(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 March 2015 (19.03.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/039084 A1

(51) International Patent Classification:
G06K 9/00 (2006.01) *H04L 9/08* (2006.01)
G06F 21/32 (2013.01)

(21) International Application Number:
PCT/US2014/055826

(22) International Filing Date:
16 September 2014 (16.09.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/878,588 16 September 2013 (16.09.2013) US
61/902,911 12 November 2013 (12.11.2013) US
14/454,148 7 August 2014 (07.08.2014) US

(71) Applicant: **EYEVERIFY** [US/US]; 1911 B. West 45th Avenue, Kansas City, KS 66103 (US).

(72) Inventors: **DERAKHSHANI, Reza, R.**; 5444 Cedar Street, Roeland Park, KS 66205 (US). **GOTTEMUKKULA, Vikas**; 3140 Woodview Ridge Drive, Apt. 205, Kansas City, KS 66103 (US). **SARIPALLE, Sashi, Kanth**; 3140 Woodview Ridge Drive, Apt. 201, Kansas City, KS 66103 (US).

(74) Agents: **ARGENTIERI, Steven, R.** et al.; Goodwin Procter LLP, Exchange Place, Boston, MA 02109 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, 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, ST, SZ,

[Continued on next page]

(54) Title: BIOMETRIC TEMPLATE SECURITY AND KEY GENERATION

(57) Abstract: Methods and systems for securing biometric templates and generating secret keys are provided. One or more images are received. Interest points are identified based on the received images, and a plurality of obfuscating data points are generated based on the interest points. An obfuscated template based on the interest points and the obfuscating data points is created and stored. A secret key can be encoded using a subset of at least one of the obfuscating data points and the interest points in the template.

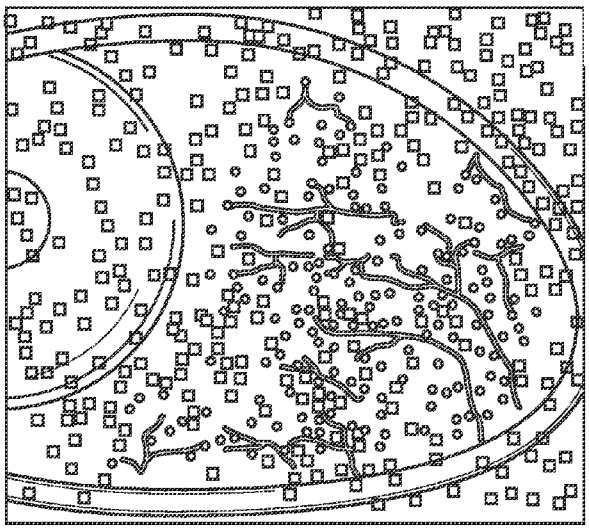

FIG. 4B

WO 2015/039084 A1

TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

BIOMETRIC TEMPLATE SECURITY AND KEY GENERATION

Cross-Reference to Related Applications

[0001] This application claims priority to and the benefit of U.S. Patent Application No. 14/454,148, filed on August 7, 2014, and entitled "Biometric Template Security and Key Generation," which claims priority to and the benefit of U.S. Provisional Patent Application No. 61/878,588, filed on September 16, 2013, and entitled "Image Detection, Authentication, and Information Hiding," and U.S. Provisional Patent Application No. 61/902,911, filed on November 12, 2013, and entitled "Detection, Authentication, and Information Hiding," the entireties of which are incorporated by reference herein.

Background

[0002] The present disclosure relates generally to biometric authentication and, more particularly, to systems and methods for securing biometric templates and encoding and decoding keys using biometric templates.

[0003] It is often desirable to restrict access to property or resources to particular individuals. Biometric systems can be used to authenticate the identity of an individual to either grant or deny access to a resource. For example, iris scanners can be used by a biometric security system to identify an individual based on unique structures in the individual's iris. Biometric data captured from an individual, such as during an enrollment process, can be stored as a template that is used to verify the identity of the individual at a later time. Templates can be stored, for example, remotely on an authentication server or locally on a device having the ability to capture biometric readings, such as a mobile phone with a camera. However, maintaining a template in its original form or in a form from which the original template can be derived creates a risk that the security of the template will be compromised.

