



(51) International Patent Classification:
G06K 19/06 (2006.01) **H04L 9/14** (2006.01)

(21) International Application Number:
PCT/SG2011/000064

(22) International Filing Date:
15 February 2011 (15.02.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/305,736 18 February 2010 (18.02.2010) US

(71) Applicant (for all designated States except US): **NATIONAL UNIVERSITY OF SINGAPORE** [SG/SG];
21 Lower Kent Ridge Road, Singapore 119077 (SG).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **CHANG, Ee Chien** [SG/SG]; c/o National University of Singapore, School of Computing, Department of Computer Science, 21 Lower Kent Ridge Road, Singapore 119077 (SG). **FANG, Chengfang** [CN/SG]; c/o National University of Singapore, School of Computing, Department of Computer Science, 21 Lower Kent Ridge Road, Singapore 119077 (SG).

(74) Agent: **AMICA LAW LLC**; 30 Raffles Place, #18-03/04 Chevron House, Singapore 048622 (SG).

(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, 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, PE, PG, PH, PL, PT, RO, RS, RU, 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, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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).

Declarations under Rule 4.17:

— of inventorship (Rule 4.17(iv))

Published:

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

(54) Title: SECURING INTERACTIVE SESSIONS USING BARCODES WITH VISUAL CUE

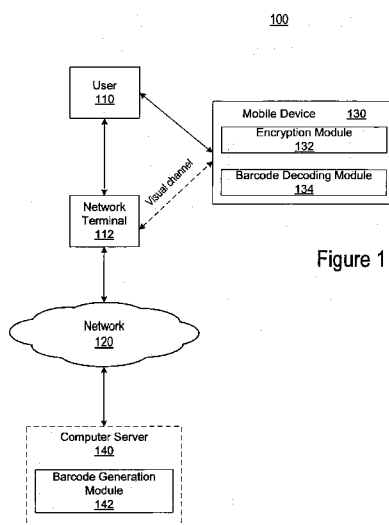


Figure 1

(57) Abstract: A computer system enables interactive sessions between a user and computer server using a mobile device and a visual channel. The computer system comprises a computer server configured to generate one or more two-dimensional barcodes to represent a message exchanged between the user and the server. The barcode has an associated visual cue to indicate where the barcode is displayed relative to other barcodes associated with the message. The computer server encrypts the original message and adds error correcting code to the encrypted message. The computer server embeds the visual cue in the encrypted message to generate the barcode. The user uses the mobile device to decode the barcode to extract the embedded message sent from the server.

Securing Interactive Sessions Using Barcodes with Visual Cue

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 61/305,736, filed on February 18, 2010, entitled “Seamless and Secure Interactions via Untrusted Kiosk Using Mobile Device’s Camera,” which is incorporated by reference in its entirety.

BACKGROUND

[0002] This invention relates to security of communications, and in particular to securing computer interactive sessions using a mobile device through a visual channel and visual inspection.

[0003] Securing communication between a user and a computer server through an untrusted network terminal (e.g., an Internet kiosk) is challenging even if the user may authenticate the connection using tools like one-time-password token, smart-card, or a mobile device (e.g., a mobile phone). Multiple communication messages representing information exchanged between the user and the server are relayed by the network terminal between the user and the server. The user may enter the messages to the server via the input device of the terminal directly, or through the input device of his/her mobile device, which transmits the messages to the terminal for transmission to the server. One of the hurdles is the difficulty in securely passing information from the terminal to the mobile device, and presenting the verified authentic information to the user in a user friendly manner.

[0004] Using a traditional communication channel to connect the mobile device and the terminal, like wireless connection or plug-and-play connection, is subject to various man-in-the-middle attacks. For example, some hackers may download spyware and catch user activity via keystroke logging. Other hackers stall hardware keystroke logging devices that capture user activity. Even if a secure channel can be established, the mobile device is traditionally not considered to help authenticating communication messages exchanged and rendered on the untrusted terminal's display.

[0005] Some existing methods utilize a digital camera in a mobile device to provide an alternative real-time communication channel between an untrusted network terminal and the mobile device. Through the alternative communication channel, communication messages are rendered on the display unit of the network terminal in a form of, such as two-dimensional (2D) barcode images, which are captured and decoded by the mobile device with its camera. Although such communication channel (also called "visual channel") can be eavesdropped by "over-the-shoulder" attacks, it is arguably impossible to modify or insert messages, and thus is secure against man-in-the-middle attack.

[0006] Visual channel has been exploited by some existing methods in verifying the session key exchanged over an unsecured channel, or in verifying the display shown in the display unit of an untrusted network terminal using a digital camera. However, the existing methods face challenges of effectively authenticating interactive sessions, where a user of the untrusted network terminal often sends and receives multiple subsequent messages through the visual channel. For example, during an interactive session, after a session key has been securely established between the server and the mobile device, there may be many subsequent communication messages between the server and the mobile device relayed by the terminal,

which require protection by the session key. These messages may need to be rendered over different webpages, or in a scrolling webpage where not all of them are visible at the same time.

[0007] An example of interactive session is communications between a user of an online banking application and a security server that provides sensitive banking transaction information. The user can browse and selectively view previous transactions, and carry out new transactions with the server. A typical screenshot of the interactive session contain important information like the user's account information, and less sensitive information like advertisements, help information, and navigation information, as shown in Figure 7.

[0008] One way to process the multiple communication messages after establishing the interactive session between the user and the server is to render the messages as 2D barcodes, each of which is protected by the same session key. To view the message embedded in a 2D barcode, the user moves his/her mobile device over the barcode, and the camera of the mobile device captures the barcode. The mobile device authenticates the captured barcode and displays the message embedded in the barcode on the display panel of the mobile device. However, as there are many barcodes associated with the same session key, it is possible for a dishonest network terminal to perform "rearrangement" attack, e.g., replaying barcodes or showing barcodes in the wrong order.

[0009] The "rearrangement" attack may arise due to the limitation that the camera of a mobile device is unable to capture the whole screen of the application webpage with sufficient precision. One may prevent the attack by requiring the user to scan all the bar-codes with his/her mobile device, and all the messages are authenticated and rendered by the mobile device. However, it is troublesome for the user to scan all the barcodes, and there are situations where the user only wants to view some, but not all, of the messages. In addition, user experience is

degrading when navigating and browsing the messages (e.g. a large table of transactions) within the relatively small display panel of his/her mobile device.

SUMMARY OF THE INVENTION

[0010] According to an embodiment of the invention, a computer-implemented method is provided for encoding an original message from a computer server to a user into one or more two-dimensional barcodes with one or more visual cues. In one embodiment, the method encrypts the original message with a message encrypting key and a message authentication key to generate an encrypted message. The method applies error correcting code to the encrypted message and determines a suitable set of visual cues. The method generates at least one two-dimensional barcode image using a visual cue embedding key. The two-dimensional barcode contains the encrypted message and visually resembles the visual cues. The barcodes, together with other messages (e.g., original message and other non-essential messages), are displayed to the user.

[0011] The user uses a mobile device as an inspection tool to extract messages from the barcodes, and/or verify the messages displayed alongside with the barcodes. The mobile device decodes the barcodes and displays the authenticated messages. The user visually checks the consistency of the visual cues, the authenticated messages and the displayed messages.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] Figure 1 shows a block diagram of a computer system for securing interactive sessions using a mobile device and visual inspection in accordance with an embodiment of the invention.

[0013] Figure 2 is a block diagram illustrating a typical computer acting as a network terminal or a computer server.

[0014] Figure 3 shows an example of encrypting a plain text into a cipher text by the mobile device.

[0015] Figure 4 is a flow diagram of generating a 2D barcode of a message by the computer server.

[0016] Figure 5A is an illustration of a visual cue represented in L-shape blocks of pixels.

[0017] Figure 5B is an illustration of a visual cue represented by a block of tiled L-shape blocks of pixels.

[0018] Figure 6 is an illustration of partial decoding of a message by the mobile device.

[0019] Figure 7 is an example screenshot of an online banking transaction webpage to be encoded by the computer server, where a mobile device can be dishonest in replaying communication messages.

[0020] Figure 8 is an illustration of the corresponding screenshot of the banking transaction webpage shown in Figure 7, which is encoded by the computer server using one embodiment of the invention.

[0021] Figure 9 is an illustration of the corresponding screenshot of the banking transaction webpage shown in Figure 7, which is encoded by the computer server using another embodiment of the invention.

[0022] The figures depict various embodiments of the invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative

embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0023] Referring to Figure 1 for purposes of explanation, embodiments of the invention provide a computer system 100 to secure interactive sessions using a mobile device 130 of a user 110. The user 110 communicates with a computer server 140 via a network 120, a network terminal 112 and/or the mobile device 130. For a message from the server 140 to the user 110, the server 140 generates a 2D barcode image of the message encoded with one or more visual cues and sends the 2D barcode image to the user 110 for verification.

[0024] In the embodiments of the invention, user 110 refers to a human being who uses his/her mobile device 130 and the network terminal 112 to communicate with the computer server 140. The network terminal 112 is a computer that has at least an input device, e.g., a keyboard, and a display unit. Figure 2 is a block diagram of a computer 200 for acting as a network terminal 112 and/or as a computer server 140. Illustrated are at least one processor 202 coupled to a chipset 204. Also coupled to the chipset 204 are a memory 206, a storage device 208, a keyboard 210, a graphics adapter 212, a pointing device 214, and a network adapter 216. A display 218 is coupled to the graphics adapter 212. In one embodiment, the functionality of the chipset 204 is provided by a memory controller hub 220 and an I/O controller hub 222. In another embodiment, the memory 206 is coupled directly to the processor 202 instead of the chipset 204.

[0025] The storage device 208 is a non-transitory computer-readable storage medium, such as a hard drive, compact disk read-only memory (CD-ROM), DVD, or a solid-state memory device and stores files. The memory 206 holds instructions and data used by the processor 202.

The pointing device 214 may be a mouse, track ball, or other type of pointing device, and is used in combination with the keyboard 210 to input data into the computer 200. The graphics adapter 212 displays images and other information on the display 218. The network adapter 216 couples the computer 200 to the network 120.

[0026] As is known in the art, a computer 200 can have different and/or other components than those shown in FIG. 2. In addition, the computer 200 can lack certain illustrated components. In one embodiment, a computer 200 acting as a computer server 140 can lack a keyboard 210, pointing device 214, graphics adapter 212, and/or display 218. Moreover, the storage device 208 can be local and/or remote from the computer 200 (such as embodied within a storage area network (SAN)).

