



- (51) **International Patent Classification:**
G06F 15/16 (2006.01)
- (21) **International Application Number:**
PCT/US2012/056363
- (22) **International Filing Date:**
20 September 2012 (20.09.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
61/537,380 21 September 2011 (21.09.2011) US
61/557,773 9 November 2011 (09.11.2011) US
- (72) **Inventor; and**
- (71) **Applicant : FORD, David K.** [US/US]; West Side Mission and 3rd, Unit A5, Carmel, California 93921 (US).
- (74) **Agents: BARRETT, Robert M.** et al.; K & L GATES LLP, P.O. Box 1135, Chicago, Illinois 60690-1135 (US).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) **Title:** METHODS AND APPARATUS FOR VALIDATING COMMUNICATIONS IN AN OPEN ARCHITECTURE SYSTEM

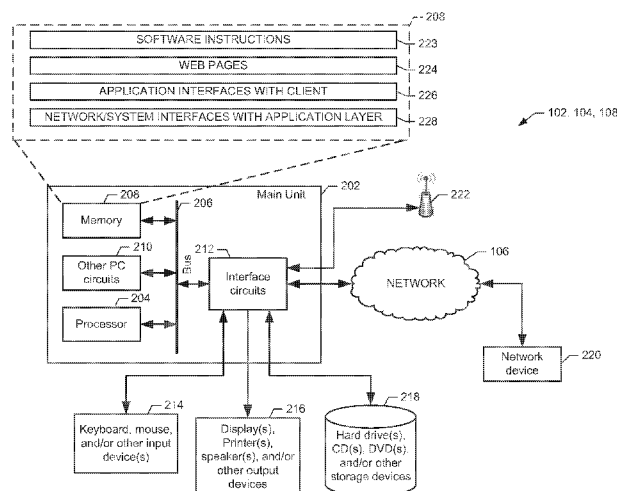


FIG. 2

(57) **Abstract:** A system, methods, and apparatus for validating communications in an open architecture system are disclosed. In an example embodiment, a method includes selecting transactional information to transmit from a server to a communicatively coupled client device based on a request from the client device, selecting presentation information corresponding to the transactional information to transmit from the server to the client device, transmitting at least one message including the presentation and transactional information from the server to the client device, determining a prediction as to how the client device will render the transactional information based on the presentation information, receiving a response message from the client, and responsive to information in the response message not matching the prediction, providing an indication there is a malicious application affecting communications between the server and the client device.

TITLE

**METHODS AND APPARATUS FOR VALIDATING COMMUNICATIONS IN AN
OPEN ARCHITECTURE SYSTEM**

BACKGROUND

[0001] Traditional communication systems address certain reliability and performance issues that arise during the transfer of information from a sender to a receiver through a medium. In an idealized situation, no errors are introduced as the information travels through the medium. As a result, the receiver obtains, with 100% fidelity, a message identical to the one transmitted into the medium by the sender.

[0002] In actual practice however, the medium is not error free. Environmental factors typically contribute haphazard information in the medium. This haphazard information is commonly referred to as “noise”. This noise can result from, for example, shot noise, neighboring radio frequencies, undesirable voltage and/or current fluctuations in circuit components, signal reflections from trees/buildings, solar flares, etc.

[0003] In information warfare, there exists a related concept of signal jamming. The idea is to increase the contribution of the noise to such an extent that it becomes practically impossible to find a set of codewords that are simultaneously robust and efficient. This type of noise is not haphazard but rather specifically crafted to render a specific medium too noisy to use. The targets of this type of purposefully crafted noise are unable to communicate.

[0004] An important purpose of traditional communication systems are to characterize a noise source and to create a set of primary codewords that are robust against that noise type. The primary codewords are designed to be efficient for communication of a wide variety of often used messages. As provided by traditional communication systems, the transmission of information through the Internet occurs over a variety of medium including cable, wireless, satellite, etc. Currently, traditional communication systems play a significant role in engineering and assuring the reliability and efficiency of those transmissions against a variety of haphazard noise sources.

[0005] Traditional communication systems have reduced the effects of haphazard noise in the communication medium as well at the sender and the receiver. For example, the sender or the receiver can include circuitry to reduce or eliminate the effects of haphazard noise. Additionally, routing devices in the medium, the sender, and the receiver can also use quality of service, data integrity, and/or error correction functions to correct for haphazard

noise. These functions can be associated with, for example, network cards and associated stacks as received packets are queued and recombined into a complete data stream.

[0006] In addition to haphazard noise, there also exists engineered malicious noise specifically created to affect, alter, or otherwise interfere with communications between a sender and a receiver. This malicious noise is an injected signal that alters codewords sent between senders and receivers in a manner that is generally not correctable by existing error correction methods of traditional communication systems. The malicious noise, created by malicious applications, are directed to interfere with communications anywhere along a communication channel through the Internet from a sender to a receiver including routers, switches, repeaters, firewalls, etc.

[0007] The malicious applications are configured to identify codeword sets and provide malicious noise that effectively switches one valid codeword for a second valid codeword. Traditional error correction schemes cannot detect this switch because they have no way of identifying that an error has occurred. The resulting altered signal is viewed as a valid codeword from the point of view of the traditional communication system. Other types of noise that commonly occur in information warfare are also deliberate and engineered (e.g. signal jamming) but the phenomena does not result in a useable codeword set.

[0008] Unlike environmentally derived haphazard noise, this malicious noise does not consist of haphazard content, nor does it disallow effective communication as a jamming signal might. Instead, this noise is specifically crafted to substitute the originally transmitted message for a second, specific, legitimate, and understandable message which is then presented to a receiver as authentic intent of the sender. The crafted noise may also occur before selected information leaves a sender (e.g., a server, database and/or directory structure) for transmission to a receiver. This crafted noise is referred to herein as malicious noise. The crafter of the malicious noise is referred to here in as a malicious application.

[0009] Using malicious noise, viruses and other types of malicious applications are able to direct a client device (e.g., a receiver) to perform actions that a communicatively coupled server (e.g., a sender) did not originally intend. Additionally, the viruses and malicious applications are able to direct a server to perform actions that communicatively coupled client devices did not originally intend. Conventional virus detection algorithms often fail to detect the malicious nature of the noise because these algorithms are configured to detect the presence of the noise's source rather than the noise itself. The noise generation algorithm (e.g., the code of the malicious application) is relatively easily disguised and able

to assume a wide variety of formats. There is accordingly a need to validate communications between servers and client devices in the presence of malicious noise.

SUMMARY

[0010] The present disclosure provides a new and innovative system, methods, and apparatus for validating communications in an open architecture system. A security processor uses variations of soft information to specify how hard information managed by a server is to be displayed on a communicatively coupled client device. The security processor creates a prediction as to how the client device will render the hard information based on the variation of the selected soft information. The security processor then compares information in a response from the client device to the prediction to determine if a malicious application has affected or otherwise altered communications between the server and the client device.

[0011] In an example embodiment, a method for validating communications includes selecting hard information to transmit from a server to a communicatively coupled client device based on a request from the client device and selecting soft information corresponding to the hard information to transmit from the server to the client device. The example method also includes transmitting at least one message including the soft and hard information from the server to the client device and determining a prediction as to how the client device will render the hard information based on the soft information. The example method further includes receiving a response message from the client and responsive to information in the response message not matching the prediction, providing an indication there is a malicious application affecting communications between the server and the client device.

[0012] Additional features and advantages of the disclosed system, methods, and apparatus are described in, and will be apparent from, the following Detailed Description and the Figures.

BRIEF DESCRIPTION OF THE FIGURES

[0013] FIG. 1 is a block diagram of an example network communicating system, according to an example embodiment of the present invention.

[0014] FIG. 2 is a detailed block diagram showing an example of a client device, application server, or database server according to an example embodiment of the present invention.

[0015] FIG. 3 is a diagram of a communication session between a client device and an application server.

[0016] FIG. 4 shows a diagram of backscattered channel information during the communication session between the server and the client device using the communication channel of FIG. 3.

[0017] FIGS. 5 and 6 show diagrams representative of a malicious application affecting the communication session of FIG. 3 between the client device and server.

[0018] FIGS. 7 to 9 show datagrams created by a security processor to identify malicious applications.

[0019] FIG. 10 shows a diagram of a datagram that includes an example of coded soft information selected by a security processor.

[0020] FIG. 11 shows a client device including rendered information and un-rendered information as a variation of soft messaging.

[0021] FIG. 12 shows a security processor validating a transaction between a client device and a server.

[0022] FIG. 13 shows an illustration of two different configurations of a client device that can be accounted for by a security processor to create multiple predictions.

[0023] FIG. 14 shows a diagram of a data structure 1400 of a prediction formed by a security processor based on soft information and secondary information acquired from global and local observable temporal channel information.

[0024] FIGS. 15 and 16 illustrate a flow diagram showing example procedures to validate a communication channel, according to an example embodiment of the present invention.

[0025] FIG. 17 shows diagrams comparing messaging without using an example embodiment of the present invention and messaging using an example embodiment of the present invention.

DETAILED DESCRIPTION

[0026] The present disclosure relates in general to a method, apparatus, and system to validate communications in an open architecture system and, in particular, to predicting responses of client device to identify malicious applications attempting to interfere with communications between servers and the client devices.

[0027] Briefly, in an example embodiment, a system is provided that detects malicious errors in a communication channel between a server and a client device. Normally,

communication errors between a server and a client device are a result of random channel noise. For instance, communications received by server-client endpoints fall outside of a set of prior selected, recognizable, messages or codewords. Channel errors are usually corrected by existing error correction schemes and internet protocols. The end user is typically unaware that a transmission error has occurred and has been corrected.

[0028] Malicious applications typically evade error correcting schemes in two ways: first by altering an original message into an alternative message, and second by creating noise in a segment of a channel where traditional error correction schemes do not operate. In the first way, a malicious application alters an original message into an alternative message that is already in a codeword set of an error correction mechanism. The malicious application may also provide additional messages that are included within the codeword set. As a result, an error correction algorithm is unaware that an error has even taken place and thereby makes no attempt to correct for the error.

[0029] In the second way, a malicious application creates noise in a segment of a channel where traditional error correction schemes do not operate. For example, once a packet successfully traverses the Internet and arrives at a network interface of a receiving device, a bit stream of the packet is processed by an application stack under an assumption that no further transmission noise sources will occur. As a result, the application stack does not anticipate errors to occur in the bit stream after processing and thereby makes no attempt to correct for any errors from this channel noise.

[0030] Malicious applications create targeted malicious noise configured to interfere with communications between a client device and a server. This channel noise is guided by a deliberate purpose of the malicious application to alter, access, or hijack data and/or content that is being communicated across a client-server connection. Oftentimes, the noise alters communications from original and authentic information to substitute authentic-appearing information. The noise is often induced in a segment of the (extended) channel that is poorly defended or entirely undefended by error correction algorithms. As a result, a malicious application is able to use channel noise to direct a server and/or a client device to perform actions that the client device or server did not originally intend.

[0031] In an example, a client device may be connected to an application server configured to facilitate banking transactions. During a transaction, the server requests the client device to provide authentication information (e.g., a username and a password) to access an account. A malicious application detects the connection and inserts malicious noise that causes the client device to display a security question in addition to the username and

password prompts (e.g., client baiting). A user of the client, believing the server provided the security question, enters the answer to the security question with the username and password. The malicious application monitors the response from the client device so as to use malicious noise to remove the answer to the security question before the response reaches the server. The malicious application may then use the newly acquired security question to later illegally access the account associated with the client device to improperly withdrawal funds.

[0032] In this example, the server is unable to detect the presence of the malicious application because the server receives a proper response to the authentication, namely the username and password. The client device also cannot detect the malicious application because the client device believes the server provided the security question. As a result, the malicious application is able to use channel noise to acquire sensitive information from the client device without being detected by the server or the client.

[0033] This client baiting is not the only method used by malicious applications. In other examples, malicious applications may use channel noise to add data transactions between a client device and a server (e.g., add banking transactions). For instance, a client device may specify three bill payment transactions and a malicious application may insert a fourth transaction. In further examples, malicious applications may use channel noise to remove, substitute, or acquire data transmitted between a server and a client, modify data flow between a server and a client, inject graphics or advertisements into webpages, add data fields to forms, or impersonate a client device or a server.

[0034] The example method, apparatus, and system disclosed herein overcome at least some of these issues caused by malicious noise by detecting malicious applications through estimated, predicted, or anticipated responses from a client device. The example method, apparatus, and system disclosed herein detect malicious applications by varying soft information describing how hard information is to be displayed by a client device. During any client-server connection, a server provides hard information and soft information. The hard information includes data, text, and other information that is important for carrying out a transaction with a client. The soft information specifies how the hard information is to be rendered and displayed by the client device.

[0035] A server uses hard and soft messaging to transmit the hard and soft information to a client device. In some instances, the soft and hard information can be combined into messages before transmission. In other examples, the soft and hard information can be transmitted to a client device in separate messages. As used herein, soft messaging refers to the transmission of soft information to a client device in separate or

combined soft/hard messages and hard messaging refers to the transmission of hard information to a client device in separate or combined soft/hard messages.

[0036] The example method, apparatus, and system disclosed herein use variations in soft information to form a best guess (e.g., a prediction or estimation) as to how hard information is displayed by a client device. The example method, apparatus, and system disclosed herein then compare a response from the client device to the best guess. If the information included within the response does not match or is not close enough to the prediction, the example method, apparatus, and system disclosed herein determine that a malicious application is affecting communications between a server and a client or, alternatively, provide an indication that a malicious application is affecting communications. As a result of this detection, the example method, apparatus, and system disclosed herein implement fail safe procedures to reduce the effects of the malicious application.

[0037] The example method, apparatus, and system disclosed herein uses soft information and messaging as a signaling language to detect malicious applications. In other words, the example method, apparatus, and system disclosed herein create an extended set of codewords for use with a user of a client device to validate that a malicious application is not interfering with communications. The created codeword set installs or uses soft messaging techniques including dynamically linked and/or static libraries, frameworks, browser helper objects, protocol filters, etc. The goal of these soft messaging techniques is to perturb the created communication channel such that the soft information cannot be reverse engineered by the malicious application but is known by the client device and the server.

[0038] For instance, FIG. 17 shows diagrams comparing messaging without the example method, apparatus, and system disclosed herein and messaging using the example method, apparatus, and system disclosed herein. Diagram 1700 shows that in the absence of the example method, apparatus, and system disclosed herein, a set of legitimate codewords (denoted by circles) is fixed. Malicious applications know how these codewords are fixed and use malicious noise (denoted by the arrow) to transform a first valid codeword into a second valid codeword. The transformation is undetected by a receiving client device and the sending server.

[0039] In contrast, diagram 1710 shows that the example method, apparatus, and system disclosed herein uses variability in soft information and messaging extends the dimensionality of the codeword set. This variability is unknown by the malicious application. Thus, an error occurs when the malicious noise combines with an intended codeword. As shown in diagram 1710, the resulting altered codeword (denoted by an "X")

does not match the set of anticipated recognized codewords, which enables the malicious noise to be detected. The example method, apparatus, and system disclosed herein are accordingly able to use this soft information and messaging variability to detect malicious noise.

[0040] As used herein, hard messaging and hard information is transactional text and/or data displayed by a client device. The transactional text, data, pictures, and/or images that can be instructional, informational, functional, etc. in nature. The hard information also includes textual options that are selectable by a client. Hard information is accordingly principal information of a transaction or service provided by a server and presented to a client by a client device.

[0041] The hard information includes any type of text and/or data needed by a server to perform a transaction or service on behalf of a client. For instance, hard information of a webpage of an account log-in screen includes text providing instructions to a client as to the nature of the webpage, text for a username field, and text for a password field. After a client has logged into the account, the hard information includes transaction numbers, transaction dates, transaction details, an account balance, and account identifying information. Hard information may be financial (e.g. on-line banking), material (e.g., flow control of raw material in manufacturing processes), or related to data management (e.g., encryption, decryption, addition to or removal from shared storage, copying, deletion, etc.).

[0042] As used herein, soft messaging and soft information is presentation information describing how hard information is to be displayed by a client device. Soft information pertains to the installation and/or system usage of dynamically linked and/or static libraries, frameworks, browser helper objects, protocol filters, javascript, plug-ins, etc. that are used to display hard information without interrupting the communication of the hard portion of the message between a client device and a server. The soft portion of the message includes information based on a server's selection of protocol, formatting, positioning, encoding, presentation, and style of a fully rendered version of hard information to be displayed at the client device endpoint. The soft information can also include preferences (e.g., character sets, language, font size, etc.) of clients as to how hard information is to be displayed. The precise details of the manner or method in which the direct, client device initiated, response information returns to the server is also a soft component of the communication and may be varied or manipulated without detracting from an ability of the server and client device to conduct e-business, e-banking, etc.

[0043] The hard part of the message is constrained, for example, by business utility (e.g., there must be a mechanism for a client device to enter intended account and transaction information and return it to the server) while the soft part of the message has fewer constraints. For example, the order in which a client device enters an account number and a transaction amount usually is not important to the overall transaction. To achieve the business purpose a server only has to receive both pieces of information.

[0044] In the client baiting example described above, the example method, apparatus, and system disclosed herein cause the server to transmit to the client device in one or more soft messages code that causes the client device to return coordinates of a mouse click of a 'submit' button. These soft messages are included with the other soft messages describing how the authentication information is to be displayed by the client. The server also determines a prediction as to what the coordinates should be based on knowing how the particular client device will render and display the information.

[0045] When the malicious application uses malicious noise to insert the security question, the malicious application has to move the 'submit' button lower on a webpage. Otherwise, the security question would appear out of place on the webpage in relation to the username and password fields. When a user of the client device uses a mouse to select the 'submit' button, the client device transmits the coordinates of the mouse click to the server. The server compares the received coordinates with the coordinates of the prediction and determines that the difference is greater than a standard deviation threshold, which indicates the presence of a malicious application. In response to detecting the malicious application, the server can initiate fail safe procedures to remedy the situation including, for example, requiring the client device to create new authentication information or restricting access to the account associated with the client device.