Brief Summary

[0004] Systems and methods for securing biometric templates and encoding and decoding keys using biometric templates are disclosed. In one aspect, a computer-implemented method comprises: receiving one or more images; identifying a plurality of interest points based on the

received images; generating a plurality of obfuscating data points based on the interest points; creating an obfuscated template based on the interest points and the obfuscating data points; and storing the obfuscated template. Other embodiments of this aspect include corresponding systems and computer programs.

[0005] In one implementation, the obfuscating data points are generated such that a spatial distribution of the interest points and a spatial distribution of the obfuscating data points are substantially similar.

[0006] In another implementation, the method further comprises associating one or more real descriptors with each interest point, wherein each real descriptor describes one or more localities surrounding the corresponding interest point.

[0007] In a further implementation, the method further comprises discarding a record of which points in the obfuscated template are the interest points.

[0008] In yet another implementation, the method further comprises encoding a key using a subset of at least one of the obfuscating data points and the interest points. Each point in the subset can be determined based on a different one of the interest points.

[0009] In another implementation, the images comprise biometric imagery. The images can comprise images of a region of an eye, each eye region image comprising a view of a vasculature of the respective eye region. The interest points can comprise vascular interest points.

[0010] In one implementation, the method further comprises associating one or more synthesized descriptors with each obfuscating data point, wherein each synthesized descriptor comprises a statistical similarity to the real descriptors.

[0011] In another implementation, the method further comprises: receiving one or more second images; identifying a second plurality of interest points based on the received second images; creating a verification template based on the second plurality of interest points; comparing the verification template with the obfuscated biometric template to identify a plurality of matching interest points; and authenticating a user based on the matching interest points. The comparing can comprise identifying the matching interest points based on one or more of the real and synthesized descriptors.

[0012] In a further implementation, the method further comprises reducing a dimensionality of the real descriptors and the synthesized descriptors. The comparing can include identifying the matching interest points based on one or more of the reduced dimensionality descriptors.

[0013] In a further implementation, the method further comprises isometrically scrambling the real descriptors and the synthesized descriptors. The comparing can further comprise identifying the matching interest points based on one or more of the scrambled descriptors.

[0014] In yet another implementation, the method further comprises decoding the key based on at least a subset of the matching interest points.

[0015] The details of one or more implementations of the subject matter described in the present specification are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

Brief Description of the Drawings

[0016] In the drawings, like reference characters generally refer to the same parts throughout the different views. Also, the drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the implementations. In the following description, various implementations are described with reference to the following drawings, in which:

[0017] FIG. 1 depicts a diagram of a system for biometric template security and key generation according to an implementation.

[0018] FIG. 2 depicts a method for securing a biometric template and encoding/decoding a secret key according to an implementation.

[0019] FIG. 3 depicts an ocular image with example vascular interest points.

[0020] FIG. 4A depicts the vascular interest points of FIG. 3 with embedded obfuscation data points.

[0021] FIG. 4B depicts the obfuscated data points from FIG. 4B superimposed on the eye image of FIG. 3.

[0022] FIG. 5 depicts the vascular interest points and obfuscating data points of FIG. 4A with a subset of tagged points.

Detailed Description

[0023] Distinctive features of an individual's visible vasculature in the whites of the eyes can be used to identify or authenticate the individual. For example, images of the white of a user's eye can be obtained and analyzed to compare features of the eye to a biometric template in order to authenticate the user and grant or deny the user access to a resource. Implementations of solutions for imaging and pattern matching the blood vessels in the white of the eye and for feature extraction and matching are described in U.S. Patent No. 8,369,595, issued on February 5, 2013, and entitled "Texture Features for Biometric Authentication," and U.S. Patent Application No. 14/274,385, filed on May 9, 2014, and entitled "Feature Extraction and Matching for Biometric Authentication," the entireties of which are incorporated by reference herein.

[0024] For example, the unique structure of an individual's visible vasculature can be reflected in texture features of images of the white of the individual's eye. Images can be segmented to identify regions on the white of the eye for texture analysis, and a set of filters can be applied to determine descriptors of the texture features of the individual vasculature in these regions. A vector of descriptors derived from filter outputs can be assembled into a descriptor vector. Then, during an authentication or identification operation, the descriptor vector determined for a user can be compared to a corresponding descriptor vector from a stored biometric record for an enrolled individual to determine the likelihood of a match between the user and the enrolled individual.

[0025] Various implementations of the template security and key generation techniques described herein are based on steganographic obfuscation of a biometric template using a large or sufficient number of "chaff" or indistinguishable noise elements. A subset of the chaff elements, which are identified upon successful verification in a device-specific scrambled space, is utilized to solve a system of equations that yields an encoded secret. These tokens are high entropy, revocable, and reveal nothing about user's biological traits.