[0027] As is known in the art, the computer 200 is adapted to execute computer program modules for providing functionality described herein. As used herein, the term “module” refers to computer program logic utilized to provide the specified functionality. Thus, a module can be implemented in hardware, firmware, and/or software. In one embodiment, program modules are stored on the storage device 208, loaded into the memory 206, and executed by the processor 202.

[0028] The network 120 enables communications between the user 110 and the computer server 140 through the network terminal 112. In one embodiment, the network 120 is the Internet, and uses standardized internetworking communications technologies and protocols (e.g., 3G mobile networking protocol and WiFi networking protocols), known now or subsequently developed that enable the user 110 to communicate with the computer server 140.

[0029] The mobile device 130 is a mobile computing device, e.g., a mobile phone, which has a digital camera, input device and a display unit and sufficient computing power. The mobile

device 130 has an encryption module 132 to encrypt a message entered through the input device of the mobile device 130 by the user 110. Any message encryption schemes available to those of ordinary skills in the art are within the scope of the invention. The user 110 enters the encrypted message (e.g., a cipher text of the message) through the input device of the network terminal 112, which replays the message from the user to the sever 140. Figure 3 shows an example of encrypting a user message (e.g., amount of money in plain text) into a cipher text.

[0030] The mobile device 130 also has a barcode decoding module 134 adapted to decode a 2D barcode image received from the server 140 via the terminal 112 to reveal message payload embedded in the barcode image (i.e., the message sent from the server 140) to the user 110. The barcode decoding module 134 is further discussed below with reference to the section of decoding barcode with visual cue.

[0031] The computer server 140 sends messages to the user 110. For each message sent from the server 140 to the user 110, the server 140 generates a 2D barcode image of the message, and encodes the 2D barcode with one or more visual cues. Generally, a visual cue is an object acting as a visible “signpost” associated with a barcode, where a user uses the visual cue to verify or navigate within a collection of barcodes. The computer server 140 sends the encoded barcode image to the user 110 for verification. The server 140 illustrated in Figure 1 has a barcode generation module 142 that encodes a message into a 2D barcode image with one or more visual cues. Other embodiments of the server 140 may include different and/or additional modules. The barcode generation module 142 is further described below with reference to Figure 4, Figures 5A-5B and the section of barcode encoding with visual cue.

SECURITY MODELS

[0032] The user 110, the mobile device 130, the network terminal 112 and the computer server 140 communicate with each other. In one embodiment, the mobile device 130 does not directly communicate with the server 140 for communication cost and security concerns. For example, there are situations where the connection (e.g., a 3G mobile network connection) between the server 140 and the mobile device 130 is not available due to cost or other constraints. A direct network connection between the mobile device 130 and the server 140 may also raise security concern due to potential man-in-the-middle attack, which is especially harmful for online transactions. In the embodiment illustrated in Figure 1, the mobile device 130 indirectly communicates with the server 140 through the network terminal 112. To differentiate from the public network 120, the communication channel between the network terminal 112 and the mobile device 130 is referred to as “visual channel.”

[0033] The visual channel between the network terminal 112 and the mobile device 130 employs multiple security models and communication protocols. In embodiment, three security models may be used to secure the communications between the user 110 and the server 140:

[0034] Model 1: the network terminal 112 is not trusted by the user 110. The mobile device 130 is trusted by the user 110. Model 1 is designed to protect both confidentiality and authenticity of the messages exchanged between the user 110 and the server 140;

[0035] Model 2: both the network terminal 112 and the mobile device 130 are not trusted by the user 110, but at least one of the terminal 112 or the mobile device 130 honestly relays the communications between the user 110 and the server 140. “Honestly” replaying a message by a party means that the party is not compromised and transmits the message as it is. Model 2 is designed to protect the authenticity of the messages exchanged between the user 110 and the server 140;

[0036] Model 3: both the network terminal 112 and the mobile device 130 are not trusted by the user 110, and both terminal 112 and the mobile device 130 may dishonestly relay the communications between the user 110 and the server 140. “Dishonestly” replaying a message by a party means that the party is compromised and may modify the message. Model 3 is designed to protect the authenticity of the messages exchanged between the user 110 and the server 140.

[0037] The third model is motivated by scenarios where the terminal 112 and the mobile device 130 are compromised, but independently by two different malicious parties, who do not communicate directly with each other. For instance, a dishonest mobile device 130 always says “authentic” for whatever authentication it is supposed to carry out, or a dishonest network terminal 112 remotely controlled by a malicious party deceives the user to accept a particular message. To detect such dishonest mobile device 110, the model 3 may require the mobile device 130 to extract and produce a human readable proof from an authentication tag associated with a message. A corresponding proof is also shown in the terminal’s display and hence the user 110 can visually verify whether the proofs are consistent.

[0038] In addition to security requirements, the computer system 100 considers user experience in designing the security models. In one embodiment, the computer server 140 uses augmented reality in generating 2D barcode images and sub-region authentication to provide better user experience. For example, given a message of m sent from the server 140 to the user 110 and a visual cue v , the computer server 140 generates a 2D barcode image that not only carries m as its message payload, but also is visually displayed at the position indicated by the visual cue. The message sent from the server 140 to the user 110 is embedded in the 2D barcode image, and contains important information, e.g., user’s bank account information. A 2D barcode has at least one visual cue, e.g., a numeric number indicating the position of the 2D barcode in a

group of related 2D barcodes. 2D barcode with visual cue generations is further described below.

COMMUNICATION PROTOCOLS

[0039] The user 110 communicates with the server 140 through the network terminal 112 and/or the mobile device 130 by exchanging one or more messages using one or more communication protocols. Assuming that the server 140 has already established a long term shared key with the mobile device 130 when the user 110 registers an account with the server 140. In addition, for model 2 and model 3 described above, it is further assumed that the user 110 has established a password with the server 140 which is secret to the mobile device 130. Before each interactive session, the server 140 authenticates the user 110 and the mobile device 130 using a session key k_s , which is to be kept secret from the network terminal 112. A secure key exchange can be derived from any session key exchange schemes known to those of ordinary skills in the art.

[0040] There are two types of communication protocols: the communication protocols for sending messages by the server 140 to the user 110 and the communication protocols for sending messages by the user 110 to the server 140. Table I provides a summary of notations used in the description of the communication protocols.

Notation	Description
m_U	The message from user 110 to server 140
m_S	The message from server 140 to user 110
k_T	The key for message authentication
k_E	The key for encoding visual cue
k_V	The key for embedding visual cue

k_s	The session key where $k_s = (k_T, k_E, k_V)$
v	A visual cue symbol
$B(k_s, m, v)$	A barcode image encoding a message m and visual cue v under key k_s
$T_{k_T}(m)$	An authentication tag of a message m under key k_T
$E_{k_E}(m)$	An encryption of a message m under key k_E
$ECC(m)$	An error correcting encoding of a message m
$A \rightarrow B : m$	The entity A sends a message m to another entity B
$A \xrightarrow{C} B : m$	The entity A sends a message m to another entity B using C as a replay point

[0041] The first type of the communication protocols is for the server 140 to send a message m_s from the server 140 to the user 110. The server 140 uses two methods, MS1 and MS2, for the message transmission. The following operations are performed to send a message m_s to the user 110 using MS1: (1) the server 140 generates a barcode image $B(k_s, m_s, v)$ and sends the barcode to the network terminal 112, where k_s is the established session key, and v is the appropriate visual cue associated with the barcode; (2) the terminal 112 displays the barcode to the user 110; (3) the user 110 inspects and verifies whether the visual cue is valid; (4) the mobile device 130 captures the barcode; and (5) if the mobile device 130 successfully verifies the payload m_s embedded in the barcode, it displays m_s to the user 110. If the mobile device 130 fails to verify m_s , it displays an error message.

[0042] The following is an example pseudo code of the message transmission using method MS1 described above:

1. Server $\rightarrow$ Terminal: $B(k_s, m_s, v)$;

2. Terminal $\rightarrow$ User: v ;
3. User verifies the visual cue v ;
4. Terminal $\rightarrow$ Mobile: $B(k_s, m_s, v)$;
5. Mobile $\rightarrow$ User: m_s if m_s is authentic; error message otherwise.

[0043] In method 2, MS2, the message m_s is displayed by both the terminal 112 and the mobile device 130 such that the user 110 is able to detect if one of the terminal 112 or the mobile device 130 is dishonest. The following operations are performed using MS2 for the message transmission: (1) the server 140 generates a barcode image $B(k_s, m_s, v)$, and sends both the barcode image and the message m_s to the terminal 112; (2) the terminal 112 displays the barcode side-by-side with the message m_s ; (3) the user 110 inspects and verifies the visual cue; (4) the mobile device 130 captures the barcode and rejects the barcode if the barcode is not authentic, and displays m_s otherwise; (5) the user reads m_s from the mobile's 130 display panel and the terminal's 112 display; and (6) the user 110 accepts m_s if the m_s in step (2) is consistent with m_s in step (4).

[0044] The following is an example pseudo code of the message transmission using method MS2 described above:

1. Server $\rightarrow$ Terminal: $B(k_s, m_s, v), m_s$;
2. Terminal $\rightarrow$ User: v, m_{s1} ;
3. User verifies the visual cue v ;
4. Terminal $\rightarrow$ Mobile: $B(k_s, m_s, v)$;
5. Mobile $\rightarrow$ User: m_{s2}
6. User accepts m_{s1} if $m_{s1} = m_{s2}$

[0045] The second type of the communication protocols is for the user 110 to send a message m_U to the server 140. There are two methods, MU1 and MU2, for the message transmission, where MU1 protects both confidentiality and authenticity of the message m_U , whereas method MU2 protects only the authenticity of the message m_U , but with less user action.

[0046] The following operations are performed to send a message m_U from the user 110 to the server 140 using MU1: (1) the user 110 enters the message m_U through the mobile device 130; (2) the mobile device 130 computes and shows the user 110 an encrypted form $E_{k_E}(m_U) \parallel T_{k_T}(E_{k_E}(m_U))$ in readable characters (e.g., uuencode) of the message m_U , where k_E is the key for encryption, k_T is the key for message authentication, and $T_{k_T}(E_{k_E}(m_U))$ is an authentication tag of the encrypted message m_U under key k_T ; (3) the user 110 types the encrypted message to the network terminal 112, which sends the user typed message to the server 140; (4) the server 140 accepts the message m_U if the authentication tag $T_{k_T}(E_{k_E}(m_U))$ of the message is valid.

