to determine the appropriate action to take based on the policy's guidelines. The subscriber may decide whether to set a DROP policy for the message. For example, if the scam detection process yields a positive indication of scam content within a message, the subscriber may update the policy store to include a DROP policy for the implicated message. This policy prevents further messages from being sent until the issue is reviewed further and resolved.

At 212, the subscriber determines if the message content is a scam. The subscriber employs various analytical techniques, such as machine learning algorithms, to evaluate the content of the message. The Machine learning model/ algorithm computes features from the message content by extracting and analyzing various characteristics from the message content, such as the frequency of certain keywords, the presence of suspicious URLs, or the use of language typically associated with scam messages.

At 218, the model records the features it has computed for each message it analyzes. This logged data is used to refine the accuracy of the model over time, as well as to provide insights into emerging scam trends and tactics. For example, if the model determines that a message is likely being used for scam purposes, it may signal the policy store to update the policy associated with that message. This update may include setting a HOLD or DROP policy to prevent further misuse.

At 220, the message handler handles the message based on the updated policy by acting in accordance with the policy directives from the policy store. If a DROP policy has been set, the message handler prevents the delivery of the message, whereas if no such policy is in place, the message handler allows the message to be sent to its intended recipient.

At 222, the enteprise cloud encrypted messaging application trains the model with training labels from messages identified as scams by human reviewers. The cloud enterprise encrypted messaging application provides anonymized data to the model, which includes examples of messages that have been previously flagged as scams. This data may be used to train the model, enhancing its ability to accurately detect scam messages in the future.

3. Results and Analysis (10pt)

In initial tests, the system - made decisions on whether a message may be fraudulent in approximately 50 seconds at 95% precision and handled around 500,000 messages per day.

Over time, the system was expanded to handle >2M messages per day, while maintaining >95% precision and 50 second latency. The recall of the system increased from 75% at initial rollout to 90% over time.

The 50 second latency is dominated by infrastructure (message delivery) latency, whereas the scam check latency itself is only on the order of 4-5 seconds.

4. Conclusion (10pt)

Detecting scams in an enterprise cloud encrypted messaging system that delivers messages from a business account to its user in real time although non trivial is possible needing an orchestrated system comprising of hardware processors configured by machine-readable instructions to held or drop identified scammy messages from the queuing system by applying a machine learning model trained on anonymized data on randomly selected samples of messages being delivered to stop the propagation of scam.

Future analysis and research on this can help improve its applicability, latency, performance for even larger volumes and precision to increase efficacy of stopping scam in messaging.

References(10pt)

[1] *What is a cyber scam?* (no date) *Young Scot*. Available at: https://young.scot/get-informed/digiknow-cyber-scams/

[2] Liu, H. and FTC, S. at the (2024) *FTC issues Annual report to Congress on Agency's actions to protect older adults*, *Federal Trade Commission*. Available at: https://www.ftc.gov/news-events/news/press-releases/2023/10/ftc-issues-annual-report-congress-agencys-actions-protect-older-adults (Accessed: 16 September 2024).

[3] Reddy, B. (2024) *Digital Financial Frauds in India: A call for improved investigation strategies*, *The Hindu*. Available at: https://www.thehindu.com/sci-tech/technology/digital-financial-frauds-in-india-a-call-for-improved-investigation-strategies/article67988607.ece