[0026] FIG. 1 illustrates one implementation of a localized system for generating secure biometric templates, performing user verification, and encoding and decoding secret keys based on the biometric templates. A user device 100 can include an image sensor 130, processor 140,

memory 150, biometric hardware and/or software 160, and a system bus that couples various system components, including the memory 150 to the processor 140. User device 100 can include, but is not limited to, a smart phone, smart watch, smart glasses, tablet computer, portable computer, television, gaming device, music player, mobile telephone, laptop, palmtop, smart or dumb terminal, network computer, personal digital assistant, wireless device, information appliance, workstation, minicomputer, mainframe computer, or other computing device that is operated as a general purpose computer or a special purpose hardware device that can execute the functionality described herein.

[0027] Biometric hardware and/or software 160 includes an image processing module 162 for performing operations on images captures by image sensor 130. For example, image processing module 162 can perform segmentation and enhancement on images of the eye of a user 110 to assist in isolating vascular structures. Template security module 166 creates biometric templates based on the vasculature imagery and performs various obfuscating and scrambling operations on the templates, as described herein, to increase template security while maintaining usability. Verification module 174 validates the identity of a user 110 by performing matching operations between a biometric verification template formed upon capturing a biometric reading and a previously stored enrollment template. Key module 178 can encode a secret key for the user 110 based on a biometric enrollment template and decode the key upon successful verification of the user's identity using a verification template.

[0028] Implementations of the system described herein can use appropriate hardware or software; for example, the system can execute on hardware capable of running an operating system such as the Microsoft Windows® operating systems, the Apple OS X® operating systems, the Apple iOS® platform, the Google Android™ platform, the Linux® operating system and other variants of UNIX® operating systems, and the like. The system can include a plurality of software processing modules (e.g., image processing module 162, template security module 166, verification module 174, and key module 178) stored in a memory 150 and executed on a processor 140. By way of illustration, the program modules can be in the form of one or more suitable programming languages, which are converted to machine language or object code to allow the processor or processors to execute the instructions. The software can be in the form of a standalone application, implemented in a suitable programming language or framework.

[0029] Additionally or alternatively, some or all of the functionality can be performed remotely, in the cloud, or via software-as-a-service. For example, certain functions (e.g., image processing, template creation, template matching, etc.) can be performed on one or more remote servers or other devices that communicate with user devices. The remote functionality can execute on server class computers that have sufficient memory, data storage, and processing power and that run a server class operating system (e.g., Oracle® Solaris®, GNU/Linux®, and the Microsoft® Windows® family of operating systems). Communication between servers and user devices can take place over media such as standard telephone lines, LAN or WAN links (e.g., T1, T3, 56kb, X.25), broadband connections (ISDN, Frame Relay, ATM), wireless links (802.11 (Wi-Fi), Bluetooth, GSM, CDMA, etc.), for example. Other communication media are contemplated. The network can carry TCP/IP protocol communications, and HTTP/HTTPS requests made by a web browser, and the connection between the user devices and servers can be communicated over such TCP/IP networks. Other communication protocols are contemplated.

[0030] Method steps of the techniques described herein can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. Method steps can also be performed by, and the modules can be implemented as, special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application-specific integrated circuit). Modules can refer to portions of the computer program and/or the processor/special circuitry that implements that functionality.

[0031] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. The essential elements of a computer are a processor for executing instructions and one or more memory devices for storing instructions and data. Information carriers suitable for embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. One or more memories can store instructions that, when executed by a processor, form the modules and other components described herein and perform

the functionality associated with the components. The processor and the memory can be supplemented by, or incorporated in special purpose logic circuitry.

[0032] The system can also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules can be located in both local and remote computer storage media including memory storage devices. Other types of system hardware and software than that described herein can also be used, depending on the capacity of the device and the amount of required data processing capability. The system can also be implemented on one or more virtual machines executing virtualized operating systems such as those mentioned above, and that operate on one or more computers having hardware such as that described herein.