[0047] The following is an example pseudo code of the message transmission using method MU1 described above:

1. User $\rightarrow$ Mobile: m_U ;
2. Mobile $\rightarrow$ User: $E_{k_E}(m_U) \parallel T_{k_T}(E_{k_E}(m_U))$;
3. User $\xrightarrow{\text{Terminal}}$ Server: $E_{k_E}(m_U) \parallel T_{k_T}(E_{k_E}(m_U))$;
4. Server accepts m_U if the authentication tag $T_{k_T}(E_{k_E}(m_U))$ of the message is valid.

[0048] In method 2, MU2, the confidentiality of message m_U is not required. The following operations are performed using MU2 for the message transmission: (1) the user 110 enters m_U

through the terminal's 112 input device, and the terminal 112 forwards m_U to the server 140; (2) the server 140 generates a barcode $B(k_s, m_U \parallel c, v)$, where c is a randomly generated nonce (e.g., an arbitrary number generated for unique identification), and sends the barcode to the terminal 112; (3) the terminal 112 displays the barcode, and the user 110 visually verifies whether the visual cue v is correct; (4) the mobile device 130 captures the barcode and rejects the barcode if the barcode is not authentic; (5) the mobile device 130 renders the message m_U and the nonce c on its display; (6) if m_U is consistent with the message the user 110 entered in step (1), the user 110 enters c to the terminal 112, and the terminal 112 forwards c to the server 140; and (7) the server 140 rejects nonce c if the nonce c is wrong.

[0049] The following is an example pseudo code of the message transmission using method MU2 described above:

1. User $\xrightarrow{\text{Terminal}}$ Server : m_U ;
2. Server $\rightarrow$ Terminal: $B(k_s, m_U \parallel c, v)$;
3. Terminal $\rightarrow$ User: v ;
4. Terminal $\rightarrow$ Mobile: $B(k_s, m_U \parallel c, v)$;
5. Mobile $\rightarrow$ User: m_U, c ;
6. User $\xrightarrow{\text{Terminal}}$ Server : c ;
7. Server accepts m_U if the nonce c is consistent, or rejects the nonce c otherwise.

ANALYSIS OF THE MODELS WITH THE PROTOCOLS

[0050] In Model 1, the computer system 100 uses method MU1 and method MS1 to achieve confidentiality and authenticity of the visual channel. Specifically, the computer system 100 uses method MU1 for sending messages from the user 110 to the server 140 and uses method MS1 for the server 140 to send messages to the user 110. For both methods MS1 and MU1, the

terminal 112 plays the role of a relay point for passing messages, and thus a malicious terminal 112 can be the man-in-the-middle for compromising the messages. In this setting, the two end points (the server 140 and the mobile device 130) use a shared key to communicate over an untrusted public channel (e.g., the network 120). The cryptographic technique (e.g., encryption and message authentication code) can be used to secure the messages by providing both confidentiality and authenticity of the messages. It is noted that methods MU2 and MS2 are not used in Model 1 because the messages are sent through the potentially dishonest terminal 112 and the confidentiality of the messages are not protected.

[0051] In Model 2, the computer system 100 uses method MU2 to send messages from the user 110 to the server 140, and uses the method MS2 for the server 140 to send messages to the user 110. Model 2 is designed to protect the authenticity of the messages. Assuming the terminal 112 is dishonest (i.e., likely to compromise the messages), in both directions of the communication, the barcode of a message can be treated as the message authentication code (MAC) of the message. Since the terminal 112 does not have the key used in generating the barcode, the authenticity of the message inherits from the MAC used in the barcode generation.

[0052] In the case where the mobile device 130 is dishonest, but the terminal 112 is honest, using method MU2, the terminal 112 forwards m_U to the server 140 as it is. It is impossible for the mobile device 130 to modify m_U without being noticed by the server 140. Similarly, in method MS2, since the actual message m_U is displayed by the honest terminal 112, the user 110 can compare the displayed message by both the terminal 112 and the mobile device 130 and detects any modification to the message. It is noted that methods MU1 and MS1 are not used in Model 2 because if the mobile device 130 is dishonest and changes the message, the user 110 or the server 140 cannot verify the modification.

[0053] Similar to Model 2, Model 3 using methods MU2 and MS2 provides the authenticity of the messages exchanged between the user 110 and the server 140. For example, in order to trick the server 140 to accept a compromised message, the terminal 112 needs to send the server 140 the compromised message and to obtain a barcode containing the compromised message and a verification code (e.g., a nonce). The server 140 accepts a compromised message only if the verification code is presented. Since the verification code is randomly chosen, the terminal 112 is unlikely to succeed in guessing correctly the verification code. Therefore, the terminal 112 needs to get the verification code from the user 110. Without any hint from the terminal 112, the mobile device 130 is not able to display the message that the user 110 is expecting and in turn, the user 110 will not enter the verification code expected by the terminal 112.

[0054] Using method MS2, the dishonest terminal 112 tries to trick the user 110 into accepting a compromised message. The terminal 112 displays the compromised message side-by-side with the barcode associated with the corresponding uncompromised message. As the terminal 112 does not know the session key used to generate the barcode, the terminal 112 is unable to forge the barcode. In case of a dishonest mobile device 130, the mobile device 130 is unable to display the compromised message to trick the user 110 for acceptance because there is no direct communication from the terminal 112 to the mobile device 130.

BARCODE GENERATION WITH VISUAL CUE

[0055] Turning now to Figure 4, Figure 4 is a flow chart of the server 140 generating a 2D barcode with a visual cue of a message from the server 140 to the user 110. Specifically, given a message m from the server 140 to the user 110 and a secret key $k_s = (k_T, k_E, k_V)$, the server 140 generates a 2D barcode $B(k_s, m, v)$ such that the visual cue v is clearly visible and the message m can be extracted under noise and any modification on message m and visual cue v are detected

with high probability. In one embodiment, the server 140 comprises a barcode generation module 142 configured to perform the operations described with the reference to Figure 4 below.

[0056] Initially, the server 140 encrypts 402 a message m_s using encryption key k_E and message authentication key k_T and generates a corresponding encrypted message $m_0 = E_{k_E}(m_s) \parallel T_{k_T}(E_{k_E}(m_s))$ of the message m_s . The server 140 applies 404 error correcting code to the encrypted message m_0 and generates a message $m_1 = \text{ECC}(m_0)$ with the error correcting code. The server 140 embeds 406 a visual cue v represented as a 2D array of bits in the message m_1 to generate a larger 2D array of bits I , which visually appears as v . The server 140 adds 408 any control information (e.g., control points around the image I for image registration purpose). The server 140 generates 410 a 2D barcode image represented by the image I and sends 412 the barcode image to the user 110.

[0057] In one embodiment, the message from the server 140 to the user 110 is a collection of fixed-length strings organized in some data structure, e.g., a 2D-array where each entry of the array can be represented in a certain number of bits. Let the message be an m by n array A , where each entry $a_{i,j}$ of the array is a short fix-length string that can be represented in a certain number of bits. Let S be an m by n array of binary values, where $s_{i,j} = 1$ if and only if $a_{i,j}$ is an entry representing sensitive data of the message. Let K be the session key shared by the server 140 and the trusted mobile device 130. For each index (i, j) , the server 140 derives a sub-key $k_{i,j}$. The server 140 also determines a visual cue $l_{i,j}$, which is a symbol from some alphabet (e.g., a set of numeric numbers) to be associated with index (i, j) . Let L be the array of visual cues. The array L is determined based on the size of message array A and the type of

information to be protected. Array L does not depend on the session key and values in the message array A .

[0058] Let D be a region in a screenshot rendered by the server 140 for message array A .

The region is divided into $m \times n$ equal-sized blocks, and let $d_{i,j}$ be the block in the i -th column and j -th row. Each block can reliably embed a certain number of bits of message data. The data embedded in a block is the payload of the block. For each $(i, j) \in S$, given the visual cue $l_{i,j}$, message $a_{i,j}$ and the sub-key $k_{i,j}$, the server 140 computes the block $d_{i,j}$ as

$$d_{i,j} = \text{Encode}_{k_{i,j}}(l_{i,j}, a_{i,j}), \text{ and } D = \text{Encode}_K(L, A).$$

[0059] To encode a block, the server 140 generates a 2D barcode image such that (1) the message can be reliably decoded under noise, (2) the confidentiality of the message is protected, (3) the integrity of the message and visual cue is protected, and (4) the visual cue is clearly visible. To protect the confidentiality of the message, the server 140 encrypts the message using the following equation:

$$\begin{aligned} t &= \text{MAC}_{k_{i,j}}(E_{k_{i,j}}(a_{i,j}), l_{i,j}) \\ m_0 &= (l_{i,j}, E_{k_{i,j}}(a_{i,j}), t) \end{aligned} \quad (1)$$

where $E_{k_{i,j}}(\cdot)$ is a symmetric encryption scheme, and the $\text{MAC}_{k_{i,j}}(\cdot)$ is a message authentication tag with $k_{i,j}$ as the key. The t is an authentication tag protecting the cipher text of the message and the visual cue.

[0060] To add error correcting code to an encrypted message, in one embodiment, the server 140 uses oversampling technique to reduce the noise due to image pixels misalignment and other noise-causing factors. For example, a camera with resolution of 640 x 480 pixels in a typical

hardware configuration covers about a 300 x250 pixels sub-image of a screenshot. That is, roughly a 2 x 2 sensor is for one pixel in display. By treating 2 x 2 pixels as one single super-pixel, the server 140 can use 4 x4 sensors for one super-pixel. Such oversampling helps to reduce noise and mitigate other encoding artifacts. Each super-pixel can encode 1 bit data. If the message is “0”, all four pixels in the super-pixel are black. If the message is “1”, all four pixels in the super-pixel are white. To correct error, the server 140, in one embodiment, applies a (63, 36, 11) BCH error correcting code on the payload m_0 computed by Equation (1) to generate message m_1 below:

$$m_1 = BCH(m_0). \quad (2)$$

[0061] In embodiment, the server 140 uses fragile image watermarking technique to encode a visual cue associated with a message payload. The visual cue is acting as a host image and the message payload is embedded as the watermark of the host image. Specifically, to encode a visual cue $l_{i,j}$ associated with the message m_1 using key $k_{i,j}$, the server 140 constructs a block $d_{i,j}$ such that the visual cue $l_{i,j}$ is visible to user 110. Given a n-bit message m_1 , the message can be arranged as a x by y binary image, where $n = x \times y$ and x is even. Assume that the visual cue $l_{i,j}$ associated with the message m_1 is a $\frac{x}{2}$ by y pixels binary image, where each pixel is either 0 (representing a black pixel) or 1 (representing a white pixel). Every two pixels in m_1 are associated with one pixel of the visual cue $l_{i,j}$, and together the message m_1 and the visual cue $l_{i,j}$ can be represented with three black-and-white pixels in the generated barcode. The three pixels can be arranged in a “L”-shape as shown in Figure 5A and the three pixels forms a L-block. The eight combinations (i.e., 2^3) of values in a L-block is divided into two groups:

W and B . The L-blocks in W group have more pixels of value of 1 and thus, the L-blocks appear as “white.” Conversely, the L-blocks in B group have more pixels with value of 0 and appear as “black.”

