



- (51) International Patent Classification:  
*G06F 21/00* (2006.01) *H04L 9/32* (2006.01)
- (21) International Application Number:  
PCT/US2012/033262
- (22) International Filing Date:  
12 April 2012 (12.04.2012)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
13/106,245 12 May 2011 (12.05.2011) US
- (71) Applicant (for all designated States except US): **KONVAX CORPORATION** [US/US]; 485 Massachusetts Avenue, Suite 300, Cambridge, Massachusetts 02139 (US).
- (72) Inventors; and
- (71) Applicants : **TALAMO, Maurizio** [IT/IT]; Via Emanuele Filiberto, 233, I-00185 Rome (IT). **ARCIERI, Franco** [IT/IT]; Via Mario Chiri, 25, I-00177 Rome (IT). **MARINELLI, Guido, Maria** [IT/IT]; Viale Appio Claudio, 36, I-00178 Rome (IT). **SCHUNCK, Christian, H.** [DE/IT]; Via Umberto Biancamano, 33, I-00185 Rome (IT).
- (74) Agent: **ROHLICEK, J., Robin**; 321 Summer Street, Boston, Massachusetts 02210 (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, 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, 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, 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).

[Continued on next page]

(54) Title: SECURE USER CREDENTIAL CONTROL

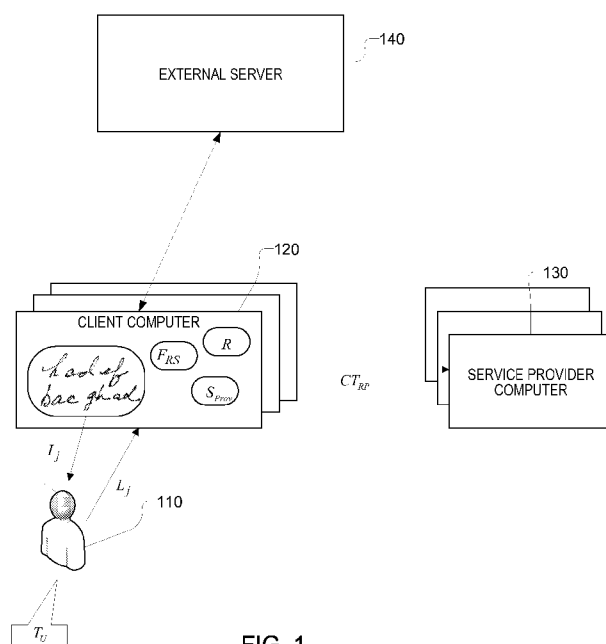


FIG. 1

(57) Abstract: In some embodiments, a user has use a single universal text- or image-based secret for generating a service-provider specific identity credential, for example username plus password, for authentication is derived. A human (i.e., the user) must interpret an image to enter this universal text (or image) based secret. For example, an image based challenge is presented to the user, and a credential is obtained based on the users response to the challenge.

**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

**Published:**

- *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

## SECURE USER CREDENTIAL CONTROL

### Cross Reference to Related Applications

[001] This application claims priority to United States Application No. 13/106,245, filed May 12, 2011, the contents of which are incorporated in their entirety.

### Background

[002] This invention relates to secure control of user credentials, and in particular, relates to security based on image-based generation of credentials.

[003] Management of user identifiers and passwords in cyberspace has become a major vulnerability of computer based systems and has become one of the major issues in countering cybercrime and breaches of commercial online systems. In April 2011, the U.S. Whitehouse issued a report titled “National Strategies for Trusted Identities in Cyberspace,” which discusses a number of open issues that must be addressed to create a secure and privacy conserving ecosystem for digital identities.

[004] One major issue in cyberspace today is that users of computer and network based services must manage a myriad of passwords to gain access to those services. For example, a user may have a different password for each online account (e.g., banking, shopping, etc.). Each service provider may have different rules for the passwords, for example, requiring a minimum number of characters, use of special characters, frequency of change, etc. The user may also provide a different identity to different service providers, for example, by providing a user-selected username or alias by which the user is identified by the provider. Such identifiers in combination with a password (or another form of shared secret) form identity credentials that the user presents to the service provider for access. From the user’s perspective, the increasing complexity and risk of managing multiple credentials threaten the convenience associated with online transactions. These factors also impart an increased risk of account compromise through insecure user management of account credentials and an increased likelihood of account abandonment.

[005] One solution to the above mentioned problem is a so-called password vault where service provider specific passwords are stored. The vault is protected by a “master”

password which is in the possession of the user. Using this password the user can access or give applications access to the other passwords stored in the vault.

[006] Some systems improve security by requiring that separately from providing a username and password, the user respond to an image-based CAPTCHA (“Completely Automated Public Turing test to tell Computers and Humans Apart”) to inhibit computer-based password guessing attacks. Some systems have used graphical passwords in which a user remembers an image, and authentication of the user by a server involves providing a set of images for display to the user, and then determining whether the user selected the same image.

### Summary

[007] In one aspect, in general, one or more embodiments described in this document allow a user to use a single (universal) text- or image-based secret to generate a recipient-specific random (e.g., pseudo-random) number and deriving from it a robust, recipient specific credential (“token”), for example representing but not limited to a username and/or password and/or other identifying (e.g., serial number MAC address) or verifying (e.g., PIN) information, for authentication or verification. In some examples, a human (i.e., the user) must interpret an image to enter this universal text (or image) based secret. For example, an image based challenge is presented to the user, and a pseudo-random number is obtained based on the user’s response to the challenge.

[008] In some versions, the infrastructure needed for the image-based challenges is provided by an external server (in some examples referred to as an “external authority” or a “validation server”) without burdening the systems seeking to authenticate the user (e.g., the service providers). This is done without the universal secret being known to the external server. Possible implementation alternatives include: 1) the external server does not know the user’s service provider specific credentials; and 2) the external server is authorized from the user to present the service provider specific credentials to the service provider

[009] In some implementations the approach can provide proof to a recipient (e.g., to a service provider) that the user answered correctly to an image-based challenge. Such a proof can also be used by the service provider to generate a new credential for the user.

[010] In another aspect, in general, a service-provider specific credential is generated based on a user’s secret. The method includes providing a challenge to the user, and

accepting a response from the user determined by the user based on the user's secret and the challenge. An identifier of a service provider is determined, and a credential-generating procedure is applied to combine the response from the user and the identifier for the service provider to form the service-provider specific credential. The service-provider specific credential is provided for transmission to the service provider without providing information sufficient to determine the user's secret in the transmission.

**[011]** In another aspect, in general, a method is directed to generating a recipient-service-provider specific (e.g., service-provider specific) credential based on a user's secret using a computing device of the user. An image-based challenge (e.g., based on image data) is provided to the user, for instance, for presentation to the user on an electronic display (e.g., a graphical touchscreen). The challenge comprises a spatial arrangement of a set of perceptually obscured symbols. A response is accepted from the user who determined the response based on the user's secret and the challenge and entered the response using an input device (e.g., a stylus, touchscreen). The response is based on or represents a sequence of locations in the spatial arrangement. A computer implemented credential-generating procedure is applied to combine the response from the user, an identifier of the recipient service provider, and a user-specific random number access from a data storage to form the recipient-service-provider specific credential, which may be as a pseudo-random number, and which is provided for transmission to the recipient (e.g., service provider over a data communication link) without providing information sufficient to determine the user's secret from the transmission.