[0046] As can be appreciated from this example, the example method, apparatus, and system disclosed herein provide server-client communication channel validation. By knowing how a client device is to display information, the example method, apparatus, and system disclosed herein enable a server to identify remotely located malicious applications that mask their activities in hard to detect channel noise. As a result, servers are able to safeguard client data and transactions from some of the hardest to detect forms of malicious third party methods to acquire information and credentials. This allows service providers that use the example method, apparatus, and system disclosed herein to provide security assurances to customers and other users of their systems.

[0047] Throughout the disclosure, reference is made to malicious applications (e.g., malware), which can include any computer virus, counterfeit hardware component, unauthorized third party access, computer worm, Trojan horse, rootkit, spyware, adware, or any other malicious or unwanted software that interferes with communications between client devices and servers. Malicious applications can interfere with communications of a live session between a server and a client device by, for example, acquiring credentials from a client device or server, using a client device to instruct the server to move resources (e.g., money) to a location associated with the malicious application, injecting information into a form, injecting information into a webpage, capturing data displayed to a client, manipulating data flow between a client device and a server, or impersonating a client device using stolen credentials to acquire client device resources.

[0048] Additionally, throughout the disclosure, reference is made to client devices, which can include any cellphone, smartphone, personal digital assistant (“PDA”), mobile device, tablet computer, computer, laptop, server, processor, console, gaming system, multimedia receiver, or any other computing device. While this disclosure refers to connection between a single client device and a server, the example method, apparatus, and system disclosed herein can be applied to multiple client devices connected to one or more servers.

[0049] Examples in this disclosure describe client devices and servers performing banking transactions. However, the example method, apparatus, and system disclosed herein can be applied to any type of transaction or controlled usage of resources between a server and a client device including, but not limited to, online purchases of goods or services, point of sale purchases of goods or services (e.g., using Near Field Communication), medical applications (e.g., intravenous medication as dispensed by an infusion pump under the control of a computer at a nurses station or medication as delivered to a home address specified in a webpage), manufacturing processes (e.g., remote manufacturing monitoring and control), infrastructure components (e.g., monitoring and control of the flow of electricity, oil, or flow of information in data networks), transmission of information with a social network, or transmission of sensitive and confidential information.

[0050] The present system may be readily realized in a network communications system. A high level block diagram of an example network communications system 100 is illustrated in FIG. 1. The illustrated system 100 includes one or more client devices 102, one or more application servers 104, and one or more database servers 106 connected to one or more databases 108. Each of these devices may communicate with each other via a

connection to one or more communication channels in a network 110. The network 110 can include, for example the Internet or some other data network, including, but not limited to, any suitable wide area network or local area network. It should be appreciated that any of the devices described herein may be directly connected to each other and/or connected through the network 110. The network 110 may also support wireless communication with wireless client devices 102.

[0051] The client devices 102 access data, services, media content, and any other type of information located on the servers 104 and 106. The client devices 102 may include any type of operating system and perform any function capable of being performed by a processor. For instance, the client devices 102 may access, read, and/or write information corresponding to services hosted by the servers 104 and 106.

[0052] Typically, servers 104 and 106 process one or more of a plurality of files, programs, data structures, databases, and/or web pages in one or more memories for use by the client devices 102, and/or other servers 104 and 106. The application servers 104 may provide services accessible to the client devices 102 while the database servers 106 provide a framework for the client devices 102 to access data stored in the database 108. The servers 104 and 106 may be configured according to their particular operating system, applications, memory, hardware, etc., and may provide various options for managing the execution of the programs and applications, as well as various administrative tasks. A server 104, 106 may interact via one or more networks with one or more other servers 104 and 106, which may be operated independently.

[0053] The example servers 104 and 106 provide data and services to the client devices 102. The servers 104 and 106 may be managed by one or more service providers, which control the information and types of services offered. These services providers also determine qualifications as to which client devices 102 are authorized to access the servers 104 and 106. The servers 104 and 106 can provide, for example, banking services, online retail services, social media content, multimedia services, government services, educational services, etc.

[0054] Additionally, the servers 104 and 106 may provide control to processes within a facility, such as a process control system. In these instances, the servers 104 and 106 provide the client devices 102 access to read, write, or subscribe to data and information associated with specific processes. For example, the application servers 104 may provide information and control to the client devices 102 for an oil refinery or a manufacturing plant.

In this example, a user of the client device 102 can access an application server 104 to view statuses of equipment within the plant or to set controls for the equipment within the plant.

[0055] While the servers 104 and 106 are shown as individual entities, each server 104 and 106 may be partitioned or distributed within a network. For instance, each server 104 and 106 may be implemented within a cloud computing network with different processes and data stored at different servers or processors. Additionally, multiple servers or processors located at different geographic locations may be grouped together as server 104 and 106. In this instance, network routers determine which client device 102 connects to which processor within the application server 104.

[0056] In the illustrated example of FIG. 1, each of the servers 104 and 106 includes a security processor 112. The security processor 112 monitors communications between the client devices 102 and the respective servers 104 and 106 for suspicious activity. The monitoring may include detecting errors in a communication channel between a client device 102 and a server 104 using hard and soft messages, as described herein.

[0057] In some embodiments, the security processor 112 may be configured to only detect channel errors that are of strategic importance. This is because malicious applications generally only target communications that convey high value information (e.g., banking information). As a result, using the security processor 112 for important communications helps reduce processing so that the security processor 112 does not validate communications that are relatively insignificant (e.g., browsing a webpage). These important communications can include authentication information, refinements to types of requested services, or details on desired allocation of resources under a client's control. These resources may be financial (e.g., on-line banking), material (e.g., flow control of raw material in manufacturing processes) or related to data management (e.g., encryption, decryption, addition to or removal from shared storage, copying, deletion, etc.).

[0058] In an example embodiment, a client device 102 requests to access data or servers hosted by a server 104. In response, the server 104 determines hard information that corresponds to the request and identifies soft information compatible with the hard information. In some instances, the server 104 may use device characteristics or information of the client device 102 to select the soft messaging. Upon selecting the soft and hard messages, the security processor 112 selects how the messages are combined into transmission packets and instructs the server 104 to transmit the packets to the client device 102. To make the packets undecipherable by malicious applications, the security processor

112 may combine hard and soft information, rearrange the order of information transmission, or mix different layers of information.

[0059] The unperturbed location of any input boxes or buttons selected by the security processor 112 for soft messaging may vary, subtly, from session to session, without being observable by a client device 102 or a malicious application. For example, the absolute and relative positioning of page elements may be obscured by the incorporation of operating system, browser, and bugz and further obscured by seemingly routine use of byte code and javascript. The security processor 112 may also use redundant measures for determining rendered page geometry and activity so that information returned from the client device 102 may be further verified. For instance, benign “pop-up windows” featuring yes/no button messages such as: “would you have time to take our brief customer survey?” may be made to appear or not appear depending on actual cursor or mouse locations when a ‘submit’ button is pressed at the client device 102. Additionally, the security processor 112 may use generic geometrical and content related soft-variations (absolute and relative locations of input boxes and buttons, the appearance or lack of appearance of benign “pop-up” boxes, buttons, advertisements or images) to validate communications with a client device 102. In other words, the security processor 112 may use soft information provided by client devices 102 to also validate a communication channel.

[0060] After selecting which soft and hard information to send to the client device 102, the security processor 112 makes a prediction, in this example, as to a location of a ‘Submit’ icon on a fully rendered webpage displayed on client device 102. This icon is part of a banking website provided by application server 104. The security processor 112 may also use backscattered information received from routing components in the network 110 to form the prediction. This backscattered information provides, for example, how the soft and hard information in the transmitted message(s) are processed, routed, and rendered.

[0061] The security processor 112 then monitors a response by the client device 102 to identify coordinates of a mouse click of the ‘Submit’ icon. The security processor 112 determines that a malicious application is affecting communications if the prediction does not match the reported coordinates of the mouse click on the icon. In response to detecting a malicious application, the security processor 112 attempts to prevent the malicious application from further affecting communications with the affected client devices 102. In some embodiments, the security processor instructs the servers 104 and 106 to alter normal operation and enter into a safe operations mode. In other embodiments, the security processor 112 restricts activities of the affected client devices 102 or requests the client

devices 102 to re-authenticate or establish a more secure connection. The security processor 112 may also store a record of the incident for processing and analysis. In further embodiments, the security processor 112 may transmit an alert and/or an alarm to the affected client devices 102, personnel associated with the servers 104 and 106, and/or operators of the security processor 112.

[0062] While each server 104 and 106 is shown as including a security processor 112, in other embodiments the security processor 112 may be remotely located from the servers 104 and 106 (e.g., the security processor 112 may be cloud-based). In these embodiments, the security processor 112 is communicatively coupled to the servers 104 and 106 and remotely monitors for suspicious activity of malicious applications. For instance, the security processor 112 may provide soft information to the servers 104 and 106. The security processor 112 may also receive client device response messages from the servers 104 and 106. In instances when the security processor 112 detects a malicious application, the security processor 112 remotely instructs the servers 104 and 106 how to remedy the situation.

[0063] A detailed block diagram of electrical systems of an example computing device (e.g., a client device 102, an application server 104, or a database server 106) is illustrated in FIG. 2. In this example, the computing device 102, 104, 106 includes a main unit 202 which preferably includes one or more processors 204 communicatively coupled by an address/data bus 206 to one or more memory devices 208, other computer circuitry 210, and one or more interface circuits 212. The processor 204 may be any suitable processor, such as a microprocessor from the INTEL PENTIUM® or CORE™ family of microprocessors. The memory 208 preferably includes volatile memory and non-volatile memory. Preferably, the memory 208 stores a software program that interacts with the other devices in the system 100, as described below. This program may be executed by the processor 204 in any suitable manner. In an example embodiment, memory 208 may be part of a “cloud” such that cloud computing may be utilized by computing devices 102, 104, 106. The memory 208 may also store digital data indicative of documents, files, programs, web pages, etc. retrieved from computing device 102, 104, 106 and/or loaded via an input device 214.

[0064] The example memory devices 208 store software instructions 223, webpages 224, user interface features, permissions, protocols, configurations, and/or preference information 226. The memory devices 208 also may store network or system interface features, permissions, protocols, configuration, and/or preference information 228 for use by

the computing devices 102, 104, 106. It will be appreciated that many other data fields and records may be stored in the memory device 208 to facilitate implementation of the methods and apparatus disclosed herein. In addition, it will be appreciated that any type of suitable data structure (e.g., a flat file data structure, a relational database, a tree data structure, etc.) may be used to facilitate implementation of the methods and apparatus disclosed herein.

[0065] The interface circuit 212 may be implemented using any suitable interface standard, such as an Ethernet interface and/or a Universal Serial Bus (USB) interface. One or more input devices 214 may be connected to the interface circuit 212 for entering data and commands into the main unit 202. For example, the input device 214 may be a keyboard, mouse, touch screen, track pad, track ball, isopoint, image sensor, character recognition, barcode scanner, microphone, and/or a speech or voice recognition system.

[0066] One or more displays, printers, speakers, and/or other output devices 216 may also be connected to the main unit 202 via the interface circuit 212. The display may be a cathode ray tube (CRTs), a liquid crystal display (LCD), or any other type of display. The display generates visual displays generated during operation of the computing device 102, 104, 106. For example, the display may provide a user interface and may display one or more webpages received from a computing device 102, 104, 106. A user interface may include prompts for human input from a user of a client device device 102 including links, buttons, tabs, checkboxes, thumbnails, text fields, drop down boxes, etc., and may provide various outputs in response to the user inputs, such as text, still images, videos, audio, and animations.

[0067] One or more storage devices 218 may also be connected to the main unit 202 via the interface circuit 212. For example, a hard drive, CD drive, DVD drive, and/or other storage devices may be connected to the main unit 202. The storage devices 218 may store any type of data, such as pricing data, transaction data, operations data, inventory data, commission data, manufacturing data, marketing data, distribution data, consumer data, mapping data, image data, video data, audio data, tagging data, historical access or usage data, statistical data, security data, etc., which may be used by the computing device 102, 104, 106.

[0068] The computing device 102, 104, 106 may also exchange data with other network devices 220 via a connection to the network 110 or a wireless transceiver 222 connected to the network 110. Network devices 220 may include one or more servers (e.g., the application servers 104 or the database servers 106), which may be used to store certain types of data, and particularly large volumes of data which may be stored in one or more data

repository. A server may include any kind of data including databases, programs, files, libraries, pricing data, transaction data, operations data, inventory data, commission data, manufacturing data, marketing data, distribution data, consumer data, mapping data, configuration data, index or tagging data, historical access or usage data, statistical data, security data, etc. A server may store and operate various applications relating to receiving, transmitting, processing, and storing the large volumes of data. It should be appreciated that various configurations of one or more servers may be used to support and maintain the system 100. For example, servers may be operated by various different entities, including sellers, retailers, manufacturers, distributors, service providers, marketers, information services, etc. Also, certain data may be stored in a client device device 102 which is also stored on a server, either temporarily or permanently, for example in memory 208 or storage device 218. The network connection may be any type of network connection, such as an Ethernet connection, digital subscriber line (DSL), telephone line, coaxial cable, wireless connection, etc.

[0069] Access to a computing device 102, 104, 106 can be controlled by appropriate security software or security measures. An individual users' access can be defined by the computing device 102, 104, 106 and limited to certain data and/or actions. Accordingly, users of the system 100 may be required to register with one or more computing devices 102, 104, 106.

The Client-Server Communication Channel

[0070] FIG. 3 shows a diagram of a communication session 300 between a client device 102 and an application server 104. The communication session 300 occurs over a communication channel 302, which is included in the network 110 of FIG. 1. The communication channel 302 includes hardware and software components that convey, relay, shape and forward information between the server 104 and the client device 102. The hardware components includes network node devices such as routers, mobile switching center components, base switching center components, data storage, caches, device proxies and firewalls. The hardware components can also include client device specific endpoints, computer architecture, processor types, mobile device chipsets, SIM cards and memory. The software components of the channel include network or endpoint device platforms, instruction sets, operating systems, operating system versions, application programming interfaces ("api"), and libraries. The software components can also include client device endpoint software, user interfaces, browser types, browser versions, cascading style sheets, scripts, document object models, javacode, byte script, etc.

[0071] In the communication channel 302, information transmitted by the server 104 (e.g., soft/hard information included within soft/hard messages) is acted upon, processed, forwarded, and rendered by the various intervening hardware and software channel components. The processing is performed by hardware and software components residing on both network and client device endpoints. The client device 102 is the ultimate recipient of the fully realized, completely processed version of the information transmitted by the server 104. The client device 102 is stimulated by the received (processed) information into prompting a user for decision(s) and/or performing one or more actions. Once a user inputs a decision, the client device 102 communicates a response message to the server 104 through the channel 302.

[0072] While FIG. 3 shows one communication channel 302, other communication channels can include different components and corresponding behavioral characteristics that vary from one server-client device connection to another. The behavioral characteristics identify ways in which information is acted upon, processed, forwarded and rendered by the hardware and software components of the channel 302. The security processor 112 uses these behavioral characteristics to help form a prediction of a response from the client device 102.

[0073] Once a server-client device connection is established across a channel 302 and the primary, intended function of that communication is initiated (e.g., the type of transaction that is to occur across the channel 302), secondary characteristics and observables are generated in the channel 302 as a consequence. There are two types of secondary characteristics and observables: “global” (involving many or all channel components) and “local” (involving a single, pair, or triple of channel components).

[0074] The “global” channel’s temporal secondary characteristics are applied across many or all hardware/software components and layers in, for example, the network 110 and include: i) number and size of discrete transmissions, ii) density of discrete transmissions, iii) frequency and other spectral content (e.g., content obtained by discrete Fourier transform, wavelet transform, etc. of an observed time series), and iv) geo-spatial density. These characteristics are derived from observables (e.g., from observation of information flow between client device 102 and server 104) that include, for example, i) delivery times, ii) delivery rates, iii) transmission requests (as reports on errors or inefficiencies), and iv) sequencing or permutations in an order of information processing events. These observables are dependent on a number of factors including, for example, hardware type, software type,

and current state (e.g., traffic load, internal queue lengths, etc.) of components that comprise the channel 302.

[0075] “Local” observables may also be generated on a per client device basis or per layer basis in the channel 302 of FIG. 3 by server 104 and/or client device 102 initiated stimuli. The variations between client devices or layers are a result of a client’s or layer’s internal, device specific, information processing prioritization rulesets/protocols, inter-component signaling rulesets, and/or protocols that use hardware or software-based signaling components. The local observables may indicate, for example, a browser type used by the client device 102, an operating system of the client device 102, or device coding schemes used by the client device 102.

[0076] In the example embodiment of FIG. 3, the security processor 112 structures the hard and soft messaging output by the server 104 so that the secondary characteristics and observables function as a secondary means of communication between the client device 102 and server 104. At the same time, the security processor 112 structures the hard and soft messaging output by the server 104 in a manner consistent with the original purpose of the connection with the client device 102. As a result, the secondary means of communication between the server 104 and the client device 104 over the channel 302 is configured to not interfere with the primary, intended function of the server-client device interaction. As a result, the security processor 112 uses the channel 302 to vary soft information without changing the nature of the intended transaction between the server 104 and the client device 102.