[0033] It should also be noted that implementations of the systems and methods can be provided as one or more computer-readable programs embodied on or in one or more articles of manufacture. The program instructions can be encoded on an artificially-generated propagated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus for execution by a data processing apparatus. A computer storage medium can be, or be included in, a computer-readable storage device, a computer-readable storage substrate, a random or serial access memory array or device, or a combination of one or more of them. Moreover, while a computer storage medium is not a propagated signal, a computer storage medium can be a source or destination of computer program instructions encoded in an artificially-generated propagated signal. The computer storage medium can also be, or be included in, one or more separate physical components or media (e.g., multiple CDs, disks, or other storage devices).

[0034] Referring to FIG. 2, in one implementation, a method for securing a biometric template starts by receiving images of a user's eye, eyes, and/or one or more regions thereof (STEP 202). The image(s) can be captured using a device 100 having an image sensor 130, e.g., a phone or tablet with a front-facing camera. If multiple images are received, a single image can be automatically selected based on its suitability for biometric identification, or some or all of the images can be automatically selected and averaged to produce a single combined image (STEP 206). The image region containing the sclera, or white of the eye, is segmented, sharpened, contrast enhanced, and/or filtered in several scales of blue-green layers,

by image processing module 162, to provide an optimal depiction of vascular patterns visible in the white of the eye (STEP 212).

[0035] In STEP 218, based on the depiction of the vascular patterns, template security module 166 identifies vascular points of interest and, in STEP 222, the module 166 associates a series of image descriptors in each locality with the corresponding vascular point of interest to create a location-descriptor structure for each point of interest. At this stage, the eye image(s) can be discarded (STEP 226). The resulting set of vascular points of interest and their associated local image descriptors form a basic biometric template (STEP 230). If the template is intended for enrolling the user, the template can be saved locally on the device 100 in a private and secure manner (e.g., in memory 150), as described below.

[0036] To secure the biometric template, the template security module 166 "hides" location-descriptor structures within a number of generated "chaff" elements, or obfuscating data points, that can be similarly structured and statistically indistinguishable from actual vascular points of interest (STEP 234). Before discarding all records of the chaff vs. non-chaff (i.e., genuine vascular point of interest) elements in STEP 242, each vascular point of interest "tags" a chaff point (or another vascular point of interest) (STEP 238). Specifically, the key module 178 inputs a vascular point of interest into a secure one-way function, which designates as output a chaff point (or vascular point of interest) to be tagged. These tagged points can be used by the key module 178 to absorb and encode linear projections of a long random key (STEP 250) as well as to decode a key upon successful verification of a user's identity, as further described below.

[0037] These chaff-delegated operations further decouple various functionalities (such as surrogate biometric verification and key generation) from the genuine template elements for added privacy, security, and revocability. The template security module 166 further secures the chaff-obfuscated template in STEP 246 by scrambling the descriptors by, for example, statistical de-correlation and normalization, and/or device-specific isometric salting and dimension reshuffling, thereby ensuring that no biometrically derived information is revealed, especially if transmitted off the device 100. The verification module 174 can perform biometric template matching during identity verification in this unique device-specific and scrambled space, adding yet another layer of security, privacy, and revocability to the local matching and key generation routines. In STEP 254, the chaff-obfuscated, scrambled

descriptor template is stored locally on the device (or, in other implementations, the template is stored remotely).

[0038] During verification of a user's identity, the same or similar image capture, segmentation, and enhancement steps are carried out by the image processing module 162. Similarly, vascular interest points are found and their local descriptors are calculated and then scrambled by the template security module 166 (STEP 258) using the unique device-and-software-specific signature used during enrollment, thereby creating a verification template. (STEP 262). This ensures that enrollment and verification can take place only on the same device and software instance. The matching process, in STEP 266, completed in the scrambled space by the verification module 174, identifies a minimum number of genuine vascular interest points by comparing the verification template with the obfuscated template in case of a successful genuine verification. The identified genuine vascular interest points in turn reveal a large-enough subset of the information-carrying chaff points tagged earlier in the enrollment process (STEP 268). This minimum number of genuine points and, thus, tagged chaff points, is of the same order as the key-encoding system of equations. The key module 178 can then use information from the tagged chaff points to solve for the system of equations and obtain in the decoded key (STEP 272). In one implementation, the key is stable, 512 bits long, and has an entropy of at least 64 bits.