**[012]** In another aspect, in general, a method for repeatedly soliciting credentials from a user includes, on each of a sequence of repetitions, externally to the computing device of the user, generating a different image-based challenge. The challenge comprises a spatial arrangement of a set of perceptually obscured symbols. A map corresponding to the spatial arrangement of the symbols is also generated, and the challenge and corresponding mapping data are provided over a secure communication link to the computing device of the user for presentation to the user and for soliciting a sequence of locations in the spatial arrangement. The mapping data provides a relationship between the spatial arrangements of the symbols in the plurality of repetitions.

**[013]** In another aspect, in general, an image-based method for verifying a user response includes providing a first image-based challenge to the user and accepting a first response to the first challenge from the user, including generating a first sequence (for instance, a first pseudo-random number) representing the user's response. Subsequently a second

image-based challenge is provided to the user and a second response to the second challenge is accepted, including generating a second sequence (for instance, a second pseudo-random number) representing the user's response. Equality of the first sequence and the second sequence serves as a proof that the second user is a human and that the second user knew the first user's first response to the first challenge (e.g., that the first user and the second user are the same user). In some examples, the steps of presenting the second image-based challenge and accepting the second response are performed as part of performing a credential generating procedure. In some examples, the method includes providing the proof for unlocking a secured physical device (e.g., a door) and/or a software application.

**[014]** One or more aspects of the approach can include one or more of the following features in combination.

**[015]** Providing the challenge to the user includes providing challenge information required to be processed by the human in determining the response.

**[016]** Determining the response based on the user's secret and the challenge requires human processing of the challenge.

**[017]** Applying the credential-generating procedure includes selecting portions of a user's fixed random number and combining the selected portions.

**[018]** Combining the selected portions includes applying a reduction function to each portion and combining the reduced portions.

**[019]** The selected portions of the user's random number form less than 1/1,000 of said random number, and preferably less than 1/10,000 of said random number.

**[020]** The credential-generating procedure is applicable to the response from the user with any of a plurality of identifiers for a plurality of service providers to form a different service-provider specific credential for each of said service providers.

**[021]** The service-provider specific credential comprises a password associated with the user and the service provider.

**[022]** The service-provider specific credential comprises at least one of a name, preferably a user name, a mail address (e.g., electronic mail address), and a Uniform Resource Locator (URL).

- [023] The credential includes an identity attribute, for example, a birthdate, in association with the user's identity.
- [024] The response is determined based on the user's secret and the challenge requires human processing of the challenge.
- [025] Determining the user's secret from the user's response requires human processing of the challenge.
- [026] Providing the challenge to the user includes providing image data for presentation to the user.
- [027] The image data represents a spatial arrangement of a plurality of symbols from an alphabet of symbols.
- [028] In some implementations the user chooses the alphabet of symbols as a subset of symbols from a large variety of symbols, logos, images, and alphanumeric characters in various alphabets.
- [029] The spatial arrangement of symbols that is provided to the user always contains the complete, previously selected subset of symbols, but may contain additional different symbols that change from time to time. In some embodiments the same symbols may appear several times in one image.
- [030] The spatial arrangement comprises a perceptual obscured spatial arrangement of the plurality of symbols.
- [031] The perceptually obscured arrangement of symbols includes at least one of spatially distorted symbols, overwritten symbols, and overlapping symbols.
- [032] The spatial arrangement comprises a grid based arrangement.
- [033] The spatial arrangement comprises an irregular two-dimensional arrangement, preferably including at least some overlapping of perceptually obscured representations of multiple of the symbols.
- [034] The alphabet of symbols comprises a set of characters from a character set, or some other set of predefined images (e.g., logos, etc.).
- [035] The character set comprises a set of alphabetic and/or numeric characters.

[036] Accepting the response from the user includes accepting a set of selections from the challenge.

[037] Accepting the set of selections includes accepting a sequence of selections.

[038] The set of selections comprises a set (e.g., a sequence) of spatial locations in an image represented by the challenge.

[039] The sequence of spatial locations comprises a sequence of grid locations.

[040] The spatial locations are mapped to a set of regions in the image.

[041] The image based challenge is generated and presented to the user from an external server. To get access to this service the user registers with this external server. The registration can be anonymous. In some implementations the user declares his universal token explicitly (for example to allow recovery in case of memory loss).

[042] Applying the credential generating procedure includes selecting a service provider specific function according to the identifier of the service provider, and then applying the service provider specific function to an argument determined from the response from the user.

[043] Applying the credential-generating procedure includes accessing a set of values associated with locations in the challenge, and then combining the response from the user and the set of values in forming the service provider specific credential.

[044] Each location is associated with a sequence of the values, and wherein combining the response and the values comprises for each element in a sequence of locations in the response, selecting a corresponding value in the sequence of values associated with that location.

[045] The set of value associated with the locations is updated to conform to a change in the user's secret such that a subsequent application of the credential -generating procedure yields the same service-provider specific credential based on a subsequent response from the user based on the changed secret.

[046] The challenge and associated mapping data is received from an external server.

[047] If the challenge and associated mapping is received from an external server the user registers with this external server. This can be done anonymously by establishing a unique username/nickname (for the domain of this external server). The server then

dispatches a software (user agent) to the user's computing devices (including smartphones etc.).

[048] Applying the credential generating procedure further comprises combining the mapping data with the response from the user to determine a mapped response

[049] Combining the response from the user and the identifier for the service provider comprises combining the mapped response and the identifier for the service provider.

[050] The steps set forth above are repeatable using a plurality of different challenges to the user yielding a plurality of different responses from the user determined by using the same user's secret such that the service-provider specific credential remains the same for each repetition.

[051] Each of the plurality of challenges represents a different perceptual obscured spatial arrangement of a plurality of symbols from an alphabet of symbols.

[052] In another aspect, in general, software stored on a computer readable medium comprising instructions for causing a data processing system to perform any combination of steps set forth above.

[053] In another aspect, in general, a system is configured to perform any combination of steps set forth above.

[054] Aspects may have one or more of the following advantages.

[055] A number of advantages relate to the problem of providing security for the user, for example, by making it more difficult for adversaries to compromise online transactions. Some specific examples of such advantages include the following.

[056] Providing a credential (e.g., a domain plus a secret) entry approach that requires a human to respond to a challenge to prove its identity makes it difficult or impossible for a machine-based attack to be successful or efficient.

[057] It is hard for malicious software to obtain the universal secret since it is not stored on any computing device at any time and it is not disclosed to the external server or the service provider.

[058] The service provider specific credentials are very robust since they are derived from pseudorandom numbers.

[059] The user has to protect only one secret.

[060] Even if the user's secret is compromised an adversary cannot derive the service provider specific credentials.

[061] The user can change his secret anytime (even after each use) without affecting the service provider specific credentials.

[062] The secret part of the service provider specific credentials can be changed while the user keeps his secret.