[0077] In FIG. 3, the channel 302 is constructed for universal use (e.g., an open architecture channel). That is, the component and collections of component technologies of the channel 302 are designed to enable a rich variety of server types, client device types, and purposeful communications between the different server and client device types. This enables the security processor 112 to use a variety of different soft messaging methods to achieve the original, intended purpose of the server-client transaction. However, each soft messaging method sets into motion a different set of (global and local) channel characteristic signals and observables. The security processor 112 is accordingly able to establish a secondary communication language between the server 104 and the client device 102 across the channel 302 using the association between variations in soft messaging methods (global, local) and corresponding channel characteristic responses.

[0078] The communication session 300 of FIG. 3 also includes malicious applications 304, which are configured to interfere with client-server communications while allowing the

primary, intended function of the server-client device interaction to occur. However in accomplishing and creating this perturbation of the primary, intended communication between the server 104 and the client device 102, the malicious applications 304 effectively become an “additional component” of the channel 302, thereby unknowingly affecting the secondary communications. As shown in FIG. 3, the malicious applications 304 can insert information into the channel 302 and/or extract information from the channel 302 using engineered channel noise. The example security processor 112 detects these malicious applications 304 by monitoring how generated malicious channel noise impacts the consistently crafted client-server secondary communications.

[0079] FIG. 4 shows a diagram of backscattered channel information 402 during the communication session 300 between the server 104 and the client device 102 using the communication channel 302 of FIG. 3. From the point of view of the server 104 (or a trusted proxy), a complete communication with the client device 102 includes two distinct segments: information sent to the client device 102 and information received from the client device 102 in response to the information sent. Information 402 regarding the progress of channel components in processing, realizing and rendering information and inter-device signaling events, scatter back to the server 104. If the server 104, via the security processor 112, subtly varies the content that it sends to the client device 102 through soft messaging, the effects of the changes will be detectable in the echoed information returning back to the server 104 from the various components and processing layers of the channel 302.

[0080] The *a priori* knowledge of the information transmitted by the server 104 (the information and stimuli actually sent into the channel 302 to the client device 102) together with the global and local backscatter information 402 from the components and layers of the channel 302, permit the server 104 (or a trusted proxy) to form a prediction as to the condition of the final, post-processing, fully rendered version of the information displayed by the client device 102. Additionally, direct, client device initiated, response messages to the server 104 (e.g., mouse clicks or user supplied account information) constitute a means for the security processor 112 to determine a prediction as to the fully rendered version of the information displayed by the client device 102. The information in the response from the client device 102 can be entered by a user using a mouse, keyboard, touchscreen, an infrared ID tag scanner, etc. For example, information of a returned mouse click informs the security processor 112 that a selectable box was 1) rendered, 2) selected, and 3) the click was preformed at (x,y) pixel coordinates.

[0081] The security processor 112 determines discrepancies between the prediction and the direct, client device 102 initiated responses of the fully rendered information to detect and identify errors (e.g., malicious applications 304) in the channel 302. The detection and identification of channel error causes the security processor 112 to alter normal operations of the server 104. In some embodiments, the security processor 112 may cause the server 104 to enter a safe operations mode, restrict authorized client device activities, and/or generate an alert and/or alarm.

The Use of Soft Messaging for Channel Verification

[0082] As discussed above, the security processor 112 can use different types and variations of soft messaging and information to help identify malicious applications. This variation helps prevent malicious applications from reverse engineering the soft messaging and circumventing the approaches described herein. As described below, the variation can include changes to font size, changes to web page arrangement of hard information and graphics, addition of characters to user inputs, changes to function definitions, requests for user prompts through banners and pop-up windows, or implementations of bugs. The variation can also include changing an order in which hard and soft information is sent from a server 104 or a client device 102.

[0083] The order in which information arrives at a server 104 or client device 102 is not relevant for business purposes. The inclusion of additional information, for example the pixel location of a mouse click, cursor, or scroll bar (e.g., soft information) in addition to account information (e.g., hard information) does not affect the business purpose. The method of encoding information, and within reasonable bounds, the amount of time information spent in transmission of channel 302 have a generally neutral impact on business purposes. “Soft” choices consistent with the “hard” business purpose exist at many layers of the channel 302 ranging from the choice(s) of physical method(s) used, transmission encoding method(s) used on the physical layer(s), to aesthetic details of information presentation and user interactions with a presented webpage. The choice of soft messaging by the server 104 (or its trusted proxy) corresponding to given hard information is a many-to-one mapping. In a similar way, the local, specialized function and contribution of each network and client device specific hardware and software channel component is decomposable into hard and soft elements consistent with achieving the overall, global intent of the interaction of the server 104 with the client device 102.

[0084] The security processor 112 accordingly maintains hard functionality of the server-client device connection (e.g., the session 300) while varying the soft information.

Soft information variations are recorded a priori by the security processor 112 or the server 104 (or its trusted proxy) in a data structure to create a large set of composite (hard and soft) messages to be transmitted together. In other embodiments, the server 104 may transmit the hard messages separate from the soft messages. The soft variations are constrained by the fact the final presentation at the client device 102 must be intelligible, not garbled. Further, the soft variations must be of sufficient complexity that the malicious applications 304 are faced with a time consuming reverse engineering problem in deciphering the accumulated impact of the soft message changes throughout the channel 302.

The Use of Bugz in Soft Messaging for Channel Verification

[0085] As mentioned above, the security processor 112 may use implementations of bugz in soft information variation. Bugz are anomalous, device, software, protocol and/or physical communication medium specific interpretations of input instructions that produce consistent although unexpected output. Bugz are inherent in many components of the channel 302 and are generally undetectable by malicious applications 304 without significant processing and analysis. The use of bugz helps enhance the complexity of soft messaging by enabling the security processor 112 to craft soft information so that the soft degrees of freedom within and between hardware and software based components of the channel 302 are combined in a multiplicative fashion. While four examples of bugz are described below, the security processor 112 can implement any type of bugz in soft messaging.

[0086] One type of bugz is based on different operating systems of client devices 102 processing the same incoming packet streams differently. As a result of this bugz, the security processor 112 can create soft messaging packet streams indented to induce certain known behaviors in an operating system to display hard information. Another type of bugz is based on different operating systems of client devices 102 interpreting the same portion of Extensible Markup Language (“xml”) code differently. Prior to initializing its service to a client device 102, a server 104 or security processor 112 selects from a variety of ways that a portion of xml code may be written and select from a variety of ways to order, time delay, and geographically position the way the packets containing that code are transmitted into the channel 302.

[0087] Yet another type of bugz is based on HyperText Markup Language (“html”) code and cascading style sheet instructions that can be written and combined in contrasting and confusing fashion by a server 104 or the security processor 112. The server 104 can also use different layers of the style sheet in opposition of each other. For example, the security processor 112 could instruct a server 104 to randomize which portions of a webpage are sent

in style sheet instructions at sequential times. As a result, a malicious application 304 is unable to easily determine which style sheet instruction corresponds to which portion of the webpage.

[0088] A further type of bugz is based on code libraries that are internally re-arranged by the security processor 112 so that functions that use the code libraries on client devices 102 are contrasted with expected performance in accord with the usage conventions of the standard library. For example, the security processor 112 can use this type of bugz to swap the definitions of the “add” and “multiply” functions. As a result of this swap, the client device 102 performs the intended function while a malicious application 304 incorrectly determines that a different function is being performed. As a result, the security processor 112 can determine if a malicious application 304 attempts to change a result of the function or transaction.

[0089] Often the ultimate resolution of the purposefully mis-engineered “spaghetti” code applied by the security processor 112 in soft messaging depends on a browser type and version at the client device 102. Java script and bytecode, for example, may be similarly obfuscated by the security processor 112 without negatively detracting from run time performance or the ability of the server 104 and client device 102 to conduct business. These effects of the examples described above may be enhanced by incorporating operating system and browser bugz into the instructions. The result of this incorporation is a soft formatting and presentation style at a client device endpoint that makes it difficult for malicious applications 304 to predict and/or automatically interpret the soft information. This makes the soft information difficult for the malicious applications 304 to alter, replace, or counterfeit in real time. Although this encoding is difficult to interpret in real time, it may be easily tested experimentally, a priori by a server 104 (or its trusted proxy). It is this a priori knowledge of the unperturbed and fully implemented rendering of the instruction set at the client device 102 that forms the basis of the prediction determination made by the security processor 112 of the formatting at the client device endpoint. The example security processor 112 creates the variation among the soft messages to increase the differences between the prediction and direct versions of the fully rendered information displayed by the client device 102.

A Comparative Example of Channel Verification with and without the Security Processor

[0090] FIGS. 5 and 6 show diagrams representative of a malicious application 304 affecting the communication session 300 between the client device 102 and server 104. In particular, FIG. 5 shows the affects of the malicious application 304 when the server 104 does not include a security processor 112 and FIG. 6 shows the affects of the malicious

application 304 when the server 104 includes the security processor 112. It should be noted that FIGS. 5 and 6 are only one example of communications between a client device 102 and server 104. Other examples can include additional affects by malicious applications 304 and/or different types of transactions performed between the server 104 and client device 102.

[0091] In FIG. 5, the server 104 intends to communicate a deliberate, per-determined datagram 402 to client device 102. Here, the datagram 402 is a webpage that prompts a user to provide a username and password. The pre-determined datagram 402 is represented as a binary form for transmission purposes, shown in FIG. 5 as the number “0” in data transmission 404. The final, fully rendered, intended, client device intelligible and/or useable form of the data transmission 404 is known to the server 102 (or its proxy) at and/or before the time of the data transmission to the client device 102.

[0092] The pre-determined, intended data transmission 404 progresses through and/or is processed by the various hardware and/or software based components, layers, and protocols of channel 302. The sequence of “0’s” represents the original intent of the server 104 and is represented in FIG. 5 as a sequence of “0’s” progressing through a sequence of rectangles in the direction of the dashed, horizontal arrows 406. The upper arrow represents the sequence of processing events experienced by the “hard” portion of the data and the lower arrow represents the sequence of processing events experienced by the “soft” data. The soft and hard data transmission paths may or may not be the same and may or may not entail identical processing events.

[0093] As transmitted data 404 progresses through and/or is processed by the channel 302 with the original intent of the server 104 intact, secondary information 408 generated by the routing and processing of the data 404 is scattered back through the channel 302 to the server 102. The secondary information 408 can include, for example, an operating system of the client device 102, a browser type used by the client device 102, a cascading style sheet type used to display the soft/hard information, java script information, byte code data, etc. In other instances, the secondary information 408 may be reported by the client device 102 as device information after initiating the communication session 300 with the server 104. The secondary information 408 is generated, for example, from Transmission Control Protocol/Internet Protocol (“TCP/IP”) negotiation, Hypertext Transfer Protocol (“HTTP”) requests and conformations, and/or rendering information. In other examples, the secondary information 408 can be generated through other channel 302 backscattering routing and/or processing.

[0094] During transmission of the data 404 to the client device 102, the malicious application 304 creates channel noise 410, which alters the data 404. The channel noise 410 causes an intelligent modification of the data 404 to be realized at the client device 102 instead of the original pre-determined datagram 402. This alteration is represented in FIG. 5 as the number “1” and may incorporate hard and/or soft information.

[0095] The client device 102 receives the final, fully rendered, client device intelligible form of the data as altered by the malicious application 104 and displays this data as datagram 412. Here, the channel noise 410 adds a security question to the webpage and moves the location of a ‘submit’ button to accommodate the security question. As a result, of this channel noise 410, the server 104 believes the client device 102 is viewing datagram 402 when in fact the client device 102 is viewing altered datagram 412. Further, a user of the client device 412 has no reason to be suspicious of the datagram 412 because the maliciously inserted security question appears to coincide with the remainder of the datagram 412.

[0096] When the client device 102 returns a response message to the server 104, the malicious application 304 detects the response and uses channel noise 410 to remove the answer to the security question. This is represented by transition of the data 404 from “1” to “0” before the data reaches the server 104. As a result, the server 104 receives a response from the client device 102 that only includes the username and password. The server 104 never received an indication that the client device 102 provided a response to a security question, and, accordingly, never detects the presence of the malicious application 304. The malicious application 304 remains hidden to carry out further stealthy compromises of account security.

[0097] FIG. 6 shows how security processor 112 can validate communications between the server 104 and the client device 102 during the same communication session 300 described in conjunction with FIG. 5. Similar to FIG. 5, the server 104 in FIG. 6 is to transmit a request for a username and password to access an account. However, unlike in FIG. 5, the security processor 112 in FIG. 6 specifically creates the soft content of a deliberate, pre-determined datagram 502 before transmission to the client device 102.

[0098] FIG. 6 shows soft information 504 and hard information 506 transmitted by the server 104. The security processor 112 varies the soft data 504 from one client-server connection to the next to prevent the client device 102 or the malicious application 304 from knowing the components of the soft information 504 beforehand. The soft information 504 is however fully understood by the server 104 (or its trusted proxy) by the time of transmission to the client device 102. The server 104 stores the soft information 504 to a data structure to

help form a prediction as to a response from the client device 102. The server 104 initiated soft and hard information 504, 506 is shown as “0’s” in the blocks.

[0099] During the communication session 300, the propagations of the soft and hard information 504, 506 through channel 302 cause secondary information 508 to be generated. The secondary information 508 is scattered back to the server 104 and the security processor 112. The security processor 112 uses the secondary information 508 in conjunction with the soft information 504 to form a datagram 510 of the prediction. FIG. 7 shows an enlarged image of the datagram 510 including the request for the username and password. The server 104 uses the datagram 510 to predict how the client device 102 will process, render, and display datagram 502. In other embodiments, the security processor 112 stores the secondary information 508 in conjunction with the soft information 504 in a data structure rather than rendering datagram 510.

[00100] Similar to FIG. 5, the malicious application 304 uses channel noise 512 to alter the soft and/or hard data 504, 506, which is shown in FIG. 6 as the number “1.” As before, the alternation includes the addition of a security question and the movement of the ‘submit button.’ The client device 102 then receives, processes, renders, and displays the altered data.

[00101] A rendered datagram 514, as displayed by the client device 102, is displayed in FIG. 8. This datagram 514 shows a security question prompt below the prompts for the username and the password. In addition, the ‘submit’ button and corresponding text have been lowered in the datagram 514 to make room for the security question. As a result, the security question appears to be genuine to a user of the client device 102.

[00102] After displaying the datagram 514, the client device 102 transmits a response, which also includes hard and soft information. Similar to FIG. 5, the malicious application 304 uses channel noise 510 to remove the response to the security question, which is shown in FIG. 6 in the transition of the hard information 506 from “1” to “0.” However, while the malicious application 304 removed the hard information 506, the malicious application 304 is not concerned with the mouse click information, and accordingly does not alter the soft information 504.

[00103] In FIG. 6, the server 104 and the security processor 112 receive the response from the client device 102, including the hard and soft information 504, 506. The security processor 112 compares the soft information 504 to the prediction and is able to determine that the communication session 300 has been compromised. In other words, the security processor 112 detects the malicious application 304 by determining that the

coordinates of the mouse click on the 'submit' button do not match the coordinates of the 'submit' button made during the prediction.

[00104] FIG. 9 shows a diagram of a comparison datagram 516 representative of the comparison made by the security processor 112 to determine if a malicious application is affecting communications between the server 104 and the client device 102. The comparison datagram 516 includes the prediction datagram 510 formed by the security processor 112 and a construction of the datagram 514 based on the soft and hard information received from the client device 102. For visually effect, the prediction datagram 510 is superimposed upon the datagram 514 from the client device 102.

[00105] As shown in FIG. 9, the geometry of the datagram 514 is altered, in particular the position of the 'submit' button as a result of the space needed to reformat the page and accommodate the additional bogus security question. In addition to the location of the data fields, the datagram 514 includes soft information 504 such as a position of a mouse click associated with the 'submit' button. In this example, the server 104 requests that the client device 102 report the mouse click as soft information, for example, by relying on a "hidden in the clear" communication protocol. In some examples, the server 104 or security processor 112 may embed the authentication form in a "trendy" image so that the relative coordinates of the mouse clicks are returned as a matter of routine and not detected by the malicious application.

[00106] In this authentication page example, by comparing the prediction position of the 'submit' button with the directly reported position, the security processor 112 detects whether an error has occurred during communication session 300. Here, the security processor 112 detects that the datagram 514 does not align with the datagram 510, and accordingly determines that the malicious application 304 is affecting communications.

[00107] In some embodiments, the security processor 112 may determine an allowable deviation or threshold for datagram 510. Thus, as long as, for example, the 'submit' button is located within the allowable deviation, the security processor 112 determines that communications are not being affected by malicious applications. The security processor 112 may determine what an allowable deviation is for the datagram 510 based on, for example, secondary information 508, characteristics of the client device 102, or history information of how the datagram 510 has been displayed by other client devices.

Examples of Channel Verification using Different Types of Soft Messaging

[00108] As disclosed, the security processor 112 uses different types and variations of soft information and soft messaging to validate communication channels

between servers 104 and client devices 102. The types of soft information and messaging can include changes to font size, changes to web page arrangement of hard information and graphics, addition of characters to user inputs, changes to function definitions, requests for user prompts through banners and pop-up windows, or implementations of bugz. The following sections describe how the security processor 112 uses different types of soft information and messaging.

Soft Messaging using Text Size and Font Variations

[00109] FIG. 10 shows a diagram of a datagram 1000 that includes a code section 1002 and a result section 1004. The datagram 1000 illustrates how soft information can be selected or created by the security processor 112 in code section 1002. The datagram 1000 also shows how the soft information would be displayed on a client device 102 in the result section 1004.

[00110] FIG. 10 shows that character sets, font types and point sizes may be varied by the security processor 112 from session to session. These variations are in addition to the geometrical and content related soft-variations described in the previous comparative example. In the code section 1002, keyboard and mouse functionality may be made functions of a number of characters typed or x,y coordinates of text boxes. These modifications may be subtle and may also be made session dependent. The security processor 112 may invoke changes using any seemingly contrasting combination of coding instructions via html, xml, CSS javascript, byte code, etc. The security processor 112 may also invoke changes by altering or restricting elements available for coding instructions to draw from, for example, available character sets.