[0039] It is to be appreciated that, although the various systems and methods presented herein utilize biometric eye imagery and interest points derived from visible vasculature, other implementations and applications of the disclosed techniques are contemplated. For example, in other implementations, features and/or points of interest are identified in other biometric image data, such as fingerprint or facial scans. Similar imaging processing procedures can be performed to enhance and isolate the interesting features/points in the imagery and, once the features/points are identified, the same or substantially similar obfuscation, scrambling, verification, and or key encoding/decoding techniques as described herein can be applied. It is of further note that the various systems and methods presented herein need not be used in conjunction with biometric imaging and authentication. Rather, the techniques disclosed herein are equally applicable to other types of images, video frames, and the like.

Enrollment

Image Capture

[0040] In one implementation, one or more eye images (and/or eye region images) are captured with an image sensor at an image quality suitable for the image processing functionality described herein, such as 720p, 1080p, or equivalent/higher resolution. The image sensor can be, for example, a one megapixel or better image sensor such as the front-facing camera generally found in cellular phones and tablets. The user's eyes can be detected using for instance Viola-Jones methods, and the user's gaze direction can be detected, all in real time. Upon detection of a stable gaze and at least one eye, a stack of images of the user's eye(s) are captured.

[0041] Spatially registered images from the input stack are averaged to lower sensor noise, and the best resulting averaged shots are selected using a reference-free image quality metric. In low or no light conditions, the backlighting of the device screen plus multi-frame noise reduction due to the aforesaid averaging enables the biometric processing operations described herein to be carried out. In one example, a number of continuous image frames (e.g., three, four, five, or more) that do not exceed an acceptable amount of variance (e.g., due to motion and blink) are registered and averaged in real time. Image stacks can be ranked using a Laplacian-of-Gaussian (LoG)-based quality metric (standard deviation of the sharpened image minus the original), and the top n are reserved for further processing (e.g., up to two for verification, up to four to six for enrollment).

Segmentation and Enhancement

[0042] Following image capture (and averaging, if performed), selected images can be color processed to better reveal blood vessels in the green-blue spectra, and segmented to delineate the white part of the eye, henceforth referred to as a region of interest (ROI). In one implementation, images are segmented by fitting multiple conic section curves to eyelids and corneal limbus boundaries. Segmentation validity is checked (e.g., the mask should be at least 40% of the bounding box of the ROI). A series of vascularity-enhancing image filtering, sharpening, and adaptive contrast manipulations provide the improved image needed for more specific biometric templates. For example, the green (red-free) layer of the images can be enhanced using contrast limited adaptive histogram equalization (CLAHE) of the LoG times

the original, as well as a specially tuned bank of even Gabor filters. A series of multi-scale and specially filtered adaptations of the enhanced image can then be used for the next step.

Interest Point Detection and Feature Extraction

[0043] For each ROI, locations (x_i, y_i) of interest points are identified, a number typically ranging between 100–400 depending on the image quality. FIG. 3 depicts an example ocular image with identified points of interest 320 of the vasculature 315 of the eye 300. The interest points 320 can be identified using a vascular point detector such as that described in U.S. Application Serial No. 14/274,385, filed on May 9, 2014, and entitled "Feature Extraction and Matching for Biometric Authentication," the entirety of which is incorporated by reference herein. Other ways of detecting interest points are possible.

[0044] Next, a set of $\vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d$ descriptor vectors that statistically (but not exactly or uniquely) describe the local image patches around vascular interest point locations (x_i, y_i) are computed. Image patch descriptor examples include, but are not limited to, Speeded Up Robust Features (SURF), (histograms of) multi-radii extended pattern local binary patterns (H LBP), and (histograms of) multi-radii extended pattern center symmetric local binary patterns (H CS LBP). For each ROI, the naive (unprotected) biometric template, T_{VPD}, which includes detected vascular interest points VPD, is then defined as:

$$T_{VPD} = \{t_i\}, t_i = \left[(x_i, y_i), \vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d\right], i = 1,2, \ldots n(T_{VPD})$$

[0045] At the time of verification, the stored enrollment template for the claimed identity is matched against the presented verification template. In one implementation, if the similarity score is above a preset threshold, which also entails pairing of certain minimum number of elements across enrollment and verification templates, then the claimant is accepted and a match decision is issued. Note that the eye images can be immediately discarded after creation of the template, and only the enrollment templates stored.

Obfuscation and Encoding

Chaff Points Added and Tagged

[0046] In one implementation, an initial step in securing a biometric template includes hiding the to-be-stored enrollment template elements from T_{VPD} among a large number of artificial synthesized elements that appear identical or substantially similar to the genuine vascular points of interest. These synthesized elements are referred to herein as "chaff." In one

implementation, the number of chaff is approximate three to seven times the number of real template elements $n(T_{VPD})$. However, other multiples are contemplated. For example, higher chaff densities can provide for even higher levels of obfuscation, albeit at the expense of an added computational footprint.