[063] In order to obtain the service provider specific credential it is necessary to get hold of the secret, the service provider specific pseudorandom number and the specific algorithm to derive the service provider specific credential. Without any of these three "ingredients" it is very hard to obtain the service provider specific credential

[064] These three items (secret, the pseudorandom number and the algorithm) are stored in different places. Moreover the service-provider specific credentials are only stored on the servers of the respective service provider. For each use the service-provider specific credentials are calculated again by the software agent (based on the user's input, the pseudorandom number and the algorithm)

[065] The user generates a pseudo-random number. An adversary that is not in possession of the mapping function between grid locations and pseudorandom number parts cannot map the pseudorandom number back to grid locations.

[066] As the user enters a sequence of symbols already known to him, the CAPTCHA symbols can be made more demanding for the human to recognize than in a scheme where the user has to recognize a sequence of symbols that is new/unknown to him. Therefore there is the potential to increase the difficulty of the underlying AI problem

[067] The approaches provide a strong defense against key-logging attacks, at least in part because the inputs are locations on a grid as opposed to key presses on a keyboard.

[068] A number of advantages relate to efficiency based on convenience for individuals who may choose to manage fewer passwords or accounts than they do today. Some specific examples of such advantages include the following.

[069] It is easier for the user to protect a single secret based on which the service-provider specific pseudorandom numbers can be generated (from which the robust, service-provider specific credentials are derived).

[070] The user can change his secret frequently (at any time he wants) without changing the service provider specific credentials

[071] The user proves in a single action both that he is human and that he knows the credential for the server.

[072] The same infrastructure can be provided to the user for use with many different servers, without adding additional burden on the external server and the user.

[073] Even if all the passwords from a set of service providers are compromised, and even if further the user's computer is compromised, an adversary does not have enough information to determine the user's universal secret or to generate credentials for other service providers.

[074] A number of advantages relate to ease of use by automating identity solutions whenever possible and basing them on technology that is easy to operate with minimal training. Some specific examples of such advantages include the following.

[075] Remembering only a single secret a user can generate the service-provider specific pseudorandom numbers from which the robust, service-provider specific credentials are derived.

[076] The user proves in a single action both that he is human and that he knows the credentials for the server.

[077] The user entry of his secret is solely (touch) screen based, or uses a pointing device like a mouse. Therefore no keypad is required. This makes this approach not only suitable for use on standard computers but also only touch and pointing based computing devices like some smartphones and tablet computers.

[078] Making this approach suitable for smartphone use with small screen can include implementing image browsing features, such as display of only part of the image, zoom, touching the same symbol x times in a row tells the system that this symbol corresponds to the xth symbol in the universal password.

[079] The user automatically generates robust credentials without the burden and complexity to evaluate or compute complex alphanumeric patterns (i.e. the user does not need to make many attempts to create a credential that is sufficiently strong).

[080] For the user it is easier to recognize the CAPTCHA symbols as they correspond to its secret, as compared to recognizing a previously unseen word or phrase.

[081] The credential entry is a single interaction that both proves that a human entered the secret and that the user knew the secret, as compared to using two separate interactions and/or mechanisms to prove these two aspects.

[082] A number of advantages relate to confidence that digital identities are adequately protected, thereby increasing the use of the Internet for various types of online transactions. Some specific examples of such advantages include the following.

[083] The user increases his confidence in using online transactions, because he is ensured that his credentials are strong and well protected.

[084] Increased privacy for individuals, who rely on their data being handled responsibly and who are routinely informed about those who are collecting their data and the purposes for which it is being used.

[085] The user can easily generate multiple credentials (for example email-addresses plus password) to exchange information with many different servers avoiding that data used in different service provider transactions are being linked based on the use of the same part of a credential (email-address) with these service providers.

[086] The user can easily classify the service providers in different categories so that the service provider specific identifiers are searchable using keywords. The user can have a different universal secret for each category.

[087] In some versions of the invention the separation of tasks between user and external server leads to an increased protection of the user's service provider specific credentials. Only the combination of the user's universal secret and of the image and corresponding map provided by the external server allow the re-generation of the user's service provider specific credential. There is some analogy to the protection of safe deposit boxes in banks where two keys (the customer's and the bank's) are required for access. In such a scenario a user who's universal secret and computing devices have been compromised can request the external server not to provide further images and maps or to continue to provide the images but with incorrect maps.

[088] Other features and advantages of the invention are apparent from the following description, and from the claims.

#### Description of Drawings

[089] FIG. 1 is a block diagram of a credential control system.

[090] FIG. 2 is a flowchart of a credential generation procedure.

[091] FIGS. 3A-B are illustrations of image/map pairs.

[092] FIG. 4 is a binary AND/OR tree.

#### Notation

[093] The following notation is generally followed in the Description below.

[094]  $U$  : a user that is providing the credential.

[095]  $S$  : a Service provider to which the User is providing a service provider specific credential

[096]  $S_{Prov}$  : an identifier for the service provider, for example, a random number that is unique to the service provider

[097]  $CT_{RP}$  : a secret credential (or “identifying credential”) being provided by the user to the service provider.

[098]  $T_U$  : a secret (“universal token”) in the mind of the User

[099]  $R$  : a random number that is specific to the User (more precisely, this may be referred to as  $R_U$ ), which may be generated to be a cryptographically strong “book of random numbers” that is used by functions at the user’s computer.

[0100]  $I_j$  and  $M_j$  : a corresponding pair of an image and a map, respectively. The map provides a function from a domain of locations (e.g., x-y coordinates) in the image to a range of indices (e.g., integers).

[0101]  $A$  : an external server or other entity/module (e.g. validation authority, CAPTCHA server) that generates the corresponding image/map pairs.

[0102]  $L_j = (l_1, l_2, \dots, l_k)$  : a sequence of locations selected by the user in the image  $I_j$ .

[0103]  $Q = (q_1, \dots, q_k) = M_j(L_j)$  : a corresponding sequence of mapped indices.

[0104]  $P$  : a translation of the sequence of mapped indices  $Q$ .

[0105]  $S_{RP} = W_S(R, P, S_{Prov})$  : a sequence of subsequences of  $R$  obtained by applying a function  $W_S$  to  $R$ ,  $P$ , and  $S_{Prov}$ .

[0106]  $F_{RS}$ : a service provider specific logical function generated using  $R$ .

[0107]  $CT_{RP} = F_{RS}(S_{RP})$ :  $F_{RS}$  applied to a single subsequence of  $S_{RP}$

### Description

#### 1 Overview

[0108] Referring to FIG. 1, in an exemplary embodiment, a number of users 110 make use of services provided by a number of service providers. For example, a single user may make use of a number of different services provided by different service providers, including, for example, separate banking, shopping, and social networking services. Each user has a user computer 120 with which he interacts to access the services. Generally, users must provide credentials (e.g., username and password) to each service provider via a service provider computer 130 in order to gain access to the services of that service provider. The discussion below focuses on a single user 110 and user computer 120, but it should be clearly understood that the approach scales to very large numbers of users without adding burden to the service providers.