[00111] For example, in the datagram 1000, the security processor 112 is subject to a ruleset based on the hard information that is required to be transmitted (e.g., the prompt for a username and password). Here, the security processor 112 selects soft information or message variation such that for the fully processed and rendered information presented to the client device 102 is structured so that the username transaction field is to be rendered by a client device 102 in a font size of 12, the first password field is to be rendered in a font size of 13, and the second password field is to be rendered in a font size of 14. In other examples, the security processor 112 may also vary a font type, font color, font weight, or any other text variation allowable for the corresponding hard information.

[00112] The variation among the font sizes is used by the security processor 112 to form a prediction. For instance, the name provided by the client device 102 is to be in 12 point font while the first password is to be in 13 point font. If a malicious application uses

channel noise to alter the username or password responses or add a second transaction, the security processor 112 is able to detect the modification by the malicious application if the returned font size does not match the prediction. If the malicious application is more sophisticated and processes the soft information returned from the client device 102 to determine the font size, the extra time spent processing the information provides an indication to the security processor 112 that a malicious application is affecting communications. As a result, the soft messaging makes it relatively difficult for a malicious application to go undetected by the security processor 112.

Soft Messaging using Programmed Keystrokes

[00113] In another embodiment, the code section 1002 may include code that instructs a client device 102 to programmatically generate keystrokes based on keystrokes provided by a user. The security processor 112 uses the algorithm for the programmatically generated keystrokes to form a prediction. The security processor 112 transmits the algorithm for the programmatically generated keystrokes through xml code, java code, etc. The security processor 112 may also use the programmatically generated keystrokes in Document Object Models (“DOMs”) of hidden form fields.

[00114] Upon receiving the code, the client device 102 applies the algorithm to the specified data fields. For example, one algorithm may specify that the letter ‘e’ is to be applied after a user types the letter ‘b’ and the number ‘4’ is applied after a user types the number ‘1.’ When the user submits the entered text, the client device 102 transmits the user provided text combined with the programmatically generated keystrokes in a response message. For instance, in the result section 1004 of FIG. 10, the client device 102 may add keystrokes to the user provided username or password.

[00115] A malicious application that uses channel noise may attempt to, alter text, inject text, or additional data fields into the response from the client device 102. However, the security processor 112 is able to identify which text was affected by the malicious application based upon which of the received text does not match the algorithm-based keystroke prediction. As a result, the security processor 112 is able to detect the malicious application.

Soft Messaging using Function Modification

[00116] In a further embodiment, the code section 1002 may include code that changes a library definition of one or more functions. For example, the code section 1002 could specify that a function named ‘add’ is to perform division and that a function named ‘subtract’ is to perform addition. The security processor 112 uses the library definitions to

form a prediction of a response from a client device 102. The security processor 112 transmits the library definition through, for example, xml code, java code, etc.

[00117] Upon receiving the code, the client device 102 applies the changed library definitions to the specified data fields in, for example, the result section 1004 of FIG. 10. In one instance, the client device 102 may prompt a user to enter a result of a mathematical equation as part of an authentication process or when entering a number of related transactions. A malicious application, attempting to alter the authentication or inject additional transactions, examines the response from the client device 102. The malicious application only sees, at most, the name of the function performed, not the definition of the function. As a result, the malicious application alters the data or applies transactions consistent with the name of the function. However, the security processor 112 is able to detect the malicious application because the received altered response would not be consistent with the functional definitions stored to the prediction.

Soft Messaging using Un-rendered Page Elements

[00118] FIG. 11 shows client device 102 including rendered information 1102 and un-rendered information 1104 as a variation of soft messaging. The rendered information 1102 is displayed to a user by the client device 102 while the un-rendered information 1104 is not displayed but instead is included within source code of soft information for a document. The security processor 112 uses the un-rendered information 1104 to determine if a malicious application is affecting communications with the client device 102. For instance, the security processor 112 detects a malicious application if an altered response from the client includes reference to the un-rendered information 1104 or accommodates the un-rendered information 1104. While FIG. 11 shows the un-rendered information 1104 as titles, the un-rendered information 1104 may also include redundant/multiple passwords, redundant/multiple forms, or redundant/multiple logical structures in DOM.

[00119] Generally, malicious applications use un-rendered, machine-readable source code to perform functions instead of the rendered version of the code. The reason is that rendering the code takes additional time and resources that may expose the malicious application. In the example shown, soft information applied to the source code by the security processor 112 enables the introduction of title and tag variations, redundancies, substitutions, embedded requests for data downloads from arbitrary locations, logical obfuscations, piecewise delivery of a final edition of machine-readable source code, transformations of the machine-readable source code based on features of previous or

currently rendered pages, transformations of the machine-readable source code based on intended client interactions with previous or currently rendered pages, etc. in the machine source code version of the page.

[00120] The soft modifications applied by the security processor 112 to the machine-readable source code produce a consistent, useable, non-varied rendered page to the intended user while producing a different varied page to the malicious application. In this manner, the intended user interacts freely with the rendered page while the attempts of the malicious application to interact with the un-rendered, machine-readable source results in a failure to interact with the source code. The un-rendered information 1104 may also cause the malicious application to experience excessively long task completion times.

[00121] Any modifications or alterations performed by a malicious application result in the activation of placeholder source page elements, which are processed and returned to the security processor 112 as indications that the returned information is based on an edition of the machine source code that was not the final edition intended for the end user. Additionally, the security processor 112 is able to detect that a malicious application altered a response from the client device 102 when the received information includes data with geographic locations or bogus data fields that correspond to the soft information of the un-rendered information 1104. For instance, the security processor 112 detects a malicious application if the response from the client device 102 includes a payee after the 'Online Poker' payee.

[00122] In addition to using data fields of un-rendered information 1104, the security processor 112 can also use behind-the-scenes, un-rendered, machine-readable source code used to generate communications. The security processor 112 may also use decision process interfaces for the intended client device 102 in technologies where the communications occur via physical medium and protocols other than HTTP traffic traveling through the network 110. Some of these communication examples include Short Message Service ("SMS") messaging, manufacturing control process signals and protocols (e.g., Foundation Fieldbus, Profibus, or Hart Communication Protocol), and/or infrared or Bluetooth-based communications. The soft messaging techniques may be used by the security processor 112 when the delivery mechanism is not Internet/HTTP based as a way to differentiate between end user presentation, end user interface level and the machine source level of response, and/or interaction with delivered content or information.

[00123] In instances when a malicious application uses the interactions and/or input of a legitimate user via a client device 102 as a means to guide itself through the logical

flow of the obfuscated, machine-readable source code, the security processor 112 may use soft information that includes the creation of additional “user” input events by the system. Examples of these user input events can include, but are not limited to, keyboard events, user focus events, mouse clicks, mouse rollovers, cursor movements etc. The specific details of the soft information or messaging generated user events are known prior to the security processor 112 as the prediction and may be later removed by server 104 or the security processor 112 to recover the legitimate client device 102 and/or end users intent.

[00124] Additionally, in instances when a malicious application exports machine-readable source code to be rendered for processing and/or navigation by a substitute recipient, the security processor 112 can use soft messaging variations among an operating system, a layout engine, a browser, Cascading Style Sheets (“CSS”), java script, bugz, and/or peculiarities acting individually or in combination so that the exported source code compiles and/or renders differently for the substitute client than it does for the originally intended end user. The just-in-time nature of the delivery of the final edition of the machine-readable source code to the intended client device 102 also differentiates between page versions, content versions compiled, and/or rendered at the communicating client device 102. The communicating client device 102 may be the original, intended client or a substitute of the malicious application. The substitute client device may be a computer program and/or technology that replicates the intended end user’s powers of observation, recognition and/or understanding.

Soft Messaging using Graphical Elements

[00125] FIG. 12 shows a client device 102 conducting a transaction with a server 104. The transaction is displayed in datagram 1202 and includes three separate transactions totaling an amount of 268.55. In this example, a malicious application 304 intercepts the transmission of the datagram 1202 from the client device 102 to the server 104. The malicious application 304 uses channel noise to add a fourth transaction and a new balance of 332.89 to the datagram 1202. As a result, the server 104 receives four transactions and the correctly appearing balance of 332.89.

[00126] To prevent such fraud, the security processor 112 uses graphical elements 1204 as soft information to verify the data transmitted by the client 102. The use of graphical elements 1204 enables the security processor 112 to validate channel communications when a client device 102 is the originator of hard and soft information. In other words, the security processor 112 uses graphical elements 1204 to confirm communications with the client device 102 when the security processor 112 may not be able

to form a prediction because the client device is the originator soft and/or hard information. The graphical elements 1204 may be presented to the user of the client device 102 as, for example, a banner, background, image, part of an advertisement, or a video. In some examples, the security processor 112 can use variations in graphical elements 1204 as soft information in conjunction with other soft messaging techniques discussed above.

[00127] In the illustrated example of FIG. 12, the security processor 112 transmits the graphical element 1204 to the client device 102. The security processor 112 embeds the balance information as code included within the graphic, which helps prevent the malicious application 304 from detecting and using channel noise to alter the balance to the amount provided originally by the client device 102. The client device 102 accordingly displays the graphical element 1204 including the balance received by the server 104. The user can then compare the balances and provide feedback that the balances do not match by, for instance, selecting the graphical element 1204. In response, the security processor 112 instructs the server 104 to disregard the datagram 1202.

[00128] In an alternative embodiment, the security processor 112 enables the client device 102 to supply comparison information. For example, a 'submit these transactions' button may be presented by the client device 102 as an active, account balance indexed grid. A user of the device 102 is expected to activate that portion of the button corresponding to the traditionally displayed account balance. As in the previous examples, the details of this button may be session dependent.

[00129] In another example, the client device 102 may be enabled by the security processor 112 to send a screen capture of the account information in the datagram 1202 to the server 104 for automated comparison by the security processor 112. The background and other features of the screen capture may be session dependent to prevent counterfeiting. For example the security processor 112 may specify in soft messaging whether the client device 102 is to create and forward a snapshot of the top 2/3 of an account balance or the lower 2/3 of the account balance and/or a blank image followed by the account balance.

Multiple Predictions for a Single Session Embodiment

[00130] FIG. 13 shows an illustration of two different configurations of a client device 102 that can be accounted for by the security processor 112 to create multiple predictions in some embodiments. In this example, the security processor 112 creates two different predictions based on an orientation of client device 102. The first prediction

corresponds to the client device 102 being in a vertical orientation 1302 and the second prediction corresponds to the client device being in a horizontal orientation 1304.

[00131] Oftentimes, many smartphones and tablet computers can display information based on how the device is orientated. However, the orientation of the device is generally not reported back to a server 104 through backscattered secondary information. As a result, the server 104 does not know the orientation of the device when the hard information is displayed. To compensate for this lack of information, the security processor 112 creates two different predictions. In some embodiments, the security processor 112 may generate, by default, multiple predictions regardless of a type of client device 102 to account for different screen sizes, orientations, etc. In other embodiments, the security processor 112 may generate a second prediction only after receiving backscatter information that indicates the client device 102 corresponds to a type of device that can have more than one orientation.

[00132] In the illustrated example of FIG. 13, the security processor 112 creates a first prediction as to how the hard information (e.g., username, password and 'submit button') is displayed based on the received soft information. The security processor 112 determines that coordinates of the features displayed by client device 102 have to fit within the vertical orientation 1302 of the client device 102. Similarly, the security processor 112 creates a second prediction as to how the hard information will be displayed within the horizontal orientation 1304. The differences between the orientations 1302, 1304 can include spacing between data fields, sizes of the data fields, location of the 'submit' button, and a location of the trademark. The security processor 112 then compares a response from the client device 102 to each of the predictions to determine if a malicious application is affecting communications.

Prediction Data Structure

[00133] FIG. 14 shows a diagram of a data structure 1400 of a prediction formed by the security processor 112 based on soft information and secondary information acquired from global and local observable temporal channel information. The data structure 1400 is representative of information used by the security processor 112 to form the prediction. In other embodiments, the security processor 112 may render a webpage based on the soft and secondary information, similar to the datagram 510 of FIGS. 5 and 7.

[00134] The example security processor 112 uses the information in data structure 1400 to determine if a response from a client device 102 is indicative of a malicious application affecting communications. The security processor 112 creates the data structure 1400 by storing soft information used in soft messaging by a server 104. The security

processor 112 supplements the data structure 1400 with secondary information received as backscatter information. As mentioned before, the soft information describes how hard information is displayed or presented while the secondary information provides indications how the soft and hard information are to be displayed on a client device 102.

[00135] In the illustrative example of FIG. 14, the soft information includes font type, font size, and positioning of three text fields. The soft information also includes coordinates of a 'submit' button including an allowable deviation or predetermined threshold. The soft information further includes programmed text to be generated automatically in the text fields and a location of a banner graphical element. In addition, the soft information includes un-rendered text at specified coordinates.

[00136] Also in the data structure 1300 of FIG. 14, the secondary information includes a browser type and operating system of the client device 102. The secondary information also includes an indication that java script is enabled. The security processor 112 uses the secondary information to modify the soft information as needed. For example, upon receiving an indication that a client device 102 is using an OPPS browser, the security processor 112 updates coordinates of the text fields and 'submit' button to reflect how the OPPS browser is known to format and render text and graphics. In this manner, the secondary information is used by the security processor 112 to refine or alter the initial prediction made when the soft information was initially transmitted to the client device 102.

Flowchart of the Example Process

[00137] FIGS. 15 and 16 are a flow diagram showing example procedures 1500, 1530, and 1560 to validate a communication channel, according to an example embodiment of the present invention. Although the procedures 1500, 1530, and 1560 are described with reference to the flow diagram illustrated in FIGS. 15 and 16, it will be appreciated that many other methods of performing the acts associated with the procedures 1500, 1530, and 1560 may be used. For example, the order of many of the blocks may be changed, certain blocks may be combined with other blocks, and many of the blocks described are optional.

[00138] The example procedure 1500 operates on, for example, the client device 102 of FIGS. 1 to 6. The procedure 1500 begins when the client device 102 transmits a connection request to a server 104 (block 1502). The connection request can include a website address or IP address that is routed by the network 110 to the appropriate server 104. The connection request can also include device information identifying secondary characteristics or information associated with the client device 102.

[00139] After receiving a connection response, the client device 102 requests to engage in a data transaction with the server 104 (block 1504). The request can include a specification of information that the client device 102 desires to read or write to information stored in a database or managed by the server 104. The request can also include one or more transactions that the client device 102 desires to complete with the server 104.

[00140] Some time after transmitting the request, the client device 102 receives hard and soft information 1507 corresponding to the requested transaction (block 1506). The hard and soft information 1507 can be received in separate messages or combined together in one or more messages. The client device 102 uses the soft information to determine how the hard information is to be rendered and displayed (block 1508). After displaying the hard information, the client device 102 transmits a response message 1509 provided by a user (block 1510). At this point, the example procedure 1500 ends when the client device 102 and server 104 stop exchanging communications (e.g., terminate a communication session). Additionally, in some embodiments, the client device 102 may receive an indication from the server 104 that a malicious application has affected at least the information in the response message 1509. As a result, the client device 102 could re-authenticate communications with the server 104 or enter a failsafe mode.

[00141] The example procedure 1530 of FIG. 15 operates on, for example, the application server 104 of FIGS. 1 to 6. The procedure begins when the server 104 receives a connection request from a client device 102 (block 1532). In instances that the connection request includes device information, the server 104 transmits the device information to a communicatively coupled security processor 112. The server 104 then transmits a connection response to the client device 102, thereby initiating a communication session (block 1534).

[00142] Some time later, the server 104 receives from the client device 102 a request to process a data transaction (block 1536). The server 104 then determines hard information 1537 associated with the requested data transaction (block 1538). For example, a request to access an account causes the server 104 to identify account log-in information. In another example, a request to perform a banking transaction causes the server 104 to identify account information and available banking options for the account. The server 104 then transmits the determined hard information 1537 to a security processor 112. In some embodiments, the security processor 112 may be instantiated within the server 104. In other embodiments, the security processor 112 may be remote from the server 104.

[00143] Responsive to receiving hard and soft information 1507 from the security processor 112, the server 104 formats and transmits the information 1507 to the

client device 102 (block 1540). In some embodiments, the server 104 receives messages with combined hard and soft information. In these embodiments, the server 104 formats the messages (e.g., structures the messages into data packets) for transmission. In other embodiments, the server 104 receives the hard and soft information. In these other embodiments, the server 104 combines the hard and soft information into one or more messages and formats these messages for transmission. The server 104 accordingly provides the client device 102 with hard and soft messaging.

[00144] After transmitting the hard and soft information 1507, the server 104 of FIG. 16 receives backscattered information 1543 from channel components used to process, route, and render the information 1507 (block 1542). The server 104 transmits this backscattered information 1543 to the security processor 112. In some instances, the server 104 transmits the information 1543 as the information is received. In other instances, the server 104 transmits the information 1543 periodically or after receiving an indication that the soft and hard information 1507 has been received and processed by the client device 102.

[00145] The server 104 then receives the response message 1509 from the client device 102 including information responding to the hard information (block 1544). The server 104 subsequently transmits the response message 1509 to the security processor 112. After the security processor 112 has compared information in the response message 1509 to a prediction, the server 104 determines whether the communication session with the client device has been validated (block 1546). If the security processor 112 does not provide an indication of a malicious application, the server 104 determines the communication session with the client device 102 is validated. The server 104 continues communications with the client device 102 and continues to validate communications until the communication session is ended.