[0047] Chaff elements can be inserted by an algorithm that ensures spatial distribution of all data points, chaff and non-chaff (i.e., actual vascular interest points), are uniform or following the same or substantially similar pattern or distribution as the vascular interest points. In one example, local spatial densities of (x_i, y_i) are about the same down to a given area granule or tile, and descriptor contents or spatial relationships do not reveal chaff from real non-chaff (actual vascular interest points) within a spatial grain. FIG. 4A depicts the vascular interest points (circles) from FIG. 3 embedded within chaff points (squares) for an approximate 3x chaff to non-chaff placement. FIG. 4B is a visualization of the obfuscated points from FIG. 4A superimposed on the original eye image from FIG. 3. Note, however, that the eye image can be discarded prior to this obfuscation stage and right after calculating T_{VPD}.

[0048] Each template point t_i, whether real (vascular interest point) or synthesized (chaff), can include two types of information: location (x, y) and patch statistics V. Spatial uniformity of the chaff-infused template for non-distinguishability of chaff data points can be achieved by several means. In one implementation, the following two-step chaff (x, y) location generation process is used. In Step 1 (coarse chaff placement): Given a typical tiling over the spatial span of the enrollment template (e.g., 4 x 5), start with placing the first portion of the chaff, needed to equalize the average of total template points (chaff and non-chaff) per tile, a goal number that is larger than the maximum number of VPD points in any tile. Continue until reaching about 50% of the vascular interest point VPD + chaff point density goal per tile. Use an initial minimum distance requirement (e.g., three pixels) among all data points (chaff or vascular interest point) for this coarse chaffing step. In Step 2 (fine chaff placement): Continue with inserting the rest of the chaff, reducing minimum distance threshold (e.g., to 1 pixel), until achieving 100% of the desired uniform vascular interest point VPD + chaff point density goal per tile.

[0049] In one implementation, the low end of (x, y) ranges for data point locations created by 1.2 MP cameras is about 80x100 pixels +/- 20. It should be noted, however, that this number can change based on the field of view of the camera, subject distance, and other

factors. The details of this method and other alternative methods are described below in the section entitled, "Sample Chaff Generation and Tagging Function Implementations."

[0050] Following chaff placement, chaff descriptor vectors $\vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d$ are synthesized to be similar to descriptors associated with genuine vascular interest points VPD. That is, the contents of the descriptors that are assigned to chaff points are formed to be statistically similar and indistinguishable from those derived for real interest points VPD. The aforementioned indistinguishability of chaff descriptors from real vascular descriptors can be achieved in various manner. In one implementation, to generate various chaff descriptors during enrollment, a small random circular shift and additive noise is applied to real vascular descriptors to get chaff descriptors that follow the same statistical distribution as those of their real counterparts. These features can later be "scrambled," as described below.

[0051] At the time of enrollment template creation, chaff points and their synthesized descriptors are structured as the real, VPD spanned part of the template:

$$T_{CHF} = \{t_i\}, t_i = \left[(x_i, y_i), \vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d\right], i = 1,2, \ldots n(T_{CHF})$$

The chaff-infused obfuscated template is thus in form of an (unordered) set given by:

$$T_A = T_{VPD} \cup T_{CHF}$$

[0052] A "tagging" function is a one-way mapping of one template element to another. Specifically, a tagging function can be used to find or "tag" a template point in a chaff-obfuscated template given any other data point from that template. In one implementation, a tagging function f_T satisfies the following properties: (1) its domain contains $\left\{\left[(x_i, y_i), \vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d\right]\right\}$; (2) it is nontrivial and many-to-one (or otherwise non-invertible or with no known or practical inverse) (e.g., based on SHA512 hash functions, which can be used in scrambling and encoding/decoding states, as well as for tagging); and (3) over the given enrollment template, the range minimally intersects with the set of vascular interest points (i.e., there is minimal self-tagging within the vascular interest point subset of the template):

$$\frac{n(f_T(VPD) \cap VPD)}{n(VPD)} \ll 1$$

[0053] Current and alternative implementations of such functions are described in the section entitled, "Sample Chaff Generation and Tagging Function Implementations." Given the nominal values for the VPD portion of the template, these tagging functions generally tag

about one point at their output per each vascular interest point at their input. In one implementation, tagging functions can be used to tag a key-encoding subset of the chaff (see below), and a trust-server-signature-carrying subset of the chaff (see "Trust Server Functionality," below). These two tagging functions can include a small overlap in their ranges.