[0109] Very generally, in some embodiments, generation of the credentials for a particular user for presentation to a particular service provider involves the following steps, as illustrated in FIG. 2:

- A random arrangement of elements is presented in an image ( $I_j$ ) (i.e., the  $j^{th}$  image ever presented) to the user ( $U$ ). For example, the image has distorted characters at different locations in image (block 210);
- A location sequence ( $L_j$ ) corresponding to user's universal secret ( $T_U$ ) is accepted from the user. For example, the user uses a mouse or a touch-sensitive screen to spell out his universal secret using the distorted characters in the image (block 220);
- The location sequence accepted from the user is combined with a map ( $M_j$ ) of the image ( $I_j$ ) producing a translation ( $P$ ) of the location sequence, such that this translation is invariant over different images presented to the user (block 230);
- The translation ( $P$ ) and the service provider identifier ( $S_{Prov}$ ) are used to determine a sequence of portions of the user's random number ( $R$ ), which functions as a user-specific "book" of random numbers. The random numbers in the sequence may be further reduced (e.g., in a service-provider-specific manner) to form the parts of the

part of the binary credentials for the service provider. For example, a provider-specific function ( $F_{RS}$ ) is applied to portions of the random number ( $R$ ) selected according to the identifying quantity for the service provider ( $S_{RP}$ ) and on the translated number ( $P$ ) to form a binary representation ( $CT_{RP}$ ) of the credentials (block 240); and

- The credentials ( $CT_{RP}$ ) are formatted (e.g., alphanumeric and special characters that satisfy the requirements of form for that service provider, for instance, at least 8 characters including at least one digit and one punctuation character) and provided to the service provider (block 250).

**[0110]** Note that there are a number of pre-requisites to performing these steps. A first set prerequisite steps are common and are performed before enrolling any particular service provider. These steps include:

- The user selects his universal secret ( $T_U$ ) but does not disclose it to anyone or to any computer. Therefore, this secret remains only in the mind of the user at this time.
- The user's secret random number ( $R$ ) is set and stored, in this embodiment, on the user's computer. In some examples, a cryptographically strong random number generator is executed at the external server, and the resulting random number (which may have over  $10^5$  or over  $10^6$  bits) is securely transferred to the user's computer.
- In some embodiments, the external server optionally provides the identifying quantities ( $S_{Prov}$ ) for service providers, for example, being strong random numbers generated by the external server.

**[0111]** A number of steps are performed in enrolling any particular service provider. These steps include:

- A service-provider-specific function ( $F_{RS}$ ) is selected at the user's computer, preferably at random from a large set of available function. A characterization of the function is stored at the user's computer in association with an identifier for that service provider.
- The steps shown in FIG. 2 are carried out, and the service-provider-specific credentials are provided to the service provider for initial enrollment. For example, a username and password are generated, and used to enroll with the service provider.

[0112] Note that the randomly arranged image and corresponding map are such that regardless of the random arrangement of the elements in the image, the translation ( $P$ ) of the location sequence is the same on each repetition. Because the service-provider specific function is fixed and the arguments to the function are fixed or identically regenerated on each repetition, this means that derived credentials are the same each time the user regenerates them for that service provider.

[0113] There are a number of characteristics of this approach that should be noted.

- The user's universal secret ( $T_U$ ) is not stored on any computer at any time.
- Determining the user's universal secret ( $T_U$ ) from the translations ( $P$ ) requires human interpretation of the image ( $I_j$ ), and knowledge of the map ( $M_j$ ) corresponding to the image.
- Based on the design of the service-provider-specific functions, knowledge of the credentials for one service provider does not allowed determination of the credentials for another service provider.

[0114] Below, each of these steps are discussed in detail.

[0115] Note that the approach to credential generation is not limited to generating a password for the user. For example, in some examples, the user can also enter a "nickname" for a service provider using such an image-based approach. This can avoid storage of service-provider specific information that could compromise access to the provider. Also, it should be understood that other types of information may be provided by the user such that the service provider obtains the information and a proof that the information was provided by a human while only requiring the single information entry step by the user.

## 2 Image and Map Generation

[0116] As introduced above, in general, an image presented to a user has a set of characters arranged in a manner that the user can select a sequence of the characters. Referring to FIG. 3A, in an example in which the user's universal secret is represented in the user's mind as a sequence of letters from the set "a" through "f", for example, the word "headache". In this example, a image ( $I_1$ ) includes eight tiles, one tile per letter. The map ( $M_1$ ) provides an index corresponding to each tile. For ease of presentation, in this example, the mapped value for the tile with a rendering of the letter "a" is 1, the tile

for “b” is 2, and so forth, but there is no requirement that the letters correspond to indexes in such an ordered manner.

[0117] In a subsequent image ( $I_j$ ), the tiles are rearranged (e.g., permuted), but the map ( $M_j$ ) has been similarly rearranged so that that is it consistent with the original image/map pair, and using the same sequence of characters the same sequence of mapped indices is produced.

[0118] Therefore, in some embodiments, repeated generation of image/map pairs for a user makes use of knowledge of the original association of characters to mapped indices so that for each new image, a suitable map may be provided. For example, referring to FIG. 1, image/map pairs for a user may be generated by an external server 140 (which is referred to as a “CAPTCHA server” in some implementations), and provided over a secure (e.g., encrypted and authenticated) communication link between the user’s computer and the external server. The external server can maintain an original association of characters to indices, for example, originally chosen at random, so that each subsequent map corresponds to the map originally provided to the user. Note that the map may be provided in incremental form (e.g., as a permutation), so that the user can reconstruct the map using the incremental components, but such that an adversary that may gain access to one of the incremental components is unable to reconstruct the map.

[0119] Referring to FIG. 3B, note that characters do not have to be presented on a regular grid. For example, the characters may be distorted across an image, and each pixel may be mapped to an index, for example, forming irregular regions of equal mapped index values.

[0120] Furthermore, it should be understood that yet more complex distortions of the characters may be used in some embodiments, for example, with overlapping, variation in color, non-character overlapping graphics, etc. On the other hand, in some embodiments, regular (i.e., “clean”) renderings of rearranged square tiles may also be used.

[0121] In some examples, the space of images used to represent the user’s secret is selected by the user. For example, in an enrollment phase, the user may select a subset of images available to the system. For example, the user may select the subset consisting of upper case roman letters, or may select lower case letter, numbers, logos of sports teams, etc. as the space of images, or any combination. The user selects his secret from this selected domain of images. The image generation process guarantees that at least this

subset is presented in each image (thereby guaranteeing that the user can enter his secret), possibly augmented by other images.

### 3 Location Sequence Entry

[0122] When the user is presented with the image, he uses his knowledge of his universal secret to generate a sequence of locations. Each location is passed through the corresponding map to produce a sequence of mapped locations ( $Q$ ) corresponding to selected locations. For example, for any of the image/map pairs shown in FIGS. 3A-C, when the user selects the locations for the sequence of  $k = 8$  characters in the word “headache”, the resulting sequence of mapped locations is

$$Q = (q_1, \dots, q_k) = (8, 5, 1, 4, 1, 3, 8, 5).$$

[0123] In some systems, for example, systems with relatively limited screens, the user can scroll and/or zoom across the image in selecting the successive elements of his universal secret. In some systems, the user may select the elements of his secret out of order, and arrange the order in a separate part of the interface screen (e.g., as an analogy to arranging a set of Scrabble tiles).