[0062] Given a binary value $v_1 \in \{0,1\}$ of a pixel of the visual cue image, the server 140 encodes two bits $\langle b_1, b_2 \rangle$ into a three pixels L-block such that the brightness of the L-block can be adjusted according to v_1 . For instance, if $v_1 = 1$, the server 140 only outputs encoding elements in W group. Since there are four elements in W group, it is possible to encode the two bits b_1 and b_2 . Beside for the value of v_1 , there is no further constraint on how to encode $\langle b_1, b_2 \rangle$ into one of the four elements in W group. In order to prevent a malicious party from modifying the appearance of the visual cue, the mapping from the two bits $\langle b_1, b_2 \rangle$ to the three pixels of the associated L-block has to be kept secret. The key space for encoding a bit pair is $4! \times 4! = 576$.

[0063] The server 140 extends the two bits encoding to the array of message m_1 with a size of $x \times y$ bits and the visual cue $l_{i,j}$. The output image of the array of message m_1 consists of

$\frac{3 \times x}{2} \times y$ pixels. The server 140 first generates a $\frac{x}{2} \times y$ binary image from the visual cue $l_{i,j}$.

From the key $k_{i,j}$, the server 140 generates a pseudo-random sequence of sub-keys, each of

which is within the $[1,576]$ key space. The server 140 divides the message m_1 into $\frac{x}{2} \times y$ pair of bits, and encodes each pair of bits and the corresponding visual cue to a L-block. The server 140 tiles the L-blocks to construct an output image. Figure 5B is an example output image of a block constructed by the server 140.

DECODING BARCODE WITH VISUAL CUE

[0064] Given a block (represented by a 2D barcode) encoded by the server 140, the mobile device 130 decodes the block and displays the message embedded in the block to the user 110 for verification. In one embodiment, the barcode decoding module 134 of the mobile device 130 is configured to decode blocks encoded by the server 140. Specifically, given a block d and a key $k_{i,j}$, the decoding module 134 extracts the visual cue $l_{i,j}$ and message payload $a_{i,j}$ if d is close to $d_{i,j}$ as the following:

$$\text{Decode}_{k_{i,j}}(d) = (l_{i,j}, a_{i,j}), \quad \text{if } \text{dist}(d, d_{i,j}) < \varepsilon \quad (3)$$

where $\text{dist}(\cdot, \cdot)$ is a distance function and ε is a pre-determined error threshold. For a random block that has an error threshold higher than ε , the decoding module 134 rejects the block with high probability. The robustness to noise during decoding is achieved through error correcting during encoding process. For example, during encoding process, the server 140 embeds the payload of a block with an authentication tag computed over the visual cue and the cipher text of the message to be sent to the user 110. For a random block that cannot be corrected during decoding process, the block is most likely comprised during the transmission to the mobile device 130.

[0065] It is desirable to have small blocks so that as the user 110 pans the mobile device 130 over a screenshot, the transition of a view to another is smoother. However, if the block size is too small, more blocks are needed to represent the screenshot and the user 110 has to verify more visual cues associated with the small blocks. In one embodiment, the decoding module 134 partially decode a block while still provides accurate barcode verification.

[0066] To allow partial decoding of a block by the decoding module 134, the server 140 modifies the encoding scheme described by Equation (1). Specifically, let a be the message to be encoded for a block (i, j) and let a be presented by sub-blocks $\langle a_1, a_2, a_3 \rangle$. The block (i, j) contains two overlapping sensitive data regions h_{12} and h_{23} . The server 140 generates a verification tag for each of the sensitive data regions h_{12} and h_{23} as follows:

$$\begin{aligned} t_{h_{12}} &= MAC_{k_{i,j,1}}(E_{k_{i,j,1}}(a_1, a_2), l_{i,j}) \\ t_{h_{23}} &= MAC_{k_{i,j,2}}(E_{k_{i,j,2}}(a_2, a_3), l_{i,j}) \end{aligned} \quad (4)$$

where $k_{i,j,1}$ and $k_{i,j,2}$ are two keys generated from key $k_{i,j}$.

[0067] Figure 6 is an example of partial decoding of a block. The block 610 is divided into three sub-blocks, e_1 , e_2 , and e_3 . The block 610 has two regions, h_{12} and h_{23} , containing sensitive data, and the two regions overlap at sub-block e_2 . Instead of decoding three visual cues associated with each sub-block, the visual cue of the block 610 lies in the overlapping sub-block e_2 . As long as the user 110 uses the mobile device 130 to capture either one of the two sensitive data regions h_{12} and h_{23} , the decoding module 134 will successfully decode the whole block 610 by partially decoding the two sensitive data regions h_{12} and h_{23} .

DETECTING MODIFICATIONS TO BARCODE

[0068] A malicious network terminal 112 tends to modify various parts of a 2D barcode generated by the server 140 so that the user 110 is enticed to release sensitive personal data (e.g., banking account number) to the terminal 112. In one scenario, the user 110 is deceived to accept compromised messages decoded by the mobile device 130, and the messages are compromised due to the modification of control points of the barcodes. The mobile device 130 uses control

points on and around a block to decide the location/index of the block. Because the arrangement of the control points are public information, a malicious network terminal 112 is able to modify the control points of a block so as to deceive the mobile device 130 to associate the block with another location.

[0069] Control points of a block are a set of 2D points generated during geometric transformation of an image (e.g., a 2D barcode image) captured by the camera of the mobile device 130 to a common coordinate system. Image acquisitions by the camera of the mobile device 130 are typically noisy and subject to geometric distortion. Hence, image registration is required to transform a few acquired images to the common coordinate system. In one embodiment, each control point is depicted as a 2x2 red dot or a 4x4 red dot, the image captured by the camera of the mobile device 130 is a view of a screenshot and the common coordinate system is the screenshot coordinate system. Red color is chosen so that the barcodes with the control points are distinct from other content of the screenshot taken by the mobile device 130. Other colors can also be used. Other embodiments may use easily identifiable shape (instead of color) to represent control points.

[0070] Given an image of a screenshot captured by the camera of the mobile device 130 and a set of control points, the server 140 finds a geometric transformation that maps the image to its original screenshot coordinate. In one embodiment, the server 140 detects control points (e.g., red points) in the captured image, and finds an affine transformation such that the matching score of a transformed control point to its original control point in the screenshot coordinate is optimal. The matching score can be measured by the Euclidean distance between the transformed control point and its original control point. The transformation, in one embodiment, is a linear

transformation and can be performed first on the larger control points (e.g., 4x4 pixels) and refined on the smaller control points (e.g., 2x2 pixels).

[0071] When a malicious network terminal 112 tries to modify the control points, such modification can be detected by the mobile device 130 with high probability because each individual sensitive block of an image captured by the camera of the mobile device 130 is protected by an authentication tag (e.g., block index). To verify the authentication tag, a sub-key generated from the block index is required. If the malicious network terminal 112 feeds a different index, which is very likely to be different from the correct index, the authentication tag will be rejected by the mobile device 130 with high probability.

[0072] In another case, a malicious network terminal 112 tries to carry out a rearrangement attack by swapping two blocks, d_i and d_j with visual cues l_i and l_j , respectively. A straightforward rearrangement attack is to swap the two blocks, which leads to an invalid visual cues arrangement. A vigilant user who visually inspects the blocks will detect the swap.

[0073] Alternatively, the arrangement attack may make the swap and modify some L-shape blocks such that the visual cue on d_i is l_j and visual cue on d_j is l_i . From visual inspection, a user may not detect the swap because any modification of an L-shape block's brightness has a $\frac{1}{4}$ chance of being detected. Suppose at least β number L-shape blocks have to be modified in

order to deceive the user, the chance for the modification to be detected is $\left(\frac{1}{4}\right)^\beta$.

[0074] In one embodiment, the server 140 uses numeric symbols (e.g., 1, 2, 3, ... n) as visual cues, where the distance between any two numeric symbols is at least 8 L-shape blocks. Thus, to change the visual cues in two blocks requires at least 16 L-shape blocks to be modified. Other

embodiments may use different distance between any two numeric symbols. Additionally, the server 140 may use auxiliary location cues of a block (e.g., header of a row or column of a block) to help user correctly identify the position of the block.

VISUAL CUES FOR VERIFICATION OF MULTIPLE BARCODES

[0075] Verification of a view of screenshot rendered by the server 140 is conducted jointly by the user 110 and the mobile device 130. The server 140 encodes each block of the screenshot as a 2D barcode image protected by an authentication tag. If the mobile device 130 fails to verify the authentication tag, the mobile device 130 informs the user 110 to reject the screenshot. The responsibility of the user 110 is to visually inspect the visual cues associated with the 2D barcodes and verify whether the visual cues are in correct arrangement. The user 110 rejects the screenshot if the visual cues arrangement does not follow one or more predefined rules.

[0076] In one embodiment, a set of predefined rules to render 2D barcodes and their associated visual cues in a table structure is defined as following:

R1: every block has a unique numeric visual cue;

R2: the value of the visual clue on the top row, leftmost block is 1. The value increments by 1 from left to right. At the end of a row, the increments process continues to the leftmost block of the row below;

R3: the rightmost block in each row has an additional visual cue (e.g., a black dot) to indicate the end of row;

R4: the rightmost block in the last row has an additional visual cue (e.g., a black rectangle) to indicate the last block.

[0077] Alternatively, the server 140 may render multiple 2D barcodes in a linear sequence. For example, a sequence of 2D barcodes appears in the order $B_1, B_2, B_3, \dots, B_n$, where the order of the blocks gives an implicit structure of the encoded message contained in the sequence of the barcodes. In this case, the visual cue for a block in the sequence can be a counter indicating the position of the block in the sequence. For example, for a block B_i , its visual cue is i indicating the block B_i appears in i -th position of the sequence. To indicate the end of the sequence, the server 140 may use an additional special symbol (e.g., a dot) in the last block of the sequence.

APPLICATIONS OF SECURING INTERACTIVE SESSIONS

[0078] Figures 7-9 are examples of securing interactive sessions between the user 110 and the server 140 using the methods and communication protocols described above. Figure 7 shows an example screenshot of an online banking transaction webpage, which contains sensitive information like the user's account information 710 and less sensitive information (e.g., advertisements 720, help information 730 and web navigation information 740).