[00146] However, responsive to the security processor 112 providing an indication of a malicious application, the server 104 enters a failsafe mode (block 1548). The failsafe mode can include the server 104 informing the client device 102 of the malicious application, requesting that the client device 102 re-authenticate, restricting access to the data transactions associated with the client device 102, transmitting an alarm or alert to appropriate personnel, and/or applying a routine or algorithm to remove or restrict further attempts by the malicious application to affect communications. Regardless of which failsafe operation is performed, the example procedure 1530 ends when the communication session with the client device 102 is terminated or when the effects of the malicious application have been remedied.

[00147] Returning to FIG. 15, the example procedure 1560 operates on, for example, the security processor 112 of FIGS. 1 to 6. The procedure 1560 begins when the security processor 112 receives device information from the server 104 (block 1562). This step can be skipped in instances where a connection request does not include device information. The security processor 112 then receives hard information 1537 from the server 104 and identifies compatible soft information (block 1564). For instance, hard information has a limited number of ways that it can be correctly displayed. The security processor 112 uses this relationship to identify which soft information is compatible with the hard information.

[00148] After identifying the compatible soft information, the security processor 112 selects a variation of the soft information (block 1566). The security processor 112 may select a different variation of soft information for each client device-server connection. As described before, this variation prevents malicious applications from reverse engineering the soft messaging used to validate communications. The security processor 112 then combines the hard information and the selected soft information 1507 into one or more messages and transmits combined information 1507 to the server 104, which then transmits the information 1507 to the client device 102 (block 1568). The security processor 112 also forms a prediction as to how the client device 102 will render and display the hard information based on the soft information (block 1570).

[00149] In FIG. 16, the security processor 112 receives the backscattered information 1543 from the server 104 and determines corresponding secondary information or characteristics (block 1572). The security processor 112 then updates or modifies the prediction based on the secondary information (block 1574). Responsive to receiving the response message 1509 from the client device 102, the security processor 112 compares the information in the response to the prediction (block 1576). The comparison includes determining if soft information returned by the client device 102 matches or is within an allowable deviation to corresponding soft information in the prediction (e.g., matching coordinates of graphics or data fields, matching programmatically entered characters, or matching font information) (block 1578).

[00150] Responsive to determining the information in the response matches the prediction, the security processor 112 validates the communication session between the server 104 and the client device 102 (block 1580). The security processor 112 then continues to validate the communication session for additional communications between the server 104 and the client device 102 until the communication session is ended. Responsive to

determining the information in the response deviates from the prediction, the security processor 112 provides an indication of a malicious application (block 1582). The security processor 112 may also remedy the effects of the malicious application or take steps to prevent the malicious application from affecting further communications between the client device 102 and the server 104. The security processor 112 then continues to validate the communication session for additional communications between the server 104 and the client device 102 until the communication session is ended.

[00151] It will be appreciated that all of the disclosed methods and procedures described herein can be implemented using one or more computer programs or components. These components may be provided as a series of computer instructions on any conventional computer-readable medium, including RAM, ROM, flash memory, magnetic or optical disks, optical memory, or other storage media. The instructions may be configured to be executed by a processor, which when executing the series of computer instructions performs or facilitates the performance of all or part of the disclosed methods and procedures.

[00152] It should be understood that various changes and modifications to the example embodiments described herein will be apparent to those skilled in the art. Such changes and modifications can be made without departing from the spirit and scope of the present subject matter and without diminishing its intended advantages. It is therefore intended that such changes and modifications be covered by the appended claims.

Additional Aspects of the Present Disclosure

[00153] To the above ends, and without limiting the following description, in a first aspect of the present disclosure, a method comprising: selecting transactional information to transmit from a server to a communicatively coupled client device based on a request from the client device, selecting presentation information corresponding to the transactional information to transmit from the server to the client device, the presentation information specifying how the transactional information is to be displayed, transmitting at least one message including the presentation and transactional information from the server to the client device, determining a prediction as to how the client device will render the transactional information based on the presentation information, receiving a response message from the client device, and responsive to information in the response message not matching the prediction, providing an indication there is a malicious application affecting communications between the server and the client device.

[00154] In accordance with a second aspect of the present disclosure, which may be used in combination with the first aspect, the presentation information includes at

least one of protocol information, formatting information, positional information, rendering information, style information, transmission encoding information, and information describing how different layers of a style sheet is to be rendered by the client.

[00155] In accordance with a third aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the presentation information includes information changing a definition of a function in a code library at the client device.

[00156] In accordance with a fourth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the transactional information includes at least one of text, data, or pictorial information.

[00157] In accordance with a fifth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the transactional information includes information requested by the server to perform a service for the client device.

[00158] In accordance with a sixth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the transactional information includes at least one of authentication information, refinement information on a type of service requested by the client, financial information, and data management information.

[00159] In accordance with a seventh aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the presentation information is soft information and the transactional information is hard information.

[00160] In accordance with an eighth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the method further includes selecting the presentation information from among a plurality of different types of presentation information compatible with the transactional information.

[00161] In accordance with a ninth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the method further includes responsive to information in the response message not matching the prediction, restricting access of the client device.

[00162] In accordance with a tenth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the method further includes responsive to information in the response message not matching the prediction, transmitting an alert and entering a safe operations mode.

[00163] In accordance with a eleventh aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the prediction is determined based at least in part on at least one of estimating locations of rendered features and functions displayed by the client device, estimating locations of rendered page geometry of the features and functions, and estimating relative locations between text, input boxes, buttons, and advertisements.

[00164] In accordance with a twelfth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the method further includes receiving backscatter information transmitted from routing devices within a channel between the server and the client device as a result of routing the at least one message and modifying the prediction based at least in part on temporal characteristics of the channel indicted in the backscatter information.

[00165] In accordance with a thirteenth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, temporal characteristics provide an indication how an operating system of the client device is to process the presentation information and how the operating system of the client device is to process Extensible Markup Language code in the presentation information.

[00166] In accordance with a fourteenth aspect of the present disclosure, which may be used in combination with the thirteenth aspect, the temporal characteristics include at least one of a number of discrete transmissions for the presentation and transactional information, a size of the discrete transmissions for the presentation and transactional information, a density of the discrete transmissions of the presentation and transactional information, delivery times of the presentation and transactional information, and delivery rates of the presentation and transactional information.

[00167] In accordance with a fifteenth aspect of the present disclosure, which may be used in combination with the thirteenth aspect, the temporal characteristics include local observables associated with the client device.

[00168] In accordance with a sixteenth aspect of the present disclosure, which may be used in combination with the fifteenth aspect, the local observables include at least one of a processing prioritization rule set of the client, protocols of the client, and an operating system of the client device.

[00169] In accordance with a seventeenth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the

response message includes coordinates of displayed information or features rendered by the client device.

[00170] In accordance with a eighteenth aspect of the present disclosure, which may be used in combination with the seventeenth aspect, the response message includes at least one of coordinates of a mouse click of a button, coordinates of data entered into a data field, a font size of entered data, a font type of entered data entered, data in un-displayed form fields, and programmatically generated keystrokes based on entered data.

[00171] In accordance with a nineteenth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the method further includes determining a second prediction as to how the client device will render the transactional information based on the presentation information, the second prediction being based on an alternative display of the client device and responsive to the information in the response message not matching the second prediction or the prediction, providing the indication of the malicious application affecting the communications between the server and the client device.

[00172] In accordance with a twentieth aspect of the present disclosure, which may be used in combination with the nineteenth aspect, the alternative display of the client device includes a different screen orientation of the client device.

[00173] In accordance with a twenty-first aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the transactional information is transmitted in messages separate from the presentation information.

[00174] In accordance with a twenty-second aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, a machine-accessible device has instructions stored thereon that, when executed, cause a machine to at least: receive a request from a client device to communicatively couple with an application server, select transactional information to transmit from the application server to the client device based on the request from the client device, the transactional information including transactional data, select presentation information from among a plurality of different presentation information that is compatible with the requested transactional data, the presentation information specifying how the request for transactional data is to be displayed, transmit at least one message including the presentation and transactional information from the server to the client device, causing the client device to display the request for transactional data, determine a prediction as to how the client device will render the

transactional information based on the presentation information, receive a response message from the client device, compare information within the response message to the prediction, and responsive to the information in the response message not matching the prediction, provide an indication there is a malicious application affecting communications between the server and the client device.

[00175] In accordance with a twenty-third aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the instructions further cause the machine to at least receive backscatter information transmitted from routing devices within a channel between the server and the client device as a result of routing the at least one message and modify the prediction based at least in part on temporal characteristics of the channel indicated in the backscatter information.

[00176] In accordance with a twenty-fourth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the instructions further cause the machine to at least determine in the prediction that coordinates of a button correspond to a first location, receive in the response message that the coordinates of a mouse click of the button correspond to a second location, responsive to a number of pixels between the first location and the second location exceeding a predetermined threshold, determine the second location does not match the first location, and responsive to the second location not matching the first location, provide the indication of the malicious application.

[00177] In accordance with a twenty-fifth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the instructions further cause the machine to at least determine in the prediction that a data field is to correspond to a first font size and a first font type, receive in the response message that data corresponding to the data field is a second font size and a second font type, responsive to the second font size not matching the first font size, provide the indication of the malicious application, and responsive to the second font type not matching the first font type, provide the indication of the malicious application.

[00178] In accordance with a twenty-sixth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the instructions further cause the machine to at least determine in the prediction that for a data field, keystrokes are to be augmented by programmatically generated keystrokes, receive in the response message data corresponding to the data field, and responsive to the received data

not including the programmatically generated keystrokes based on the keystrokes, provide the indication of the malicious application.

[00179] In accordance with a twenty-seventh aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the instructions further cause the machine to at least select the presentation information that includes a alteration of functions in a code library, determine in the prediction that a result of the functions is to be based on the alteration, receive in the response message data corresponding to the function, and responsive to the received data not matching the result of the functions based on the alteration, provide the indication of the malicious application.

[00180] In accordance with a twenty-eighth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, the instructions further cause the machine to at least select the presentation information that includes a data field to be displayed at a first location, determine in the prediction coordinates of the data field at the first location, receive in the response message that coordinates of data provided in the data field correspond to a second location, responsive to a number of pixels between the first location and the second location exceeding a predetermined threshold, determine the second location does not match the first location, and responsive to the second location not matching the first location, provide the indication of the malicious application.

[00181] In accordance with a twenty-ninth aspect of the present disclosure, which may be used in combination with any one or more of the preceding aspects, an apparatus comprises a server configured to: communicatively couple to a client device in response to receiving a connection request to connect from the client device, select transactional information to transmit to the client device based on a transaction request received from the client device, transmit at least one message to the client device, and receive a response message from the client device responsive to the at least one message and a security processor communicatively coupled to the server and configured to: select presentation information corresponding to the transactional information, select a variation of the presentation information, combine the selected presentation and transactional information into the at least one message, determine a prediction as to how the client device will render the transactional information based on the presentation information and responsive to information in the response message from the client device not matching the prediction, provide an indication there is a malicious application affecting communications between the server and the client device.

[00182] In accordance with a thirtieth aspect of the present disclosure, which may be used with any one or more of the preceding aspects in combination with the twenty-ninth aspect, the server is to receive backscatter information transmitted from routing devices within a channel between the server and the client device as a result of routing the at least one message and the security processor is to modify the prediction based at least in part on temporal characteristics of the channel indicated in the backscatter information.

[00183] In accordance with a thirty-first aspect of the present disclosure, which may be used with any one or more of the preceding aspects in combination with the twenty-ninth aspect, the security processor is to select the presentation information from among a plurality of different types of presentation information compatible with the transactional information.

[00184] In accordance with a thirty-second aspect of the present disclosure, which may be used with any one or more of the preceding aspects in combination with the twenty-ninth aspect, the security processor is to determine a second prediction as to how the client device will render the transactional information based on the presentation information, the second prediction being based on an alternative display of the client device and responsive to the information in the response message not matching the second prediction or the prediction, provide the indication of the malicious application affecting the communications between the server and the client device.

[00185] In accordance with a thirty-third aspect of the present disclosure, which may be used with any one or more of the preceding aspects in combination with the twenty-ninth aspect, the security processor is remotely located from the server.

[00186] In accordance with a thirty-fourth aspect of the present disclosure, which may be used with any one or more of the preceding aspects in combination with the twenty-ninth aspect, the server includes at least one of an application server or a database server.

[00187] In accordance with a thirty-fifth aspect of the present disclosure, any of the structure and functionality illustrated and described in connection with FIGS. 1 to 17 may be used in combination with any of the structure and functionality illustrated and described in connection with any of the other of FIGS. 1 to 17 and with any one or more of the preceding aspects.

CLAIMS

The invention is claimed as follows:

1. A method comprising:
 - selecting transactional information to transmit from a server to a communicatively coupled client device based on a request from the client device;
 - selecting presentation information corresponding to the transactional information to transmit from the server to the client device, the presentation information specifying how the transactional information is to be displayed;
 - transmitting at least one message including the presentation and transactional information from the server to the client device;
 - determining a prediction as to how the client device will render the transactional information based on the presentation information;
 - receiving a response message from the client device; and
 - responsive to information in the response message not matching the prediction, providing an indication there is a malicious application affecting communications between the server and the client device.
2. The method of Claim 1, wherein the presentation information includes at least one of protocol information, formatting information, positional information, rendering information, style information, transmission encoding information, and information describing how different layers of a style sheet is to be rendered by the client.
3. The method of Claim 1, wherein the presentation information includes information changing a definition of a function in a code library at the client device.
4. The method of Claim 1, wherein the transactional information includes at least one of text, data, or pictorial information.
5. The method of Claim 1, wherein the transactional information includes information requested by the server to perform a service for the client device.

6. The method of Claim 1, wherein the transactional information includes at least one of authentication information, refinement information on a type of service requested by the client, financial information, and data management information.

7. The method of Claim 1, wherein the presentation information is soft information and the transactional information is hard information.

8. The method of Claim 1, further comprising selecting the presentation information from among a plurality of different types of presentation information compatible with the transactional information.

9. The method of Claim 1, further comprising responsive to information in the response message not matching the prediction, restricting access of the client device.

10. The method of Claim 1, further comprising responsive to information in the response message not matching the prediction, transmitting an alert and entering a safe operations mode.

11. The method of Claim 1, wherein the prediction is determined based at least in part on at least one of estimating locations of rendered features and functions displayed by the client device, estimating locations of rendered page geometry of the features and functions, and estimating relative locations between text, input boxes, buttons, and advertisements.

12. The method of Claim 1, further comprising:
receiving backscatter information transmitted from routing devices within a channel between the server and the client device as a result of routing the at least one message; and
modifying the prediction based at least in part on temporal characteristics of the channel indicated in the backscatter information.

13. The method of Claim 1, wherein temporal characteristics provide an indication how an operating system of the client device is to process the presentation information and

how the operating system of the client device is to process Extensible Markup Language code in the presentation information.

14. The method of Claim 13, wherein the temporal characteristics include at least one of a number of discrete transmissions for the presentation and transactional information, a size of the discrete transmissions for the presentation and transactional information, a density of the discrete transmissions of the presentation and transactional information, delivery times of the presentation and transactional information, and delivery rates of the presentation and transactional information.

15. The method of Claim 13, wherein the temporal characteristics include local observables associated with the client device.

16. The method of Claim 15, wherein the local observables include at least one of a processing prioritization rule set of the client, protocols of the client, and an operating system of the client device.

17. The method of Claim 1, wherein the response message includes coordinates of displayed information or features rendered by the client device.

18. The method of Claim 17, wherein the response message includes at least one of coordinates of a mouse click of a button, coordinates of data entered into a data field, a font size of entered data, a font type of entered data entered, data in un-displayed form fields, and programmatically generated keystrokes based on entered data.

19. The method of Claim 1, further comprising:
determining a second prediction as to how the client device will render the transactional information based on the presentation information, the second prediction being based on an alternative display of the client device;
responsive to the information in the response message not matching the second prediction or the prediction, providing the indication of the malicious application affecting the communications between the server and the client device.

20. The method of Claim 19, wherein the alternative display of the client device includes a different screen orientation of the client device.

21. The method of Claim 1, wherein the transactional information is transmitted in messages separate from the presentation information.

22. A machine-accessible device having instructions stored thereon that, when executed, cause a machine to at least:

- receive a request from a client device to communicatively couple with an application server;

- select transactional information to transmit from the application server to the client device based on the request from the client device, the transactional information including transactional data;

- select presentation information from among a plurality of different presentation information that is compatible with the requested transactional data, the presentation information specifying how the request for transactional data is to be displayed;

- transmit at least one message including the presentation and transactional information from the server to the client device, causing the client device to display the request for transactional data;

- determine a prediction as to how the client device will render the transactional information based on the presentation information;

- receive a response message from the client device;

- compare information within the response message to the prediction; and

- responsive to the information in the response message not matching the prediction, provide an indication there is a malicious application affecting communications between the server and the client device.

23. The machine-accessible device of Claim 22, further comprising instructions stored thereon that are configured when executed to cause a machine to at least:

- receive backscatter information transmitted from routing devices within a channel between the server and the client device as a result of routing the at least one message; and

- modify the prediction based at least in part on temporal characteristics of the channel indicated in the backscatter information.

24. The machine-accessible device of Claim 22, further comprising instructions stored thereon that are configured when executed to cause a machine to at least:

determine in the prediction that coordinates of a button correspond to a first location;
receive in the response message that the coordinates of a mouse click of the button correspond to a second location;

responsive to a number of pixels between the first location and the second location exceeding a predetermined threshold, determine the second location does not match the first location; and

responsive to the second location not matching the first location, provide the indication of the malicious application.

25. The machine-accessible device of Claim 22, further comprising instructions stored thereon that are configured when executed to cause a machine to at least:

determine in the prediction that a data field is to correspond to a first font size and a first font type;

receive in the response message that data corresponding to the data field is a second font size and a second font type;

responsive to the second font size not matching the first font size, provide the indication of the malicious application; and

responsive to the second font type not matching the first font type, provide the indication of the malicious application.