### 4 Location Sequence Translation

[0124] In some embodiments, the index sequence  $Q$  is used directly to form the translation ( $P$ ) of the location sequence. For example, with index values in the range  $1 \leq q_i \leq 8$  could be translated as

$$P = \sum_{i=1}^k (q_i - 1) 8^{(i-1)},$$

which for  $k = 8$  yields a 24-bit integer.

[0125] In some embodiments, a further degree of translation is provided by using an array of random number  $r_{q,i}$  where  $q$  is over the range of the index map, and  $i$  is over the possible length of the user’s universal secret. For example, if there are 8 possible characters, and the user’s secret is at most 10 characters long, a total of 80 random numbers  $r_{q,i}$  are needed. In some examples, this set of random numbers may be provided along with the map, or alternatively may be generated and kept secret by the user. One way of computing the translation ( $P$ ) using such random numbers is as

$$P = \sum_{i=1}^k r_{q,i} 2^{64(i-1)},$$

which is suitable for 64-bit random numbers, and effectively corresponds to a concatenation of a selection of the user-selected random numbers. Note that effectively,

each position  $q$  has successive layers of values (e.g., random numbers) “behind” it forming a sequence, and which layer  $i$  is used depends on where in the sequence the position is used.

[0126] In some examples in which these random numbers are provided along with the map, the function of the map and the random numbers may be combined such that the set of random numbers are permuted in such a way that  $\tilde{r}_{l,i} = r_{M(l),i}$  and therefore the

translation may be performed in one step as

$$P = \sum_{i=1}^k \tilde{r}_{l,i} 2^{64(i-1)}.$$

[0127] In some examples, in order to deal the variable length of user’s secrets, the secret may be padded, for example, with index values equal to 0. Similarly, blanks may be allowed in the user’s secret by allowing for a space character that maps to a particular index (e.g., 0 as well).

## 5 Setting of the User’s Random Number

[0128] Prior to generating any credentials, the a user-specific random (e.g., pseudo-random) number ( $R$ ) is determined and stored at the user’s computer. In general, this random number is relatively long (e.g., thousands or millions of bits long). In some embodiments, the random number is generated by a cryptographically strong pseudo-random number generator at the external server, and transferred to the user’s computer. In some embodiments, the number is generated at the user’s computer. This random number can be thought of as the user’s book of random numbers that is used by functions executing on the user’s computer that require generation of pseudo-random numbers. For example, one strong way of generating random numbers is to generate a sequence of offsets into  $R$  where to locate portions of the pseudo-random number, for example, where to locate successive bits of the pseudo-random number. Another stronger example is to extract a sequence of bits at each location, and then to apply a reduction function to generate a smaller number of bits (e.g., one bit), generally using a non-invertible function. A yet stronger example, which is described in the context of certain embodiments below, is to select the reduction function itself at random, thereby having three intermixed sources of randomness in the selection of the pseudo-random number: the randomness of  $R$ ; the randomness of the selection of the location sequence; and the randomness of the selection of the reduction function.

## 6 Processing to the Translated Sequence

[0129] A first step in processing the translation ( $P$ ) of the user's input is to determine a sequence of subsequences of the bits of the random number  $R$ . In some embodiments, the sequence may be chosen to depend of the service provider (e.g., on  $S_{PR}$ ) and a combination (e.g., concatenation) of the selected subsequences (e.g., one or more bits each) could be used as the credentials for that service provider. As introduced above, each subsequence may be optionally reduced by a non-invertible function (e.g., XOR, majority vote, etc.) before being combined. In some such cases, each of these subsequences is passed through a service-provider-specific reduction function to determine a portion of the binary representation of the credentials.

[0130] One particular example of this approach uses the translation  $P$ , the user's random number  $R$ , and an identifier for the service provider  $S_{prov}$  together as a seed for a random number generator. This random number generator is used to generate a sequence of offsets into the user's random number  $R$ . For example, if the binary representation of the credentials is to have  $L$  bits, a sequence of  $L$  offsets are determined this manner. At each offset, a  $(c+1)$  length sequence of the random number is extracted and applied as an argument to the service provider specific function  $F_{RS}$ . In this example, each application of this function yields a single bit of the credentials, which are concatenated to form the overall  $L$  bit quantity.

## 7 Selection of Service-Provide Specific Functions

[0131] In some embodiments, the service-provider-specific functions  $F_{RS}$  are selected at random from a space of Boolean combinations of the bits of the  $(c+1)$  length input sequence. One particular way of selecting such a random function is to generate a  $c$  bit random number, and to build a balanced AND/OR  $c$  node and  $c+1$  leaf binary tree based on the values. Specifically, each bit position in the  $c$  bit random number corresponds to a different node in the tree. Nodes with a 0 bit value perform an AND of the values, and nodes with a 1 bit value perform an OR.

[0132] In some embodiments, the  $c$  bit random numbers are selected by using a random number generator at the user's computer to generate an offset into the user's random number  $R$ , and  $c$  bits are read starting at that offset.

[0133] Referring to FIG. 4, and example of a  $(c+1) = 4$  tree generated with a random number 011 is shown. As examples, this tree produces an output a 1 with an input of 1001 and an output of 0 with an input 1100.

[0134] In some embodiments, the space of functions from which the random selection is made is restricted. For example, it is generally desirable that the service-provide function have high uncertainty (entropy) with a random input. As an extreme example, a function determined as a tree of all OR nodes would have very low entropy because the probability of its output being 1 is extremely close to one. Therefore, the random selection of a function includes a test on the probability of the output being 1 with independent inputs each with probability 0.5. If this probability is in the range  $0.5 \pm \epsilon$ , for example, for  $\epsilon = 10^{-3}$ , (or alternatively if the entropy is at least  $1 - \epsilon$  bits), the function is retained, and if not, a new random offset into  $R$  is drawn.

[0135] Because the user's random number  $R$  is already stored at the user's computer, the function is fully characterized by the offset into the random number to configure the function tree.

[0136] When a service provider requires that the credentials be changed, for example, because they require that the password be changed every two weeks, this function selection process is repeated. In situations in which some components of the credentials are to be kept constant (e.g., username) and some periodically changed (e.g., password), then multiple separate functions may be associated with a service provider, one for each part of the credentials.

## 8 Formatting of Credentials for the Service Provider

[0137] The above procedure addresses generation of a binary representation of the credentials for a service provider. In practice, many service interfaces require text passwords, for example, restricted to a limited set of characters, for example, upper and lower case letters, digits, and punctuation (e.g., a set of 72 possible characters). Some service providers may only accept capital letters, while others may require at least one punctuation character.

[0138] One way to address these constraints is to store a data characterization of the constraints, and to implement a configurable formatting function to transform the binary credentials into appropriately formatted credentials.

[0139] In some embodiments, it may be difficult to find such a translation function, and instead, the initial selection of the service-provider specific function ( $F_{RS}$ ) includes a test to ensure that the processing of the user's universal secret yields a suitable formatting. For example, if a password requires at least one punctuation character, at the time that the function is being tested according to its output probability, it may also be tested and possibly rejected according to provider-specific format requirements.