[0079] Figure 8 is the corresponding screenshot of the banking transaction webpage shown in Figure 7, which is encoded by the server 140 using the model 1 (where the mobile device 130 is trusted by the user 130). As illustrated in Figure 8, the sensitive user account information is treated as a long message to be encoded by the server 140. The sensitive user account information is replaced by eight 2D barcode images 810, each of which has a visible numeric number as the visual cue. The server 140 sends the barcodes to the network terminal 112, which displays the barcodes for the user 110 to verify. The numeric number associated with a barcode indicates the position of the barcode in the display. For example, the leftmost barcode on the first row has a visual cue of numeric number "1" indicating the position of the barcode in the

display. The numeric numbers increase by 1 from the left to the right and top to bottom. The block dot besides the numeric number “2” of the second barcode indicates that the barcode is at the end of the row. Similarly, the black rectangle associated with the 8th barcode indicates that the barcode is the last barcode of the message.

[0080] Furthermore, in addition to user visual inspection of the visual cues displayed by the terminal 112, the user 110 uses the mobile device 130 as an inspection device and places the camera of the mobile device 130 over the region of the screenshot to be inspected. The mobile device 130 captures and verifies the 2D barcode identified by the user 110. If it is authentic, the mobile device 130 displays the message embedded in the 2D barcode. Specifically, the mobile device 130 decodes the barcodes to reveal the messages embedded in the barcodes and displays the messages to the user 110. As illustrated in Figure 8, the camera of the mobile device 130 captures the second barcode. The barcode decoding module 134 of the mobile device 130 decodes the partial message 820 embedded in the second barcode and displays the message on the mobile device 130 for user verification. The non-sensitive portion of the screenshot is also displayed as it is to help the user 110 to navigate.

[0081] Figure 9 is the corresponding screenshot of the banking transaction webpage shown in Figure 7, which is encoded by the server 140 using the model 2 (where both the mobile device 130 and the terminal 112 are not trusted by the user 130, but one of them honestly performs its communication tasks). As illustrated in Figure 9, the sensitive user account information 920 is displayed by the terminal 112 side-by-side with its corresponding barcode 910 generated by the server 140. The user 110 uses the mobile device 130 to verify the barcodes. For example, the mobile device 130 needs to decode the barcode identified by the user 110 and extracts and generates a human readable proof from the authentication tag associated with the barcode. The

user 110 compares the proof generated by the mobile device 130 with the corresponding information displayed by the terminal 112 and rejects the barcode if the proof does not match the information displayed by the terminal 112.

[0082] In the example illustrated in Figure 9, the camera of the mobile device 130 captures the second barcode. The mobile device 130 decodes the barcode to reveal the message 930 embedded in the barcode. The mobile device 130 further displays the decoded message along with the corresponding user account information. The user 110 compares the decoded message with the user account information displayed by the terminal 112 and rejects the barcode if the decoded message does not match the account information displayed by the terminal 112.

[0083] An example of one embodiment of the computer system 100 is implemented on Android API targeting at OS version v1.6, and tested on three mobile devices: (1) a Acer Liquid mobile phone running on Android OS v1.6 with a 3.5 inches 480 x 800 TFT display screen, 256MB RAM, 768 MHz processor, video streaming maximum rate at 20 frames per second (fps); (2) a Motorola Milestone XT mobile phone running on Android OS v2.1-update1 with a 3.7 in 480 x 854 FWVGA display screen, 256MB RAM, 720 MHz processor and video streaming maximum rate of 24 fps; and (3) a HTC Legend mobile phone running on Android OS v2.1 with 3.2 inches 320 x 480 HVGA display screen, 384 MB RAM, 600 MHz processor, video streaming maximum rate at 30 fps. The implementation of the computer system 100 is tested on three different display units: (1) a 19 inch flat TFT monitor in Dell model Optiplex 755; (2) a 13.3 inch display of a Toshiba portege M900 laptop; and (3) a 15 inch Dell CRT monitor. Figure 10 shows one testing result displayed on the Toshiba portege M900 laptop monitor using the Acer Liquid mobile phone.

[0084]

[0085] The methods and techniques described herein can be performed by a computer program product and/or on a computer-implemented system. For example, to perform the steps described, appropriate modules are designed to implement the method in software, hardware, firmware, or a combination thereof. The invention therefore encompasses a system, such as a computer system installed with appropriate software, that is adapted to perform these techniques for creating soft shadows. Similarly, the invention includes a computer program product comprising a computer-readable medium containing computer program code for performing these techniques for creating soft shadows, and specifically for determining an extent to which an area light source is occluded at a particular shading point in an image.

[0086] The foregoing description of the embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Persons skilled in the relevant art can appreciate that many modifications and variations are possible in light of the above teaching. It is therefore intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

What is claimed is:

1. A computer-implemented method for encoding a message into one or more two dimensional barcodes with one or more visual cues, the method comprising:
 - encrypting an original message with a message encryption key and a message authentication key to generate an encrypted message, wherein the original message is a collection of strings, and the encrypted message contains an encrypted collection of the strings;
 - applying error correcting code to the encrypted message;
 - for each string of the encrypted collection of the strings in the encrypted message,
 - determining a visual cue suitable for the string; and
 - generating a two-dimensional barcode image containing the string of the encrypted collection of the strings, wherein the two-dimensional barcode visually resembles the visual cue.
2. The method of claim 1, further comprising:
 - adding control information to the embedded visual cue, wherein the control information indicates a relative position of the visual cue among a plurality of visual cues associated with the original message.
3. The method of claim 1, wherein the collection of strings of the original message comprises a plurality of fix-length strings organized in a form of data structure, wherein the visual cues are determined from the form of data structure.

4. The method of claim 3, wherein the data structure is a two-dimensional array, each entry of the array corresponds to one fix-length string of the collection of fix-length strings and each entry of the array is associated with a visual cue.

5. The method of claim 1, wherein encrypting the original message comprises:
applying the message encryption key to the original message; and
generating an authentication tag for the encrypted message and the visual cues
associated with the original message using the message authentication key.

6. The method of claim 1, wherein the error correcting code applied to the encrypted message is a (63, 36, 11) BCH error correcting code.

7. The method of claim 1, wherein visually resembling the visual cue comprises:
arranging the original message into a message binary image containing two or more
pixels;
associating every two pixel of the message binary image with one pixel of a binary
visual cue image, wherein the visual cue binary image represents the visual
cue;
forming a plurality of three-pixel blocks based on the message binary image and the
visual cue binary image; and
dividing the combinations of the three pixels of the block into a black group and a
white group.

8. The method of claim 7, wherein a combination of the three pixels of the block in a black group has at least two pixels of value of 1.

9. The method of claim 7, wherein a combination of the three pixels of the block in a white group has at least two pixels of value of 0.

10. The method of claim 7, further comprising constructing an output block consisting of the plurality of the three-pixel blocks, wherein the output block embeds the encrypted message and visually resembles the visual cue.

11. The method of claim 10, wherein constructing the output block further comprises applying a visual cue embedding key.

12. The method of claim 1, further comprises sending the two-dimensional barcode images to a user over a public network.

13. A computer-implemented method for decoding a message embedded in one or more two dimensional barcodes with visual cues, the method comprising:

capturing the two-dimensional barcodes displayed on a display screen, wherein each

barcode embeds a message and visually resembles a visual cue;

decoding the barcode to reveal the embedded message by:

extracting the embedded message and a message authentication tag;

authenticating the extracted message using the message authentication tag;

determining whether the extracted message contains an error that is larger than a pre-determined error threshold; and
displaying the embedded message to the user on a display unit.

14. A computer program product for encoding a message into one or more two dimensional barcodes with one or more visual cues, the computer program product comprising a non-transitory computer-readable storage medium containing computer program code for performing the operations:

encrypting an original message with a message encryption key and a message authentication key to generate an encrypted message, wherein the original message is a collection of strings, and the encrypted message contains an encrypted collection of the strings;
applying error correcting code to the encrypted message;
for each string of the encrypted collection of the strings in the encrypted message:
determining a visual cue suitable for the string; and
generating a two-dimensional barcode image containing the string of the encrypted collection of the strings, wherein the two-dimensional barcode visually resembles the visual cue.

15. The computer program product of claim 14, further comprising computer program code for:

adding control information to the embedded visual cue, wherein the control information indicates a relative position of the visual cue among a plurality of visual cues associated with the original message.

16. The computer program product of claim 14, wherein the collection of strings of the original message comprises a plurality of fix-length strings organized in a form of data structure, wherein the visual cues are determined from the form of data structure.

17. The computer program product of claim 16, wherein the data structure is a two-dimensional array, and each entry of the array is associated with a visual cue.

18. The computer program product of claim 14, wherein the computer program code for encrypting the original message comprises computer program code for:

applying the message encryption key to the original message; and

generating an authentication tag for the encrypted message and the visual cues

associated with the original message using the message authentication key.

19. The computer program product of claim 14, wherein the error correcting code applied to the encrypted message is a (63, 36, 11) BCH error correcting code.

20. The computer program product of claim 14, wherein the computer program code for visually resembling the visual cue associated with the original message comprises computer program code for:

arranging the original message into a message binary image containing two or more pixels;

associating every two pixel of the message binary image with one pixel of a binary visual cue image, wherein the visual cue binary image represents the visual cue;

forming a plurality of three-pixel blocks based on the message binary image and the visual cue binary image; and
dividing the combinations of the three pixels of the block into a black group and a white group .

21. The computer program product of claim 20, wherein a combination of the three pixels of the block in a black group has at least two pixels of value of 1.

22. The computer program product of claim 20, wherein a combination of the three pixels of the block in a white group has at least two pixels of value of 0.

23. The computer program product of claim 20, further comprising computer program code for constructing an output block consisting of the plurality of the three-pixel blocks, wherein the output block embeds the encrypted message and visually resembles the visual cue.

24. The computer program product of claim 23, wherein the computer program code for constructing the output block further comprises computer program code for applying a visual cue embedding key.

25. The computer program product of claim 14, further comprises sending the two-dimensional barcode images to a user over a public network.

26. A computer program product for decoding a message embedded in one or more two dimensional barcodes with visual cues, the computer program product comprising a non-

transitory computer-readable storage medium containing computer program code for performing the operations:

- capturing the two-dimensional barcodes displayed on a display screen, wherein each
 - barcode embeds a message and visually resembles a visual cue;
- decoding the barcode to reveal the embedded message by:
 - extracting the embedded message and a message authentication tag;
 - authenticating the extracted message using the message authentication tag;
 - determining whether the extracted message contains an error that is larger than a pre-determined error threshold; and
- displaying the embedded message to the user on a display unit.