26. The machine-accessible device of Claim 22, further comprising instructions stored thereon that are configured when executed to cause a machine to at least:

determine in the prediction that for a data field, keystrokes are to be augmented by programmatically generated keystrokes;

receive in the response message data corresponding to the data field; and

responsive to the received data not including the programmatically generated keystrokes based on the keystrokes, provide the indication of the malicious application.

27. The machine-accessible device of Claim 22, further comprising instructions stored thereon that are configured when executed to cause a machine to at least:

select the presentation information that includes an alteration of functions in a code library;

determine in the prediction that a result of the functions is to be based on the alteration;

receive in the response message data corresponding to the function; and

responsive to the received data not matching the result of the functions based on the alteration, provide the indication of the malicious application.

28. The machine-accessible device of Claim 22, further comprising instructions stored thereon that are configured when executed to cause a machine to at least:

select the presentation information that includes a data field to be displayed at a first location;

determine in the prediction coordinates of the data field at the first location;

receive in the response message that coordinates of data provided in the data field correspond to a second location;

responsive to a number of pixels between the first location and the second location exceeding a predetermined threshold, determine the second location does not match the first location; and

responsive to the second location not matching the first location, provide the indication of the malicious application.

29. An apparatus comprising:

a server configured to:

communicatively couple to a client device in response to receiving a connection request to connect from the client device;

select transactional information to transmit to the client device based on a transaction request received from the client device;

transmit at least one message to the client device; and

receive a response message from the client device responsive to the at least one message; and

a security processor communicatively coupled to the server and configured to:

select presentation information corresponding to the transactional information;

select a variation of the presentation information;

combine the selected presentation and transactional information into the at least one message;

determine a prediction as to how the client device will render the transactional information based on the presentation information; and

responsive to information in the response message from the client device not matching the prediction, provide an indication there is a malicious application affecting communications between the server and the client device.

30. The apparatus of Claim 29, wherein:

the server is to receive backscatter information transmitted from routing devices within a channel between the server and the client device as a result of routing the at least one message; and

the security processor is to modify the prediction based at least in part on temporal characteristics of the channel indicated in the backscatter information.

31. The apparatus of Claim 29, wherein the security processor is to select the presentation information from among a plurality of different types of presentation information compatible with the transactional information.

32. The apparatus of Claim 29, wherein the security processor is to:

determine a second prediction as to how the client device will render the transactional information based on the presentation information, the second prediction being based on an alternative display of the client device; and

responsive to the information in the response message not matching the second prediction or the prediction, provide the indication of the malicious application affecting the communications between the server and the client device.

33. The apparatus of Claim 29, wherein the security processor is remotely located from the server.

34. The apparatus of Claim 29, wherein the server includes at least one of an application server or a database server.

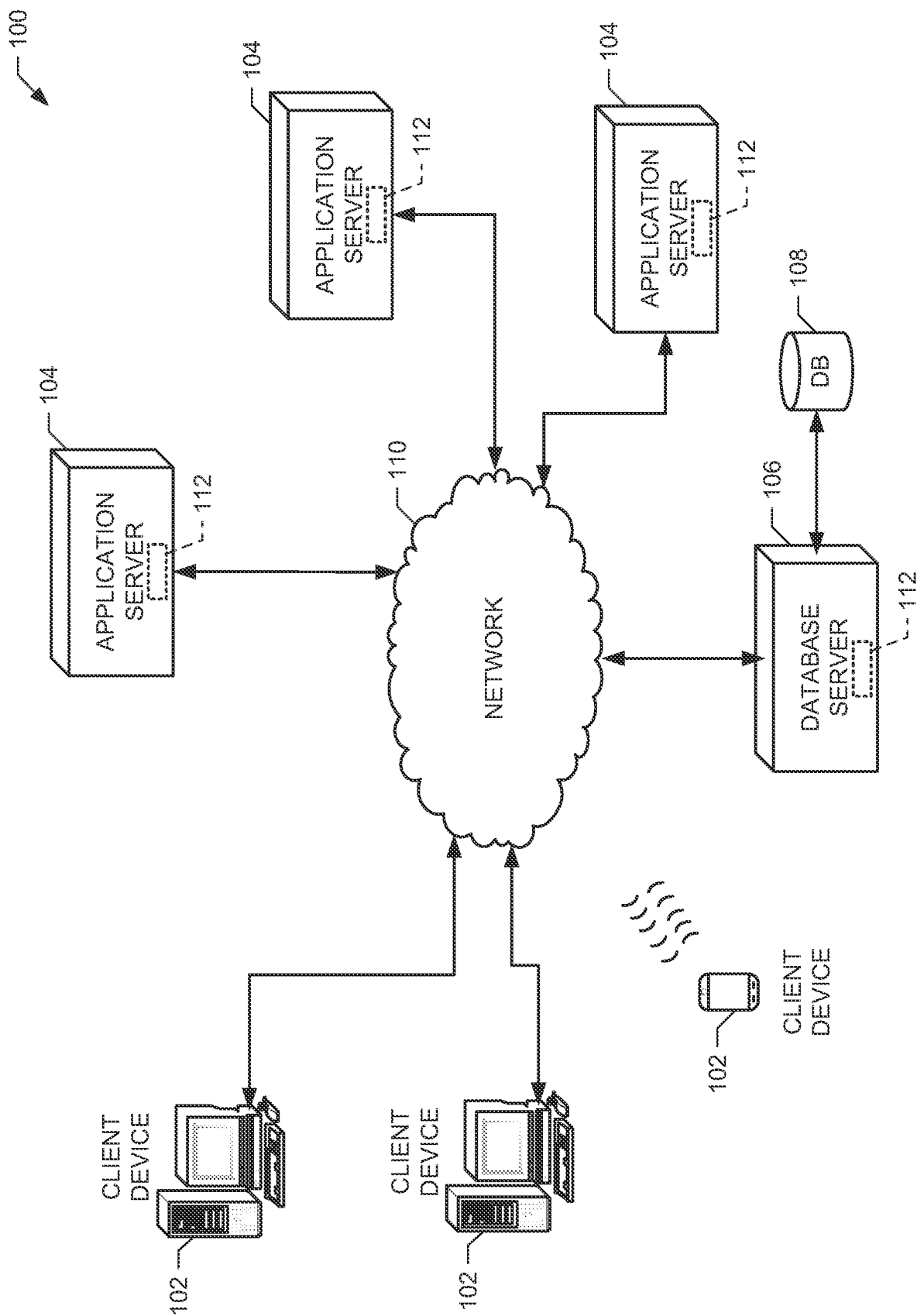


FIG. 1

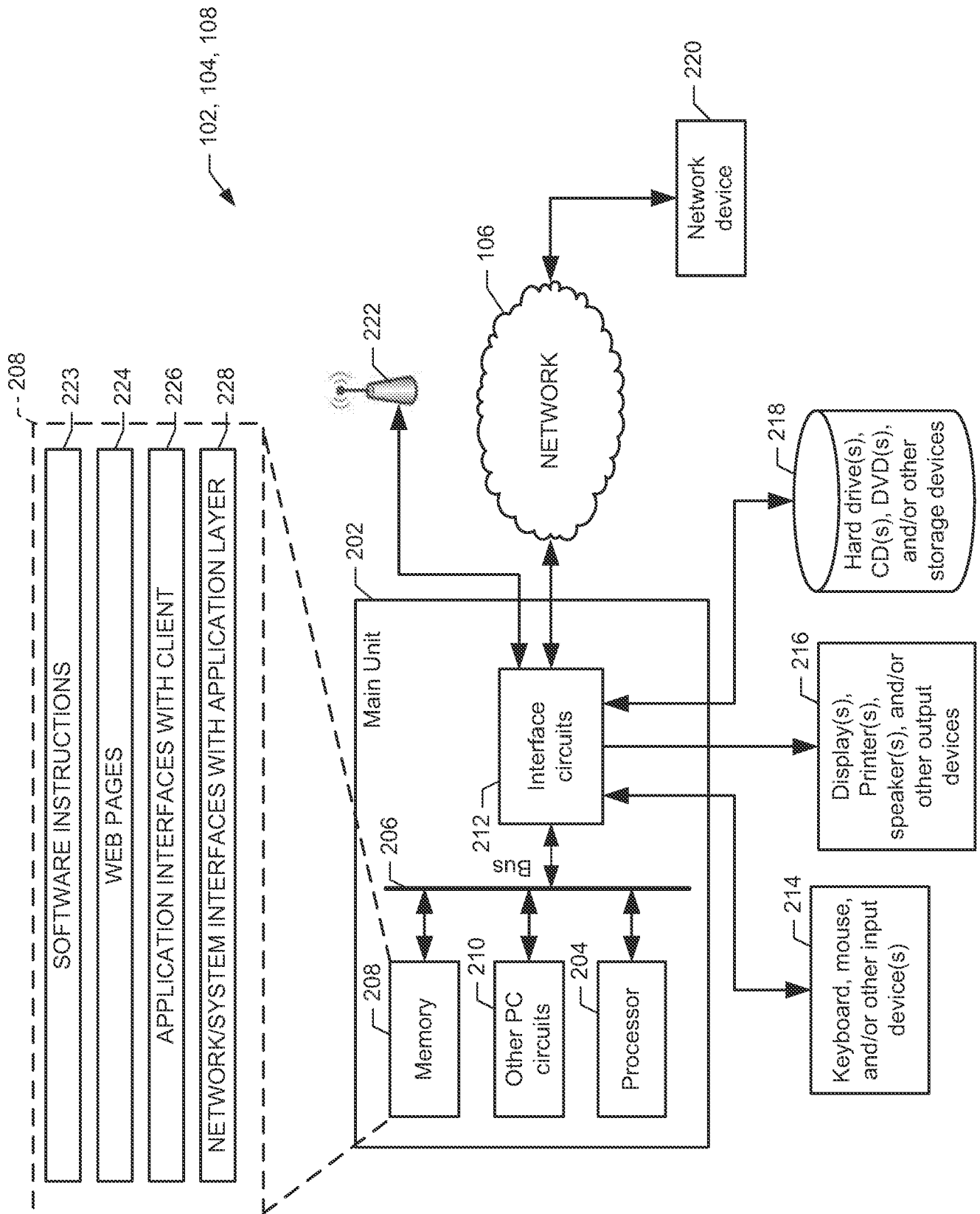


FIG. 2

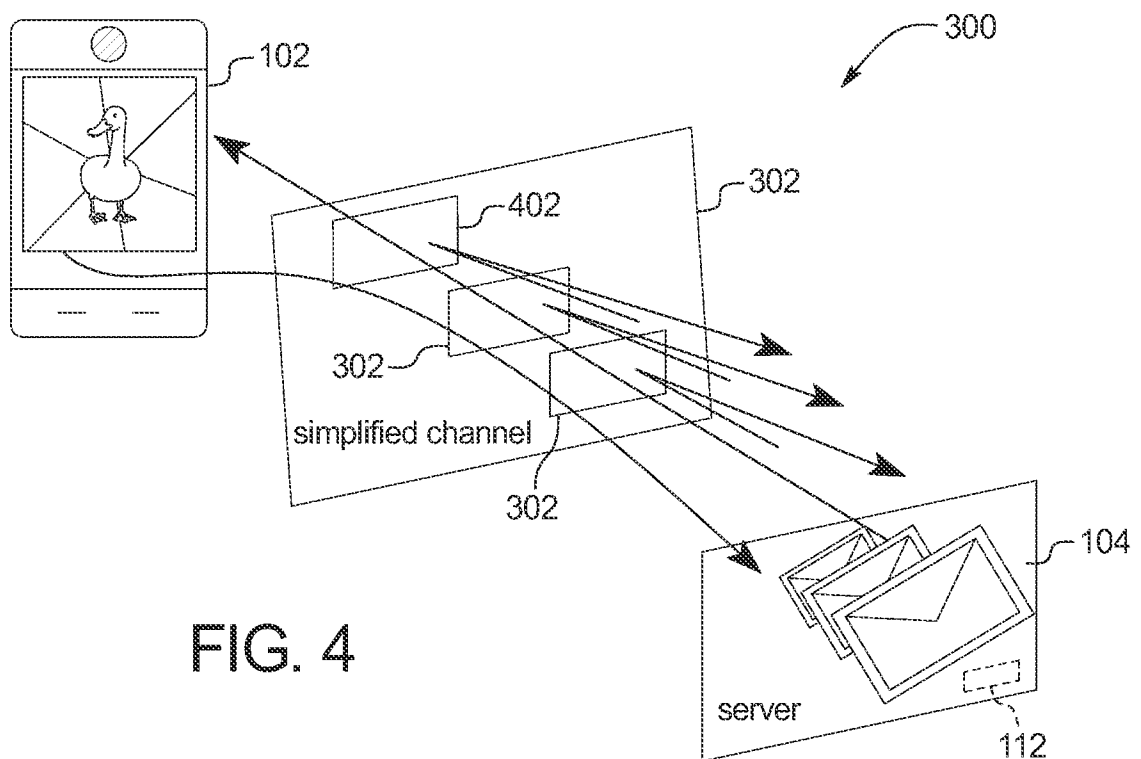
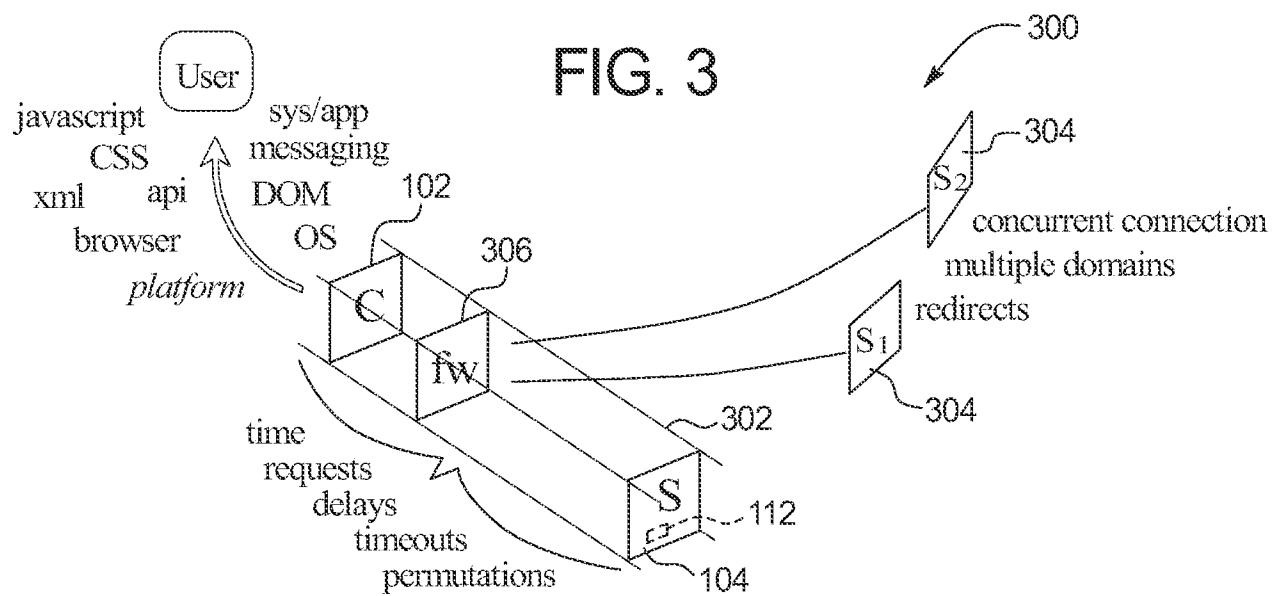