## 9 Change of a User's Universal Secret

[0140] In some situations, the user may wish to change his universal secret without having to provide new credentials to each of service providers. What is needed is that after the change in the user's secret, the translations of the selected locations in an image remain unchanged.

[0141] In examples in which the value  $P$  is determined using a set of random numbers  $r_{q,i}$ , one way to accommodate the change in secret is to rearrange these values. For example, if the user initially chose the secret "headache" such that  $Q = (8, 5, 1, 4, 1, 3, 8, 5)$ , and later wishes to change the secret to "cabbage", which would correspond to  $Q = (3, 1, 2, 2, 1, 7, 5)$ , a suitable rearrangement of the random number would have  $r_{3,1} \leftrightarrow r_{8,1}$ ,  $r_{1,2} \leftrightarrow r_{5,2}$ , ...,  $r_{5,8} \leftrightarrow r_{5,8}$ .

## 10 Extensions and Optional Features

[0142] In some embodiments the external server provides proof to the service provider that the user answered correctly to an image-based challenge. This proof can subsequently be used by the service provider to generate the new credential to the user.

[0143] The scheme can also be used to safely store and then recover already existing credentials, passwords and similar secrets (this is important as for example usernames can often not be changed once they have been created). One approach to storage and recovery of such credentials is to form and store transformation data that is used to process the service-provider binary credential to yield the existing value. One example of such transformation data is as a sequence of offsets and lengths of parts of the generated binary credential that are concatenated to form the existing value. These offsets and lengths are determined by a search at an enrollment time. If the value is changed, then the search can be performed again without changing the procedure for generating the binary credential for the service provider. In some examples, rather than forming the existing

value directly from the parts of the binary credential, the existing value is first encrypted, and after reconstruction from the parts of the credential decrypted to recover the existing value.

**[0144]** The approaches described above can also be used to obtain one-time passwords to be used on untrusted devices. For example, the user obtains a credential on a trusted device (e.g. mobile phone) to get access to a service provider from an untrusted device (an airport computer terminal). In one implementation the user can sign-in (using the universal secret mechanism) on his trusted device to a web-application of his service provider. This web application solely provides a one-time credential if the sign-in via the universal token mechanism was successful. The user can then use this credential on the untrusted device to get one-time access to the web portal of the e-service provider. In another implementation the external server negotiates one-time credentials with the service providers and then provides one of those to the user. In yet other implementation the previous techniques are combined so that only one-time credentials are used with a service provider: before the user signs out – the next sign-in credential is negotiated between the user's software agent and the service provider. This credential is then stored on the user's computing device, for example, using transformation data as described above for storing existing credentials.

**[0145]** It should also be understood that a number of different components are described above in combination, but that this components may be used independently in other security approaches. An example of this is the use of multiple layers of values “behind” each of a sequence of selection alternative. In an embodiment described above, each location  $q$  had a sequence  $r_{q,i}$  “behind” it. When the location  $q$  is selected in a sequence at the  $i^{th}$  point in that sequence, the particular  $r_{q,i}$  is used as the translation of that location. Therefore, the same location results in a different translation depending on where it is in the sequence. This same approach may be used in other applications, for example, in translation of keypad input. In an example of such an approach, each key (i.e., 1, 2, ..., 9, 0) is associated with a sequence of characters (i.e., the  $r_{q,i}$ ). When the user selects a sequence of keys, the selected characters are produced. This makes inversion of the letter sequence to a key sequence difficult. For example, a service provider may provide the letter sequences behind the keys at random whenever the keypad is provided to the user for input.

**[0146]** In another implementation the methods described above can also be used to unlock physical devices like a computer, a smartphone or a door as well as software

applications like a key-store. To unlock a device the CAPTCHA server expects to receive a proof that is in part based on a device specific seed  $S_{device}$  (e.g. the device serial number, phone number etc.). If the user, using his device, is able to send the proof to the CAPTCHA server, the CAPTCHA server sends in response a command to unlock the device. In this example the CAPTCHA server could be operating either inside or outside the user's device (e.g. a mobile phone, a tablet, etc.) and unlock the device after it received the proof. The CAPTCHA server could also be used to open other devices of the same user (e.g. by using a cable connection, WiFi connection, Bluetooth, etc.).

[0147] In a secure key-store implementation after receiving the proof, the CAPTCHA server sends the key to decrypt the credential key-store. In this scenario the CAPTCHA server can always be implemented as a remote server because access to the credential key-store is only useful if there is an active network connection.

[0148] It should also be understood that the approaches described above are not strictly limited to “credentials” in a narrow sense of the word. The user can provide various forms of identifying, verifying, and/or functional information in the same manner in which he provides his universal secret, and through one action by the user, the service provider can obtain the information and the proof that a human provided the information.

## 11 Implementation Alternatives

[0149] In some implementations, the steps described above are all carried out on the user's computing device (including smartphones and tablets), and the user's random number and characteristics of the service-provider-specific functions are all stored on that computer.

[0150] In some implementations, some steps are segregated from other parts. One way to segregate the parts is with a software based segregation. For example, the image and map may be provided from the external server is a secure “wrapper” such that it is difficult for an adversary to access the image and the processing to produce the translation  $P$ . For example, these parts may be provided in an Adobe Flash media or HTML 5 format. Another way to segregate this part of the processing is to provide the image and map to a separate device for the user, for example, to a user's smartphone, and only provide the translation  $P$  from the smartphone to the user's computer.

[0151] In some implementations, the use of the random image is not required. For example, a fixed keypad may be provided to the user and the translation  $P$  based on a

secret set of random numbers embedded in the keypad is provided based on the user's input. In some implementations, such a keypad may have programmable labels, allowing for the permutation of the locations for each use.

**[0152]** Input of the locations by the user can make use of various devices. For example, a mouse, a touch sensitive screen, a trackball, cursors, or eye tracker may be used to determine the locations entered by the user. Because a keyboard is not necessarily used, key loggers may not be effective against this type of input.

**[0153]** Various types of integration may be used. For example, some service providers may have an interface that directly communicates with software on the user's computer to accept binary credentials without formatting. In some examples, the software on the user's computer can emulate typing by the user to enter a password, or may emulate a Web form filling to send (e.g., http POST) form data that contains the credentials. In some examples, the formatted credentials are put on a software clipboard, and the user pasts them into the required parts on a service provider interface.

**[0154]** It should also be understood that a user is not limited to a single credential for each service provider. For example, the user may have multiple personalities on a social networking service, each with its own nickname and password. Furthermore, in some implementations, the generated credentials include an email alias that is forwarded by an anonymizing service to the user. For example, when a service provider requires a unique email address by which to identify and communicate with the user, the approach can be used to generate such an alias, which is then registered with an email forwarding service.

**[0155]** The credentials can include a wide range of identity attributes for the user. For example, some providers require a birthday, which could be generated for the user by the system (i.e., so that the user's real birthdate does not have to be disclosed).

**[0156]** It should also be recognized that the external server is not required. For example, the images and corresponding maps may be generated on the user's computer, on an attached device (e.g., a smartphone), or at the service provider.