27. A computer system for encoding a message into one or more two dimensional barcodes with one or more visual cues, the system comprising:

- a non-transitory computer-readable storage medium storing executable computer program modules comprising:
 - a barcode generation module configured to:
 - encrypt an original message with a message encryption key and a message authentication key to generate an encrypted message, wherein the original message is a collection of strings, and the encrypted message contains an encrypted collection of the strings;
 - apply error correcting code to the encrypted message;
 - for each string of the encrypted collection of the strings in the encrypted message:
 - determine a visual cue suitable for the string; and

generate a two-dimensional barcode image containing the string of the encrypted collection of the strings, wherein the two-dimensional barcode visually resembles the visual cue.
a processor for executing the computer program modules.

28. The system of claim 27, wherein the barcode generation module is further configured to:

arrange the original message into a message binary image containing two or more pixels;

associate every two pixel of the message binary image with one pixel of a binary visual cue image, wherein the visual cue binary image represents the visual cue;

form a plurality of three-pixel blocks based on the message binary image and the visual cue binary image; and

divide the combinations of the three pixels of the block into a black group and a white group.

29. A computer system for decoding a message embedded in one or more two dimensional barcodes with visual cues, the system comprising:

a non-transitory computer-readable storage medium storing executable computer program modules comprising:

a barcode decoding module configured to:

capture the two-dimensional barcodes displayed on a display screen,
wherein each barcode embeds a message and visually resembles a
visual cue
decode the barcode to reveal the embedded message by:
extract the embedded message and a message authentication tag;
authenticate the extracted message using the message
authentication tag;
determine whether the extracted message contains an error that is
larger than a pre-determined error threshold;
display the embedded message to the user on a display unit; and
a processor for executing the computer program modules.

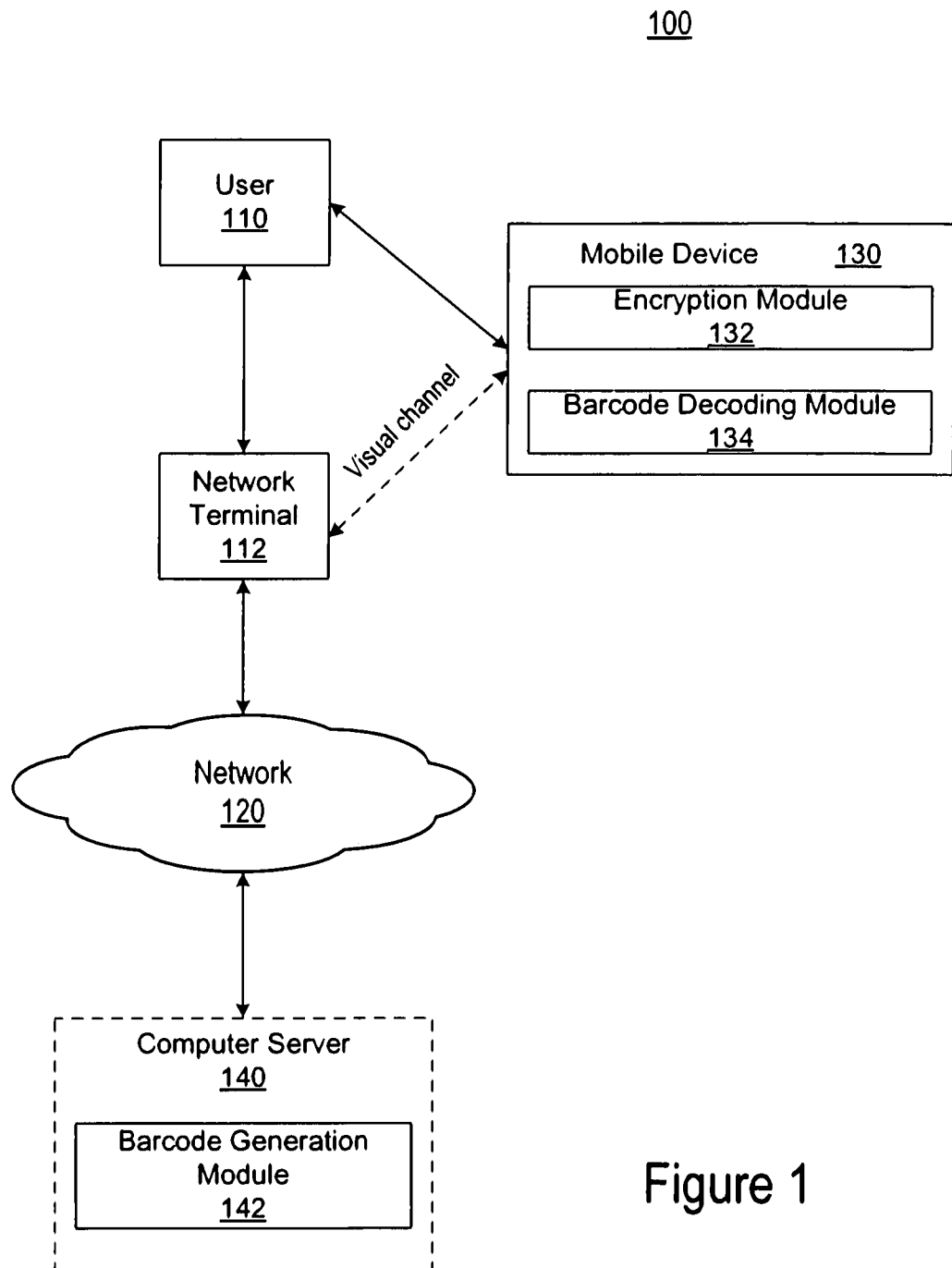


Figure 1

2 / 9

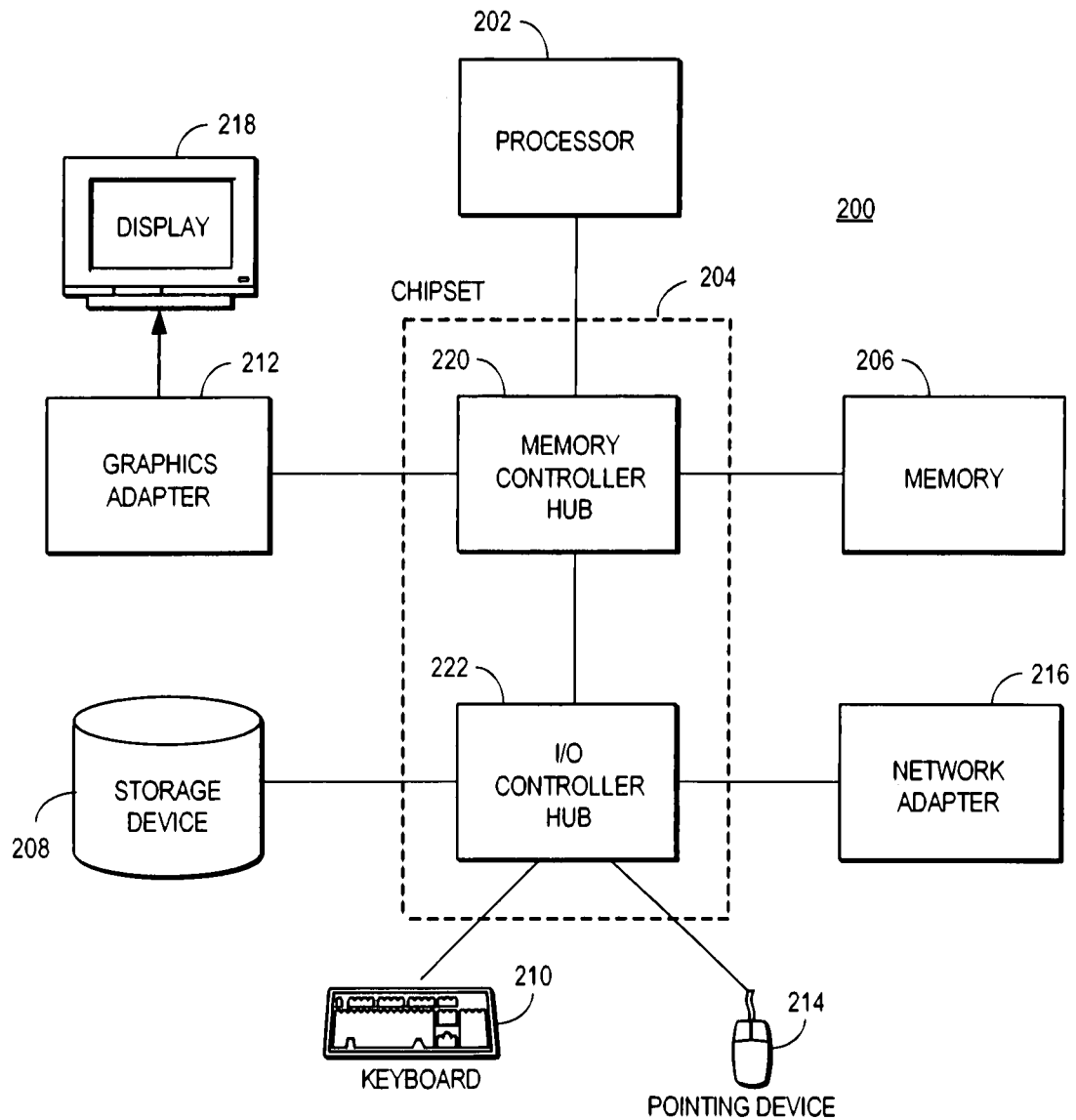


Figure 2

3/9

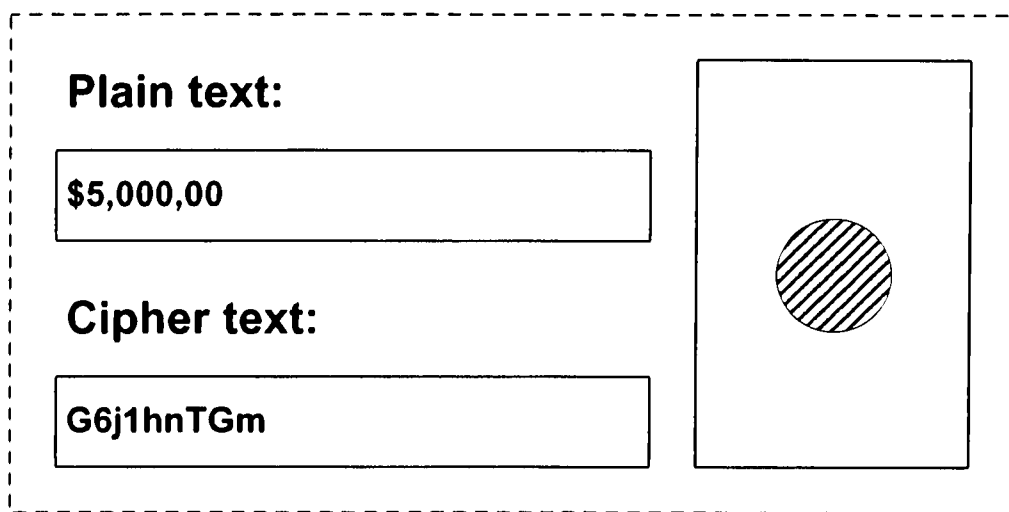


Figure 3

4 / 9

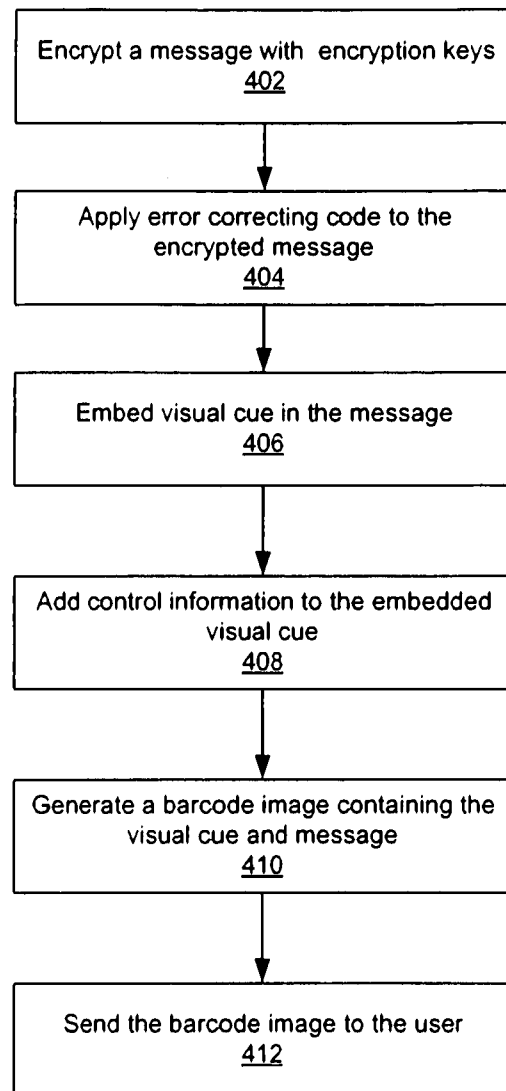


Figure 4

5/9

$$\begin{aligned}
 W &= \left\{ \begin{array}{cc} \square & \square \\ \square & \end{array} \quad \begin{array}{cc} \blacksquare & \square \\ \square & \end{array} \quad \begin{array}{cc} \square & \blacksquare \\ \square & \end{array} \quad \begin{array}{cc} \square & \square \\ \blacksquare & \end{array} \right\} \\
 B &= \left\{ \begin{array}{cc} \square & \blacksquare \\ \blacksquare & \end{array} \quad \begin{array}{cc} \blacksquare & \blacksquare \\ \square & \end{array} \quad \begin{array}{cc} \blacksquare & \square \\ \blacksquare & \end{array} \quad \begin{array}{cc} \blacksquare & \blacksquare \\ \blacksquare & \end{array} \right\}
 \end{aligned}$$

Figure 5A

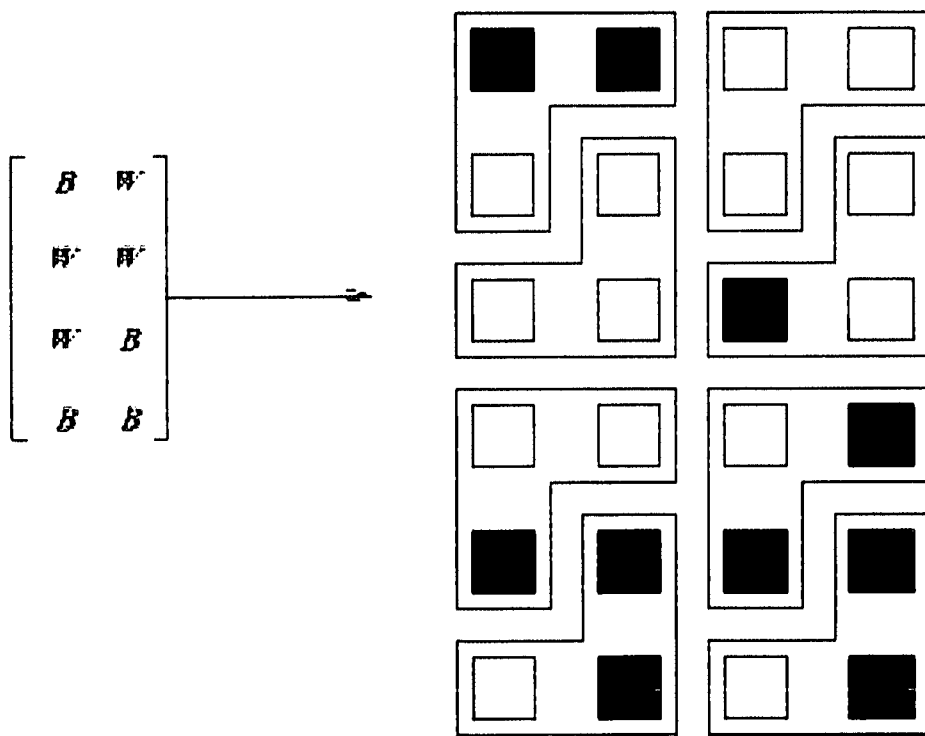


Figure 5B

6/9

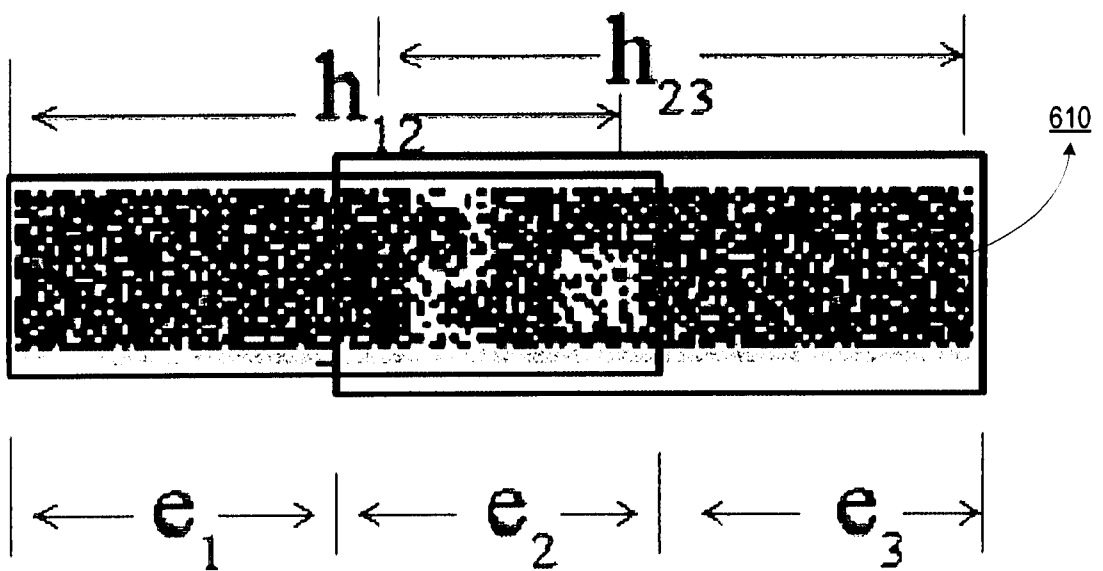


Figure 6

7/9

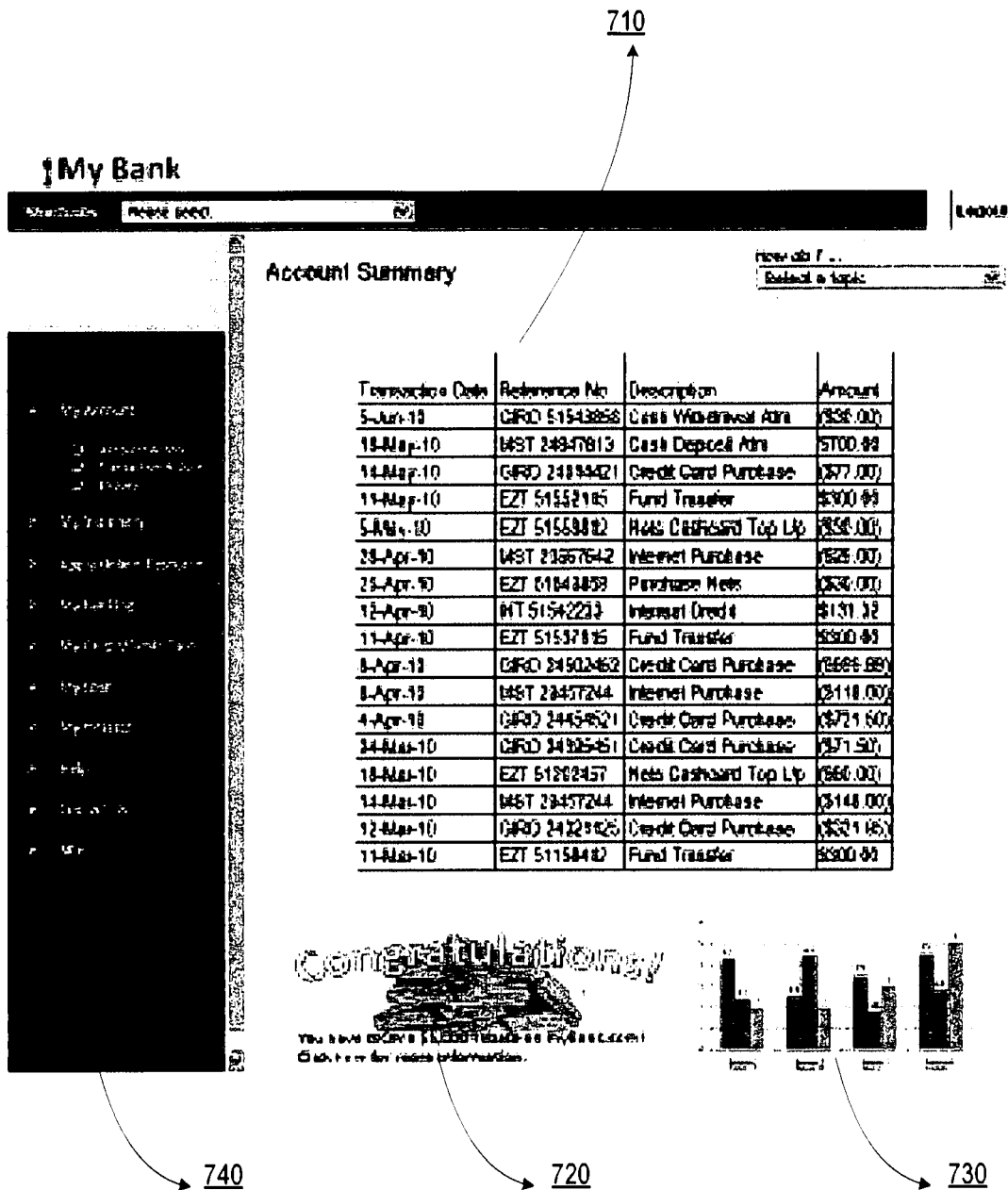


Figure 7

8/9

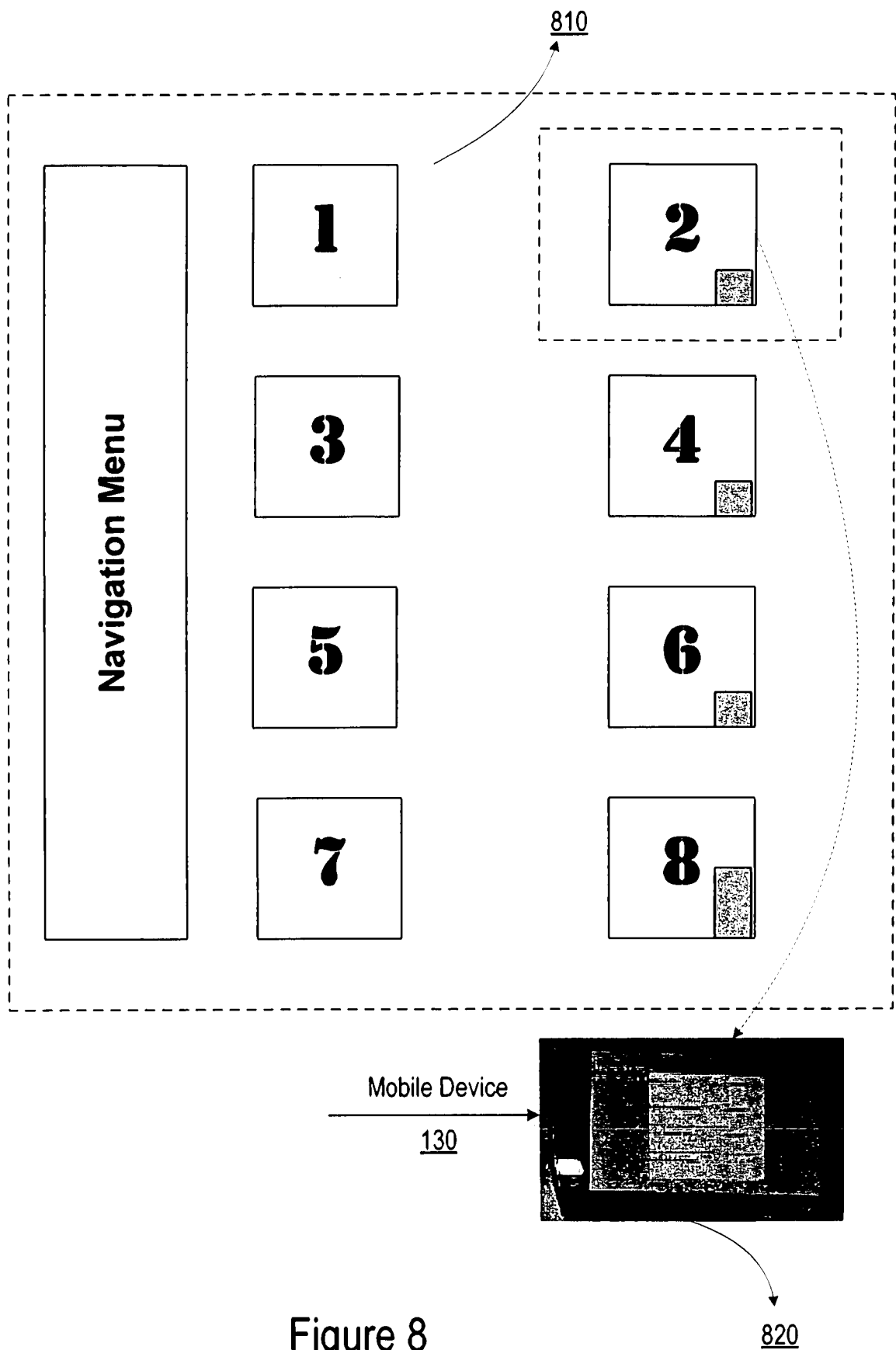


Figure 8

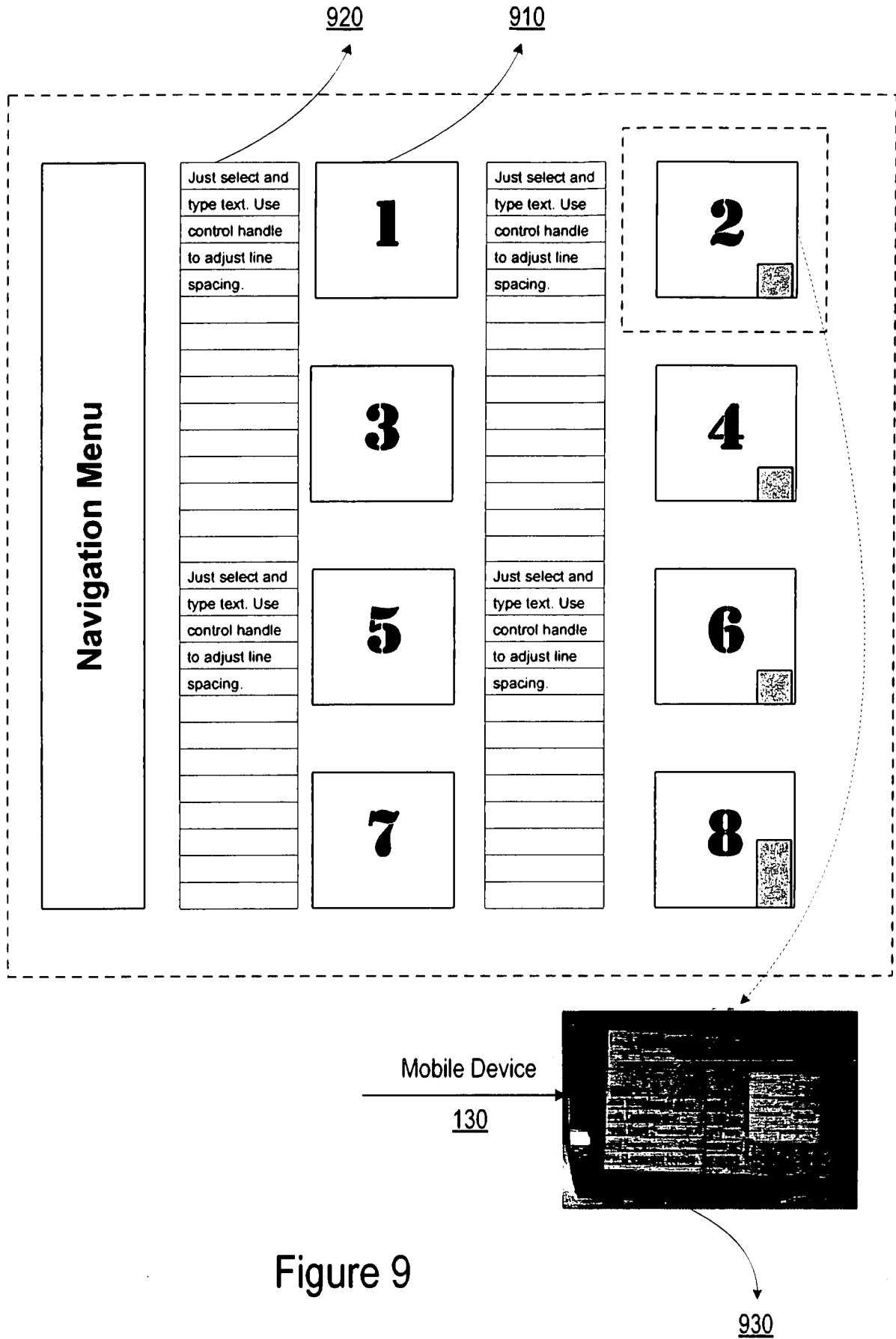


Figure 9

INTERNATIONAL SEARCH REPORT

International application No.

PCT/SG2011/000064

A. CLASSIFICATION OF SUBJECT MATTER

Int. Cl.

G06K 19/06 (2006.01)

H04L 9/14 (2006.01)

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)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPAT, EPODOC (keywords: barcode, dimension, matrix, encrypt, logo and similar terms)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2007/0102521 A1 (PETERSSON) 10 May 2007, see the abstract; figure 3A; paragraph 25 in context of the whole document	1-6, 12-19, 25-27, 29
Y	US 2003/0161475 A1 (CRUMLY et al) 28 August 2003, see figure 5; paragraph 29 in context of the whole document *combined with US 2007/0102521	1-6, 12-19, 25-27, 29
Y	WO 2007/107533 A2 (CONNVISION AG) 27 September 2007, see page 4 lines 8-22; page 22 lines 11-25; figures 4-7 in context of the whole document *combined with US 2007/0102521	1-6, 12-19, 25-27, 29



Further documents are listed in the continuation of Box C



See patent family annex

* Special categories of cited documents:	
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"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
"O" document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

20 April 2011

Date of mailing of the international search report

27 APR 2011

Name and mailing address of the ISA/AU

AUSTRALIAN PATENT OFFICE
PO BOX 200, WODEN ACT 2606, AUSTRALIA
E-mail address: pct@ipaaustralia.gov.au
Facsimile No. +61 2 6283 7999

Authorized officer

TIM YANG
AUSTRALIAN PATENT OFFICE
(ISO 9001 Quality Certified Service)
Telephone No : +61 2 6283 7923

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/SG2011/000064

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report			Patent Family Member	
US	2007102521	NONE		
US	2003161475	GB	2386784	JP 2003304233
WO	2007107533	NONE		
Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001.				
END OF ANNEX				