[0054] A tagging function f_K, such as described herein, can be used to find the template points T_K into which the real T_{VPD} part of the template map (mostly chaff, given the third property of tagging functions), so that $T_K = f_K(T_{VPD})$. FIG. 5 depicts the real points (circles) and obfuscated points (squares) from FIG. 4A, with a subset of tagged points (solid circles and squares). Optionally, another similar (but not identical) subset of template can be tagged using a second tagging function f_S, different from f_K by virtue of difference in design or meta parameters, to yield $T_S = f_S(T_{VPD})$, which can be used for optional trust server functionality.

[0055] T_K can then be used to encode a secret key. Note that T_{VPD} is known only during the enrollment process and prior to its obfuscation in T_{CHF}. No record of T_{VPD} is kept, and only a subset of T_{VPD} is revealed during a successful genuine biometric verification.

Scramble Descriptors

[0056] In one implementation, to reduce dimensionality, improve the accuracy and speed of matching, and to de-correlate and thus further "flatten" and strengthen the uniformity of chaff-obfuscated enrollment templates, the loadings for principal component analysis (PCA) projections of different feature vectors $\{\vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d\}, i = 1,2, \ldots, n(T_A)$ are pre-calculated using a large representative training set and stored. Next, the descriptors in chaff-infused templates are reduced to a fraction of their original length, e.g., about 30%, while keeping a significant (e.g., more than 80%) of their original explained variations using Scree graph analysis. Optional variance normalization of PCA projections after mean subtraction creates whitened stored template that has a diagonal normalized covariance matrix across all its features. Given the properties of PCA, the result preserves most Euclidean distance information needed for matching. Finally, the scrambling process can use a hash of different software and device hardware signatures to seed (a) a salting process to alter the PCA-shortened features using a SHA512-derived bias vector added to all descriptors (both for enrollment and verification templates, and prior to saving for enrollment templates), and (b) seed-modulated

reordering of the coordinates of the resulting feature vectors (prior to saving for enrollment templates).

[0057] Note that, in addition to the lossy PCA projection, both (a) and (b) preserve the Euclidean distance, enabling matching to proceed in a scrambled space tied to the user's device. This is a particularly notable attribute because matching in an isometric (distance-preserving) and revocable surrogate space is crucial to secure and private biometric pattern matching, and leads to two-factor authentication because both the device and the genuine user will be needed for the aforesaid biometric authentication to succeed. Not only is it unnecessary to de-scramble descriptors during matching (and thus avoid risk of exposure), but a unique software-revocable and device-specific scramble space can be spanned for each installation of the biometric authentication application.

Key Encoding

[0058] One implementation of the augmented template structure for key generation (i.e., computing a secret key as a byproduct of a biometric match) will now be described. Assume that there is a system of linear equations of order k, whose coefficients are considered a secret numerical $\vec{S}, (\dim(\vec{S}) = k)$. During verification, k is the minimum number of vascular interest points found during a successful matching process between enrollment and verification templates of a genuine user, operating at empirical 0% false accept ratio (FAR) threshold (i.e., a decision threshold that does not admit any impostors using the largest biometric eye reading dataset available). A system of linear equations can be used to encode the key, as an ordered set of data points is not required to solve for that key (the key can be encoded directly into a system of linear equations exactly solved given the high sensitivity and specificity of eye vein pattern matching arising from their complex, intricate, and high entropy structures).

[0059] Thus, a set of data points $D = \{d_i\}, n(D) \geq k$ is needed to uniquely solve a system of linear equations to retrieve the encoded secret numerical vector, $\vec{S}$, made possible by a successful genuine verification leading to recovery of k equations needed to solve for k unknowns making up the key (to further enforce a standard length and strength in terms of key bit sequence flow, SHA512 can be applied to the operational version of this key to have a pattern-unpredictable 512-bit private key sequence). Note that the order of recovered matched points and thus equations does not matter. The key generation information is inter-dispersed across a subset of augmented (with descriptor projection values for function-fitting) elements of

the chaff-obfuscated enrollment template, henceforth referred to as T_{AK}, and defined as:

$$T_{AK} = \{t_i\}, t_i = \left[(x_i, y_i), \vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d, \vec{Y}_i^1, \vec{Y}_i^2, \ldots \vec{Y}_i^d\right], i = 1,2, \ldots n(T_A)$$

where (x_i, y_i) are the locations of interest and chaff points i in T_A. The augmented part of the template is $\vec{Y}_i^1, \vec{Y}_i^2, \ldots \vec{Y}_i^d$, a collection of vectors similar to $\vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d$ in dimensionality, but with each element of Y being the projection of the corresponding element from V using a k-way vectorizing function (see "Vectorizing Functions," below) and then inner product operation with an $\vec{S}$, providing the right-hand side of the earlier mentioned system of equations (notice that each element of $\vec{V}$encodes a different $\vec{S}$). The (collection) of secret vector $\vec{S}$ is later retrieved upon successful biometric authentication by a genuine user. The aforesaid process is described through the following encoding and decoding steps, which are enabled by tagging and vectorizing functions to enhance security and privacy while maintaining numerical stability.

Encoding Process

[0060] In one implementation, the key generation functionality is based on a successful genuine accept (true positive verification) producing at least k matched points between enrollment and verification templates, even when obfuscated by indistinguishable chaff. Thus, if a system of k equations with k unknowns is built upon this matching process, where k data points for the equation can practically only be known through successful genuine matching, then the equation and thus the key can be uniquely solved only if a true match occurs.

[0061] Note that k is a function of image quality and matcher strength, and can be increased with improvements to either, or by matching multiple ROI/templates (from enrollment and verification banks) with a same encoded key in multiple enrollment templates and taking the union of the found tagged points before solving the equation to recover the secret key.

[0062] In one example, $k = 40$ for single glance, single comparison, 2-ROI matching, given observations over collected datasets at empirical FAR = 0 threshold. Matched points are template entries that are selected after being compared with their corresponding verification counterparts through proximity of their descriptors and after rejection of outliers using a random sample consensus (RANSAC) with an affine transform hypothesis (or similar). No false accepts occur if the number of such matched template entries are k or higher (i.e., the

generated or released secret is unique to each unlocking user at that threshold within the bounds of the observations). For less sensitive applications, if one assumes that the matcher is not broken or compromised, a smaller k can be used to reduce key generation false rejection ratio, assuming that a false acceptance event at key generation stage will not proceed given that the matcher is rejecting the request (that is, in cases where the match score indicates a match while the number of matched points is slightly less than k, assuming that the match score has higher sensitivity and specificity than the number of points matched).

[0063] Continuing with key generation, at the time of chaff-obfuscated template creation, $T_A = T_{VPD} \cup T_{CHF}$ is produced (there can be small overlaps between T_{VPD}, T_S, and T_K). T_K subset of chaff, tagged by $f_K(T_{VPD})$, is provided to a function (e.g., a linear projection) that encodes one or more (random) secret keys $\vec{S}$ using contents of T_K and a system of linear equations. Assume that there is (about) one tagged point $t_i \in T_K$ per each tagging vascular element from VPD subset, $i = 1,2, \ldots, n(VPD)$. Because the key-encoding process can be similar for all the different descriptor sets (e.g., SURF, histograms of LBPs, and so on), the process can be demonstrated for one generic type of such features.

[0064] Assume the simplified yet to be augmented form of $T_A = T_{VPD} \cup T_{CHF}$ (using a single type of descriptor and chaff-infused), T, is as follows:

$$T = \{t_i\}, t_i = \left[(x_i, y_i), \vec{V}_i\right]$$

If the dimensionality of V_i is D, then one can encode any key matrix W, composed of $D \times k$ numbers (real or otherwise, where each row can be considered as a different key vector $\vec{S}$) as the matrix of secret keys $W_{D\times k} = [W_{jd}]$ as follows. Each scalar element of the VPD subset of feature vectors V_i in T_A, $v_{i,d}, d = 1,2, \ldots D, i = 1,2, \ldots n(T)$ is vectorized (split), using a non-obvious and non-invertible vectorizing function, into k specific values. The vectorizing (splitter) function thus performs the following:

$$\vec{X} = \vec{\varphi}(x), \quad \dim(x) = 1, \dim\left(\vec{X}\right) = k$$

[0065] A lighter version without a vectorizing function, where a key vector of max dimensionality D is directly encoded as a linear combination of each $\vec{V}_i$, assuming $D \geq k$ (and thus one Y_i per