**[0157]** Implementations may include software with instructions for controlling a physical or virtual machine (e.g., a Java VM), and these instructions may be stored on a machine readable medium, for example, at the user's computer or for transmission to the user's computer.

**[0158]** It is to be understood that the foregoing description is intended to illustrate and not to limit the scope of the invention, which is defined by the scope of the appended claims. Other embodiments are within the scope of the following claims.

What is claimed is:

1. A method for generating a recipient-specific credential based on a user's secret using a computing device of a user, the method comprising:
  - providing an image-based challenge for presentation to the user on an electronic display, the image-based challenge comprising a spatial arrangement of a set of perceptually obscured symbols;
  - accepting a response entered by the user via an input device and determined by the user based on the user's secret and the challenge, the response being based on a sequence of locations in the spatial arrangement;
  - determining an identifier of the recipient;
  - accessing from a data storage a user-specific predetermined random number;
  - applying a computer-implemented credential-generating procedure to combine the response from the user, the identifier for the recipient, and the user-specific random number to form the recipient-specific credential; and
  - providing the recipient-specific credential for transmission to the recipient without providing information sufficient to determine the user's secret from the transmission.
2. The method of claim 1 wherein providing the recipient-specific credential serves as providing a proof that the user that determined the response is a human knowing the user's secret.
3. The method of claim 1 wherein the recipient is a service provider and providing the credential includes providing said credential over a data communication link to the service provider.
4. The method of claim 1 wherein providing the recipient-specific credential includes providing said credential for access to a secure physical device or a software application.

5. The method of claim 1 wherein applying the credential-generating procedure includes selecting portions of a user's fixed random number and combining the selected portions, and preferably includes applying a reduction function to each portion and combining the reduced portions.
6. The method of claim 1 wherein the spatial arrangement comprises a grid based arrangement.
7. The method of claim 1 wherein the spatial arrangement comprises an irregular two-dimensional arrangement
8. The method of claim 7 wherein the irregular two-dimensional arrangement includes at least some overlapping of perceptually obscured representations of multiple of the symbols.
9. The method of claim 1 further comprising modifying the credential generating procedure according to a change of the user's secret.
10. The method of claim 1 further comprising:
  - receiving the image-based challenge and associated mapping data from an external server, and wherein applying the credential-generating procedure further comprises combining the mapping data with the response from the user to determine a mapped response, and
  - wherein combining the response from the user and the identifier for the recipient comprises combining the mapped response and the identifier for the recipient.
11. The method of claim 1 wherein the steps of claim 1 are repeatable using a plurality of different image-based challenges to the user yielding a plurality of different responses from the user determined by using the same user's secret such that the recipient-specific credential remains the same for each repetition.

12. A method for soliciting recipient-specific credentials from a user comprising, on each of a plurality of repetitions:

externally to a computing device of the user, generating an image-based challenge, the challenge comprising a spatial arrangement of a set of perceptually obscured symbols, and generating a map corresponding to the spatial arrangement of the symbols; and

providing the challenge and corresponding mapping data over a secure communication link to the computing device of the user for presentation to the user and for soliciting a sequence of locations in the spatial arrangement;

wherein the mapping data provides a relationship between the spatial arrangements of the symbols in the plurality of repetitions.

13. An image-based method for verifying a user response, the method comprising:

providing a first image-based challenge to the user;

accepting a first response to the first challenge from the user, including generating a first sequence representing the user's response;

subsequently presenting a second image-based challenge to the user; and

accepting a second response to the second challenge, including generating a second sequence representing the user's response;

wherein equality of the first sequence and the second sequence serves as a proof that the second user is a human and that the second user knew the first user's first response to the first challenge.

14. The method of claim 13 wherein the first sequence comprises a first pseudorandom number, and the second sequence comprises a second pseudorandom number.

15. The method of claim 13 comprising performing a credential generating procedure including the steps of presenting the second image-based challenge and accepting the second response.

16. The method of claim 13, further comprising providing the proof for unlocking a secured physical device and/or a software application

17. Software tangibly embodied on a computer-readable medium comprising instructions for causing a data processing system to:

provide an image-based challenge to the user, the challenge comprising a spatial arrangement of a set of perceptually obscured symbols;

accept a response from the user determined by the user based on the user's secret and the challenge, the response being based on a sequence of locations in the spatial arrangement;

determine an identifier of a recipient;

access a user-specific predetermined random number;

apply a credential-generating procedure to combining the response from the user, the identifier for the recipient, and the user-specific random number to form the recipient-specific credential; and

provide the recipient-specific credential for transmission to the recipient without providing information sufficient to determine the user's secret in the transmission.

18. A computer-implemented system for providing a recipient-specific credential based on a user's secret, the system including a software controlled component configured to:

provide an image-based challenge to the user, the challenge comprising a spatial arrangement of a set of perceptually obscured symbols;

accept a response from the user determined by the user based on the user's secret and the challenge, the response being based on a sequence of locations in the spatial arrangement;

determine an identifier of a recipient;

access a user-specific predetermined random number;

apply a credential-generating procedure to combining the response from the user, the identifier for the recipient, and the user-specific random number to form the recipient-specific credential; and

provide the recipient-specific credential for transmission to the recipient without providing information sufficient to determine the user's secret in the transmission.