FIG. 5

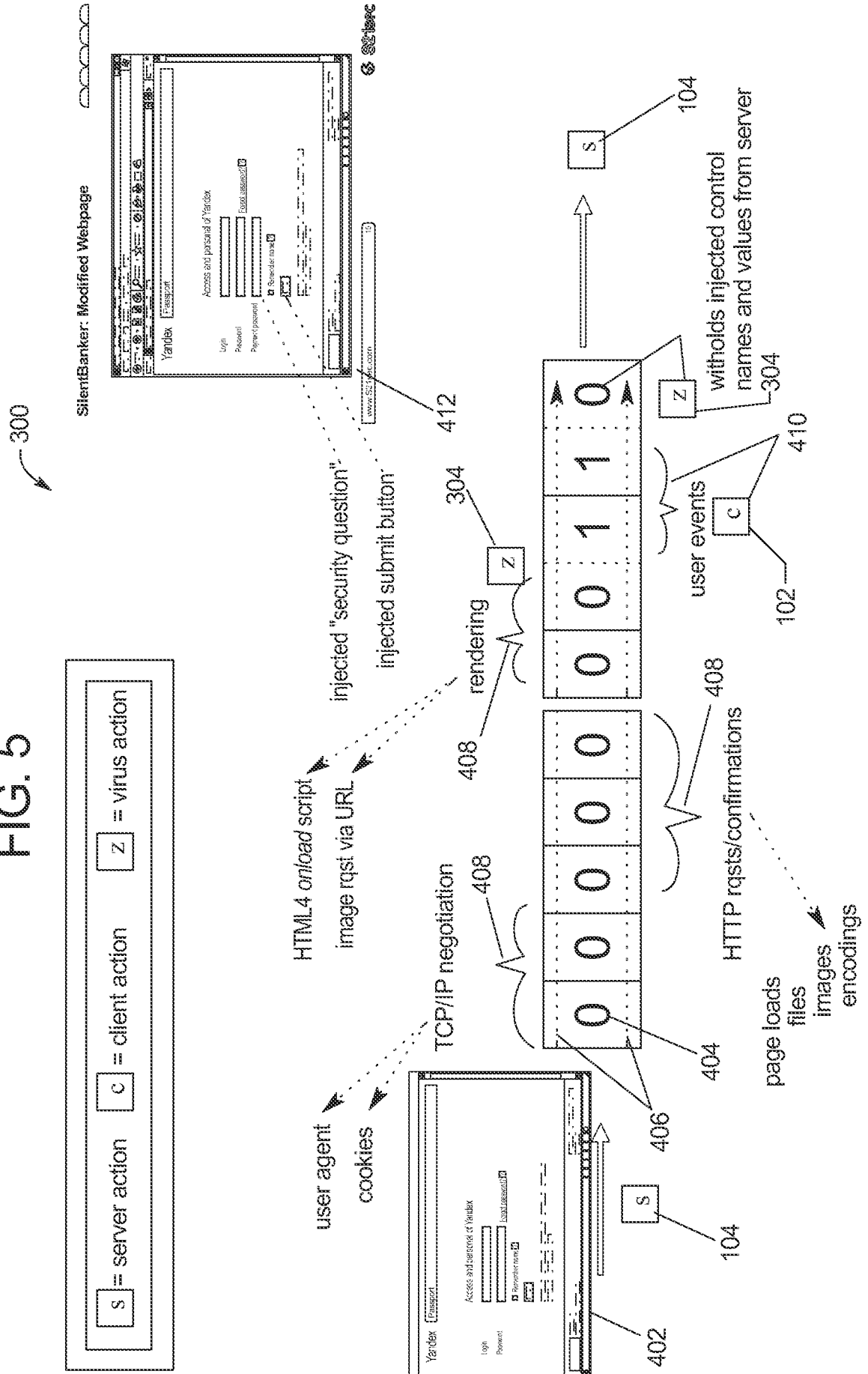
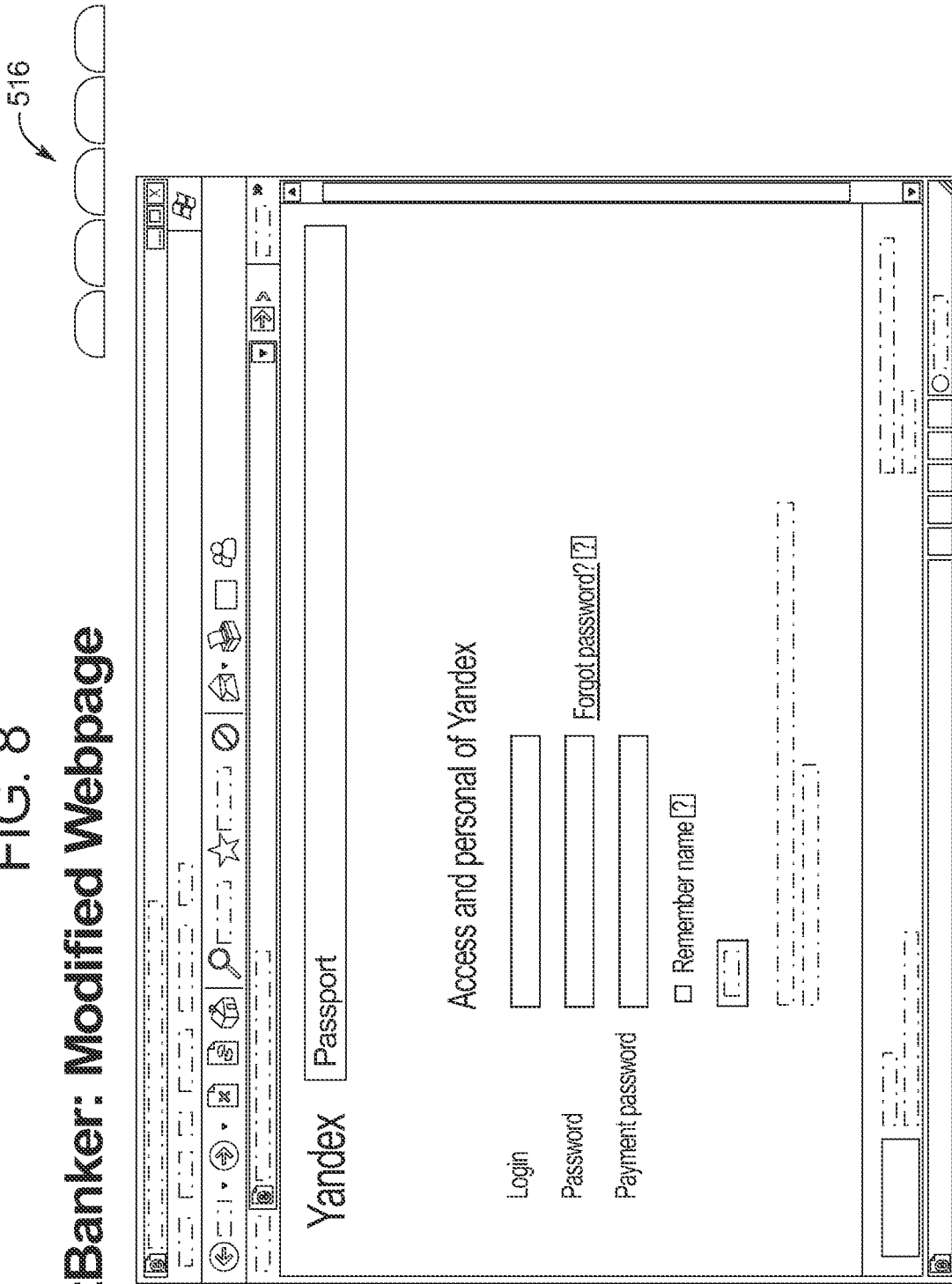
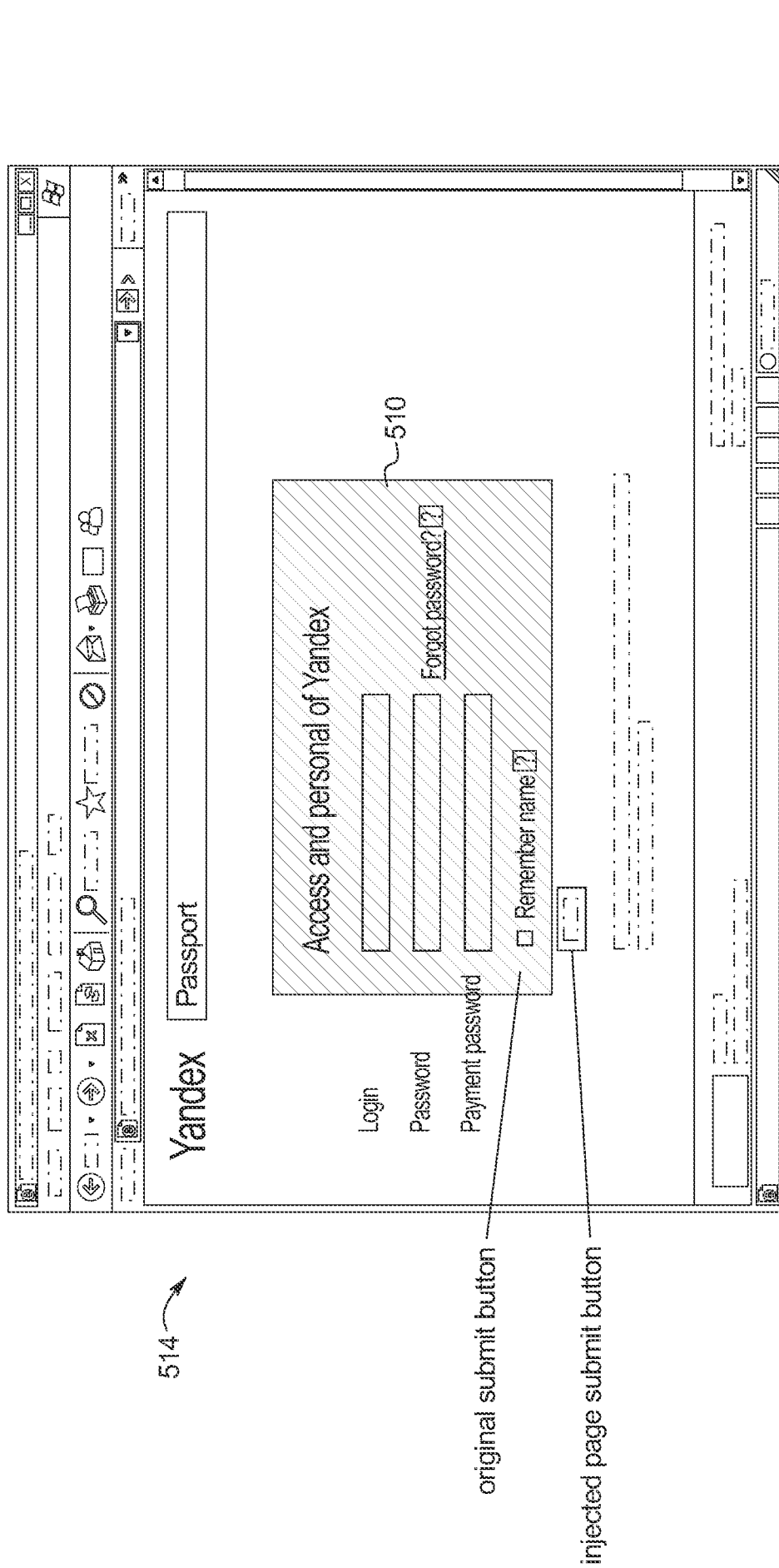