217418

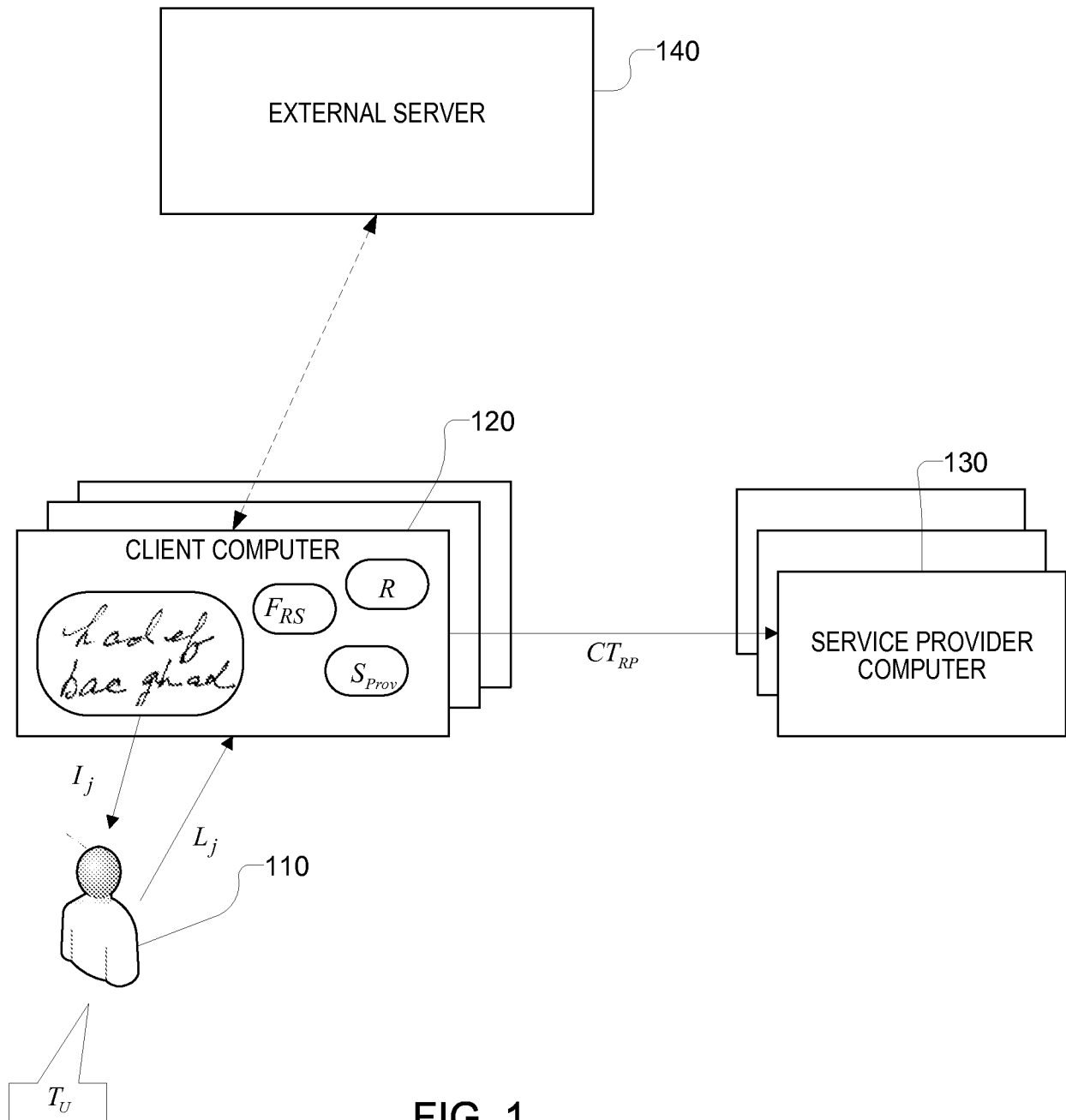


FIG. 1

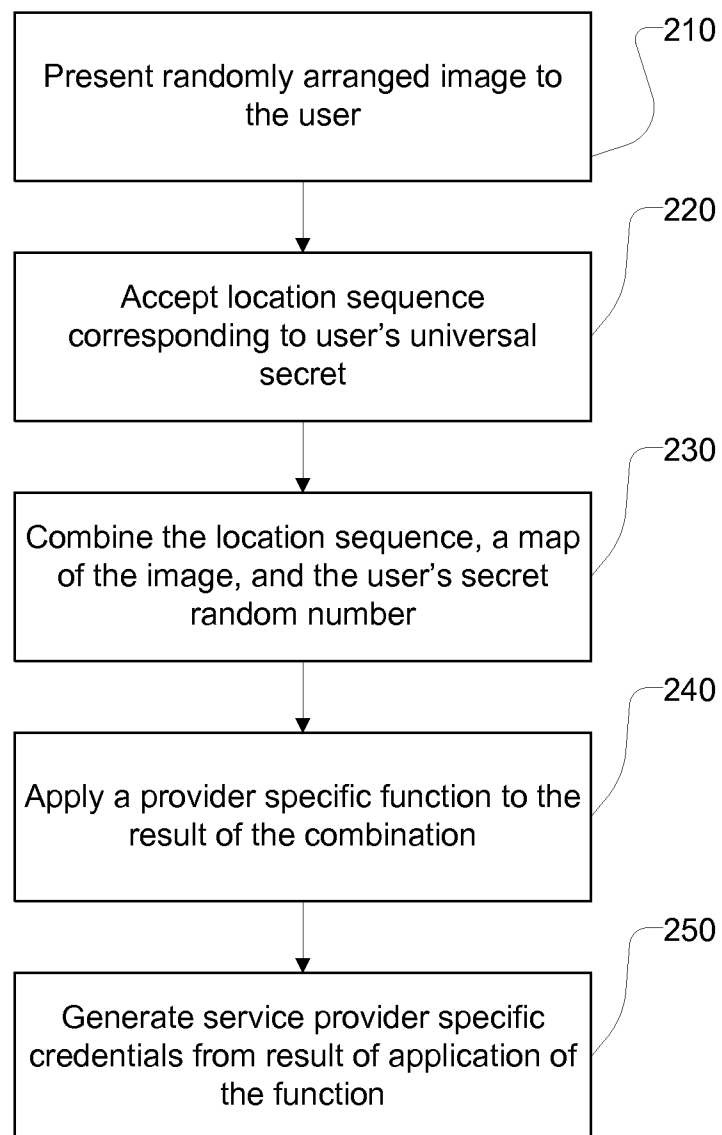


FIG. 2

$I_1 =$

|   |   |   |   |
|---|---|---|---|
| d | a | e | k |
| b | g | c | f |

$M_1 =$

|   |   |   |   |
|---|---|---|---|
| 4 | 1 | 5 | 8 |
| 2 | 7 | 3 | 6 |

FIG. 3A

$I =$

|   |   |   |   |
|---|---|---|---|
| k | a | e | f |
| b | a | e | g |

$M =$

|   |   |   |   |   |
|---|---|---|---|---|
| 8 | 1 | 4 | 5 | 6 |
| 2 | 1 | 3 | 7 | 8 |

FIG. 3B

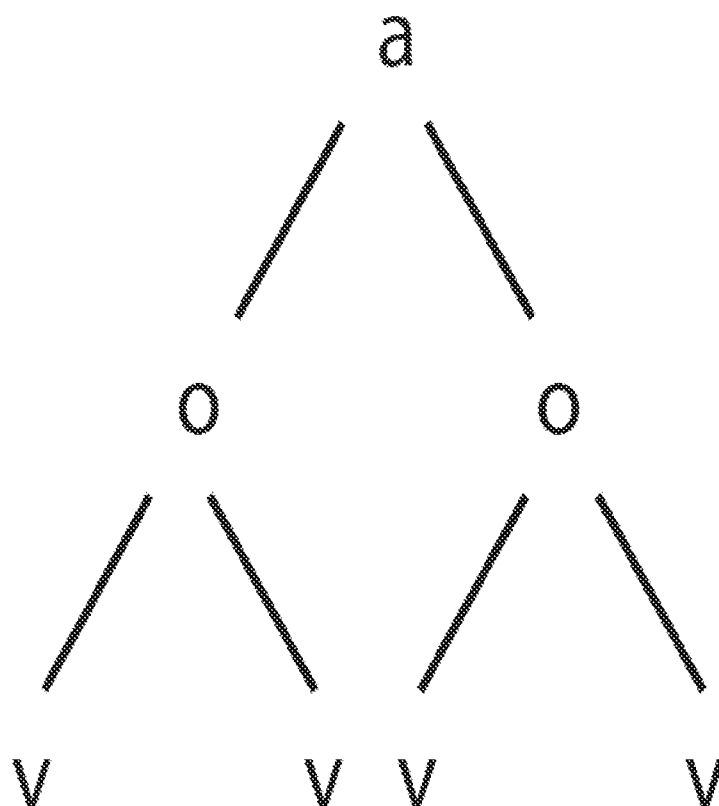


FIG. 4