FIG. 8
SilentBanker: Modified Webpage



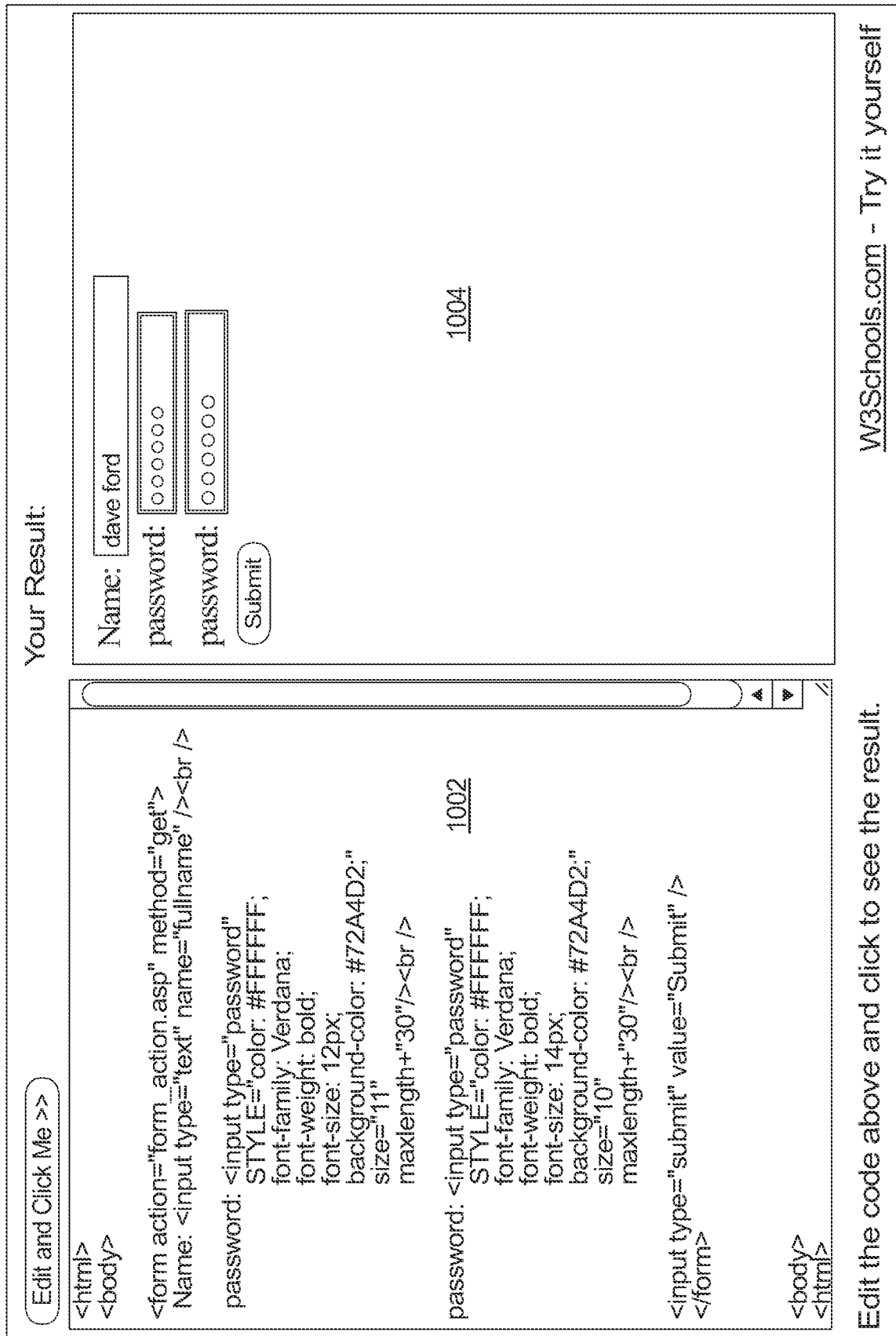
S21sec

FIG. 9
SilentBanker: Modified Webpage



S21sec

FIG. 10



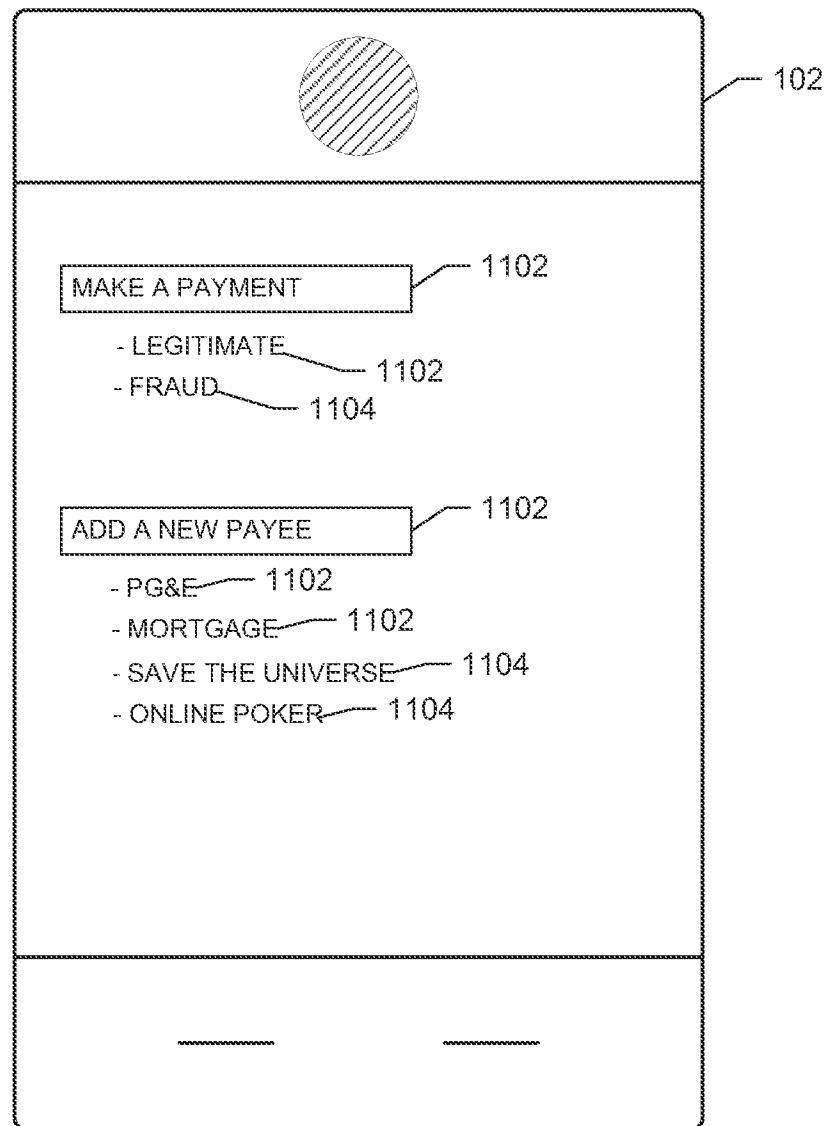


FIG. 11

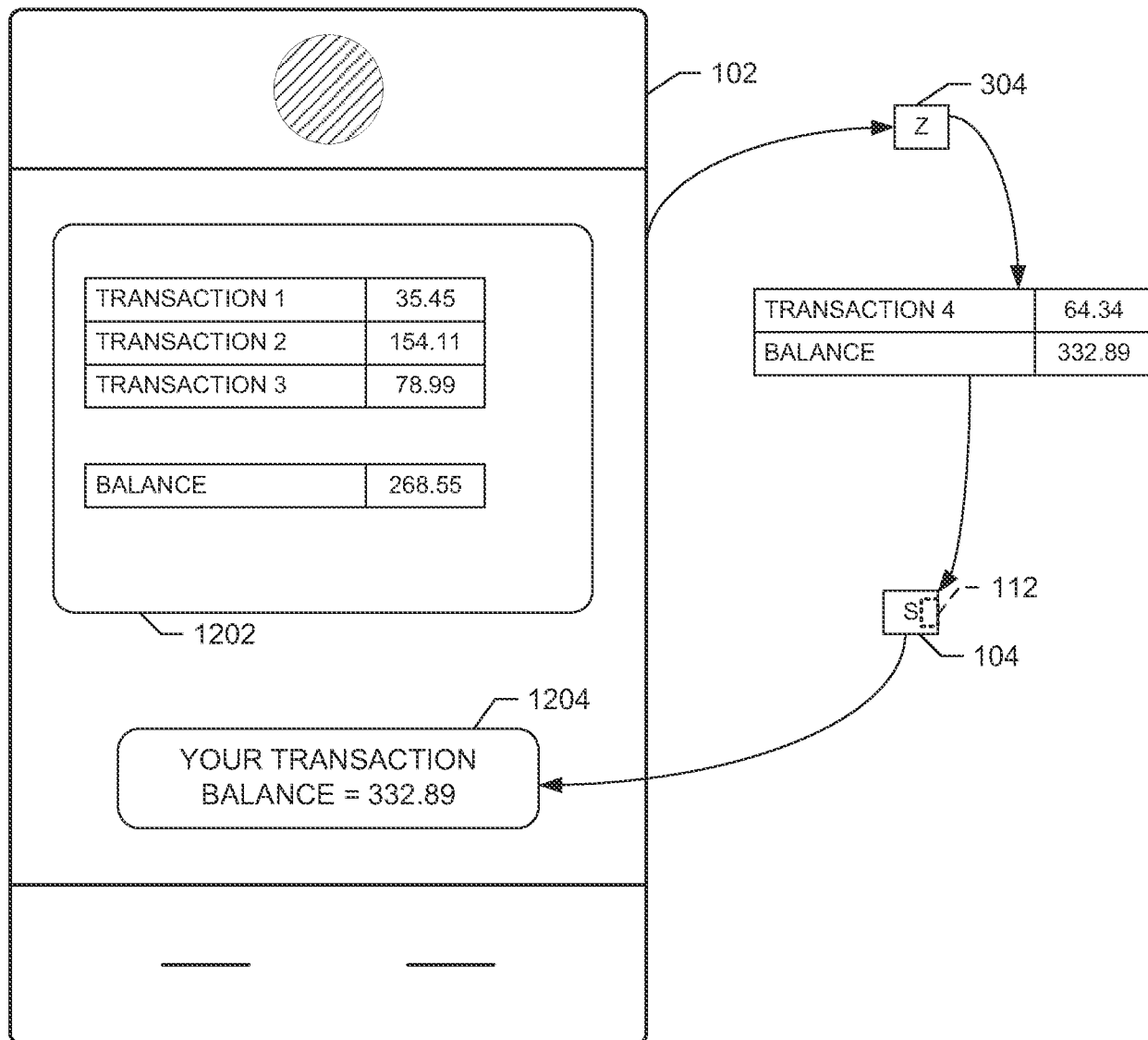


FIG. 12

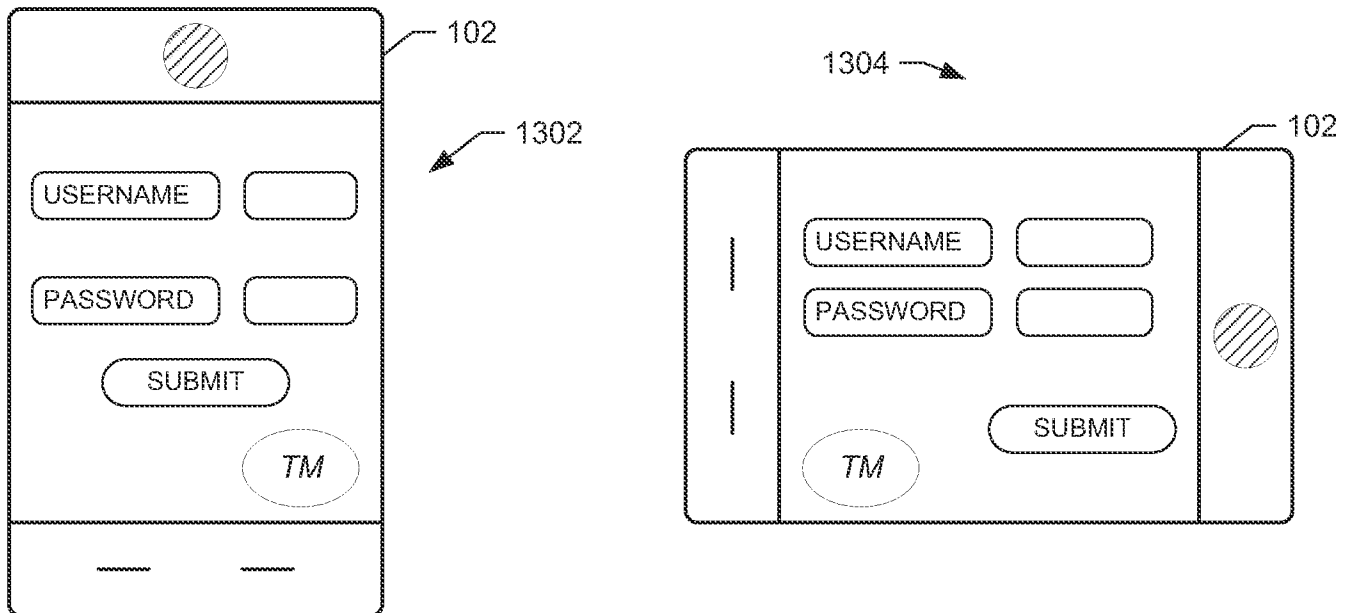


FIG. 13

1400

PREDICTION DATA STRUCTURE FOR SESSION WITH CLIENT DEVICE XYZ	
SOFT INFORMATION	
TEXT FIELD 1 – (32, 50), FONT 12, TIMES NEW ROMAN, BLACK	
TEXT FIELD 2 – (32, 70), FONT 14, ARIAL, BLACK	
TEXT FIELD 3 – (32, 90), FONT 12, ARIAL, BLUE, BOLD	
SUBMIT BUTTON – (50, 110), DEVIATION +/- 10	
PROGRAMMED TEXT – "E" EVERY THIRD CHARACTER, "Z" EVERY FIFTH CHARACTER	
BANNER LOCATION (10, 50)	
UN-RENDERED TEXT (32, 60), (32, 100)	
SECONDARY INFORMATION	
BROWSER TYPE – OPP2	
OPERATING SYSTEM – OX22	
JAVA SCRIPT - ENABLED	

FIG. 14

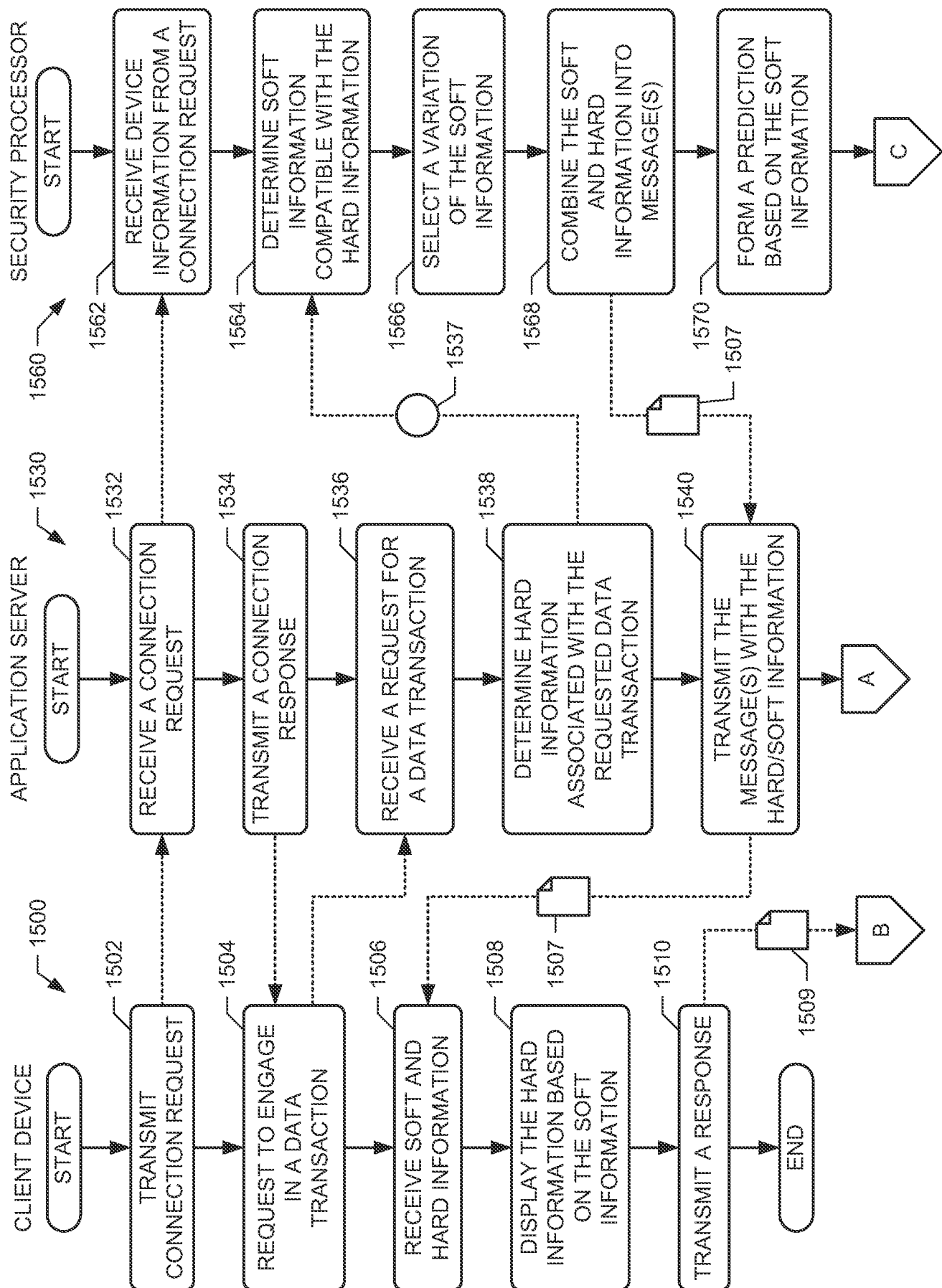
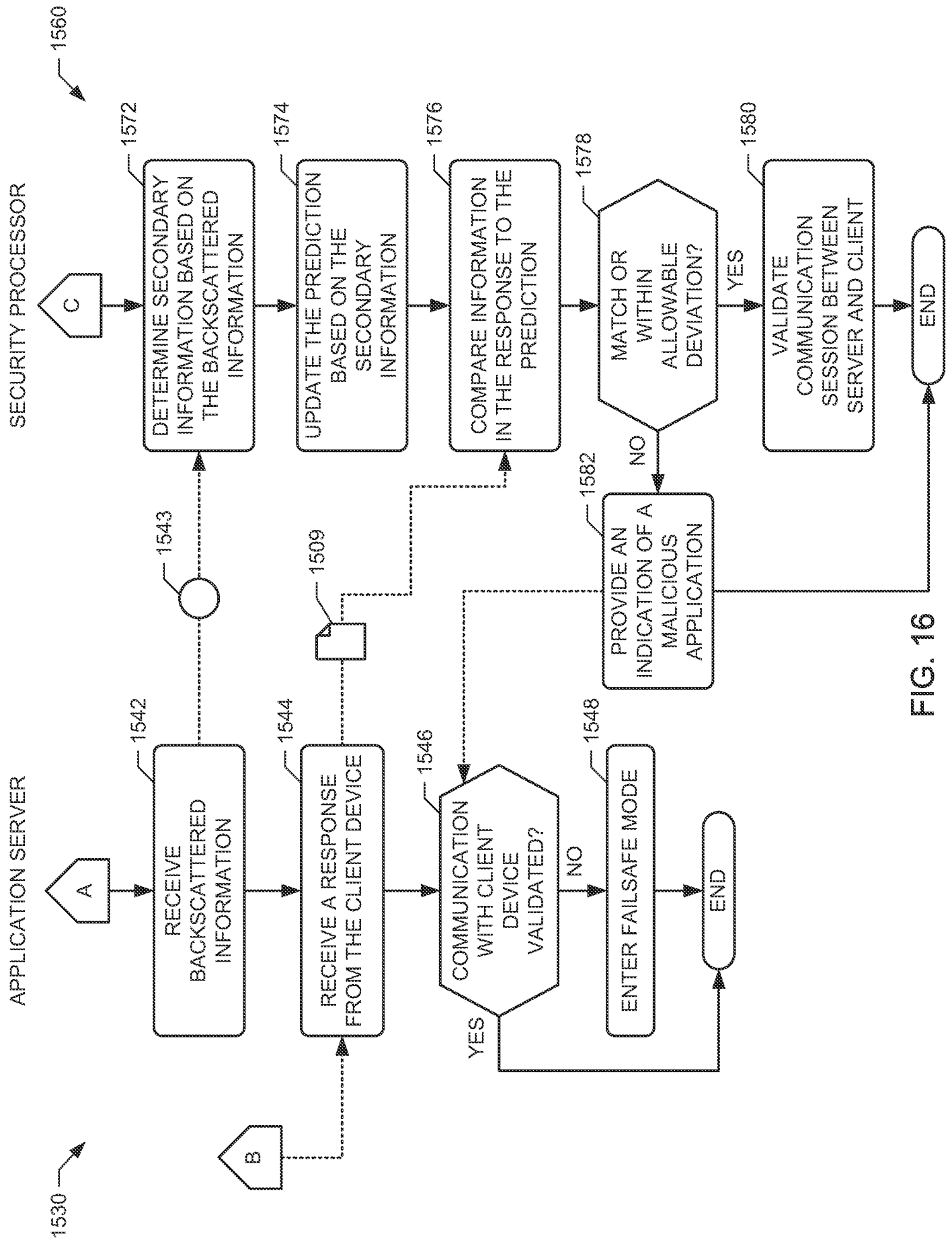
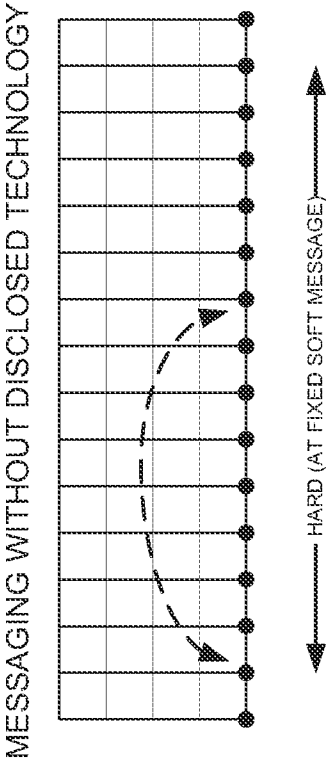


FIG. 15



1700



1710

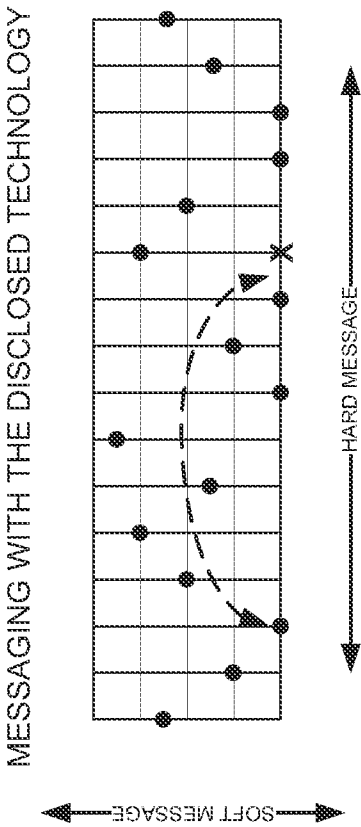


FIG. 17

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 12/56363

A. CLASSIFICATION OF SUBJECT MATTER IPC(8) - G06F 15/16 (2012.01) USPC - 709/203 According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) USPC: 709/203 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched USPC: 709/201-203, 217-219; 715/200-204, 700 (keyword limited - see terms below) Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) PatBase; GOOGLE; GoogleScholar; GooglePatents; PubWEST(PGPB,USPT,EPAB,JPAB) Search Terms: request, transmission, message, authentication, malicious, virus, backscatter, layer, prediction, time, temporal, library, style, font, textual, render, present, select, display, client, protocol		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2010/0194753 A1 (Robotham et al.) 05 August 2010 (05.08.2010), entire document, especially; abstract, para. [0002]-[0005], [0033], [0047], [0053]	1 - 34
A	US 2010/0011446 A1 (Klucher et al.) 14 January 2010 (14.01.2010), entire document, especially; abstract, para. [0028]-[0031], [0070], [0095]	1 - 34
A	US 2010/0080411 A1 (Deliyannis) 01 April 2010 (01.04.2010), entire document, especially; para [0020], [0025], [0035], [0039]	1 - 34
A	US 2010/0031139 A1 (Ihara) 04 February 2010 (04.02.2010), para [0049]-[0059], [0062]-[0063]	1 - 34
A	US 2010/0218253 A1 (Sutton et al.) 26 August 2010 (26.08.2010), entire document	1 - 34
A	US 6,178,510 B1 (O'Connor et al.) 23 January 2001 (23.01.2001), entire document	1 - 34
A	US 2009/0113528 A1 (Ananda et al.) 30 April 2009 (30.04.2009), entire document	1 - 34
A	US 2010/0333042 A1 (Yoshii) 30 December 2010 (30.12.2010), entire document	1 - 34
A	US 2010/0211893 A1 (Fanning et al.) 19 August 2010 (19.08.2010), entire document	1 - 34
A	US 2003/0195923 A1 (Bloch et al.) 16 October 2003 (16.10.2003), entire document	1 - 34
☒ Further documents are listed in the continuation of Box C. ☐		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 26 October 2012 (26.10.2012)		Date of mailing of the international search report 30 NOV 2012
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201		Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 12/56363

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2010/0100958 A1 (Jeremiah) 22 April 2010 (22.04.2010), entire document	1 - 34
A	US 2008/0172741 A1 (Reumann et al.) 17 July 2008 (17.07.2008), entire document	1 - 34
A	US 2005/0283833 A1 (Lalonde et al.) 22 December 2005 (22.12.2005), entire document	1 - 34
A	US 2002/0023145 A1 (Orr et al.) 21 February 2002 (21.02.2002), entire document	1 - 34
A	US 2008/0244715 A1 (Pedone) 02 October 2008 (02.10.2008), entire document	1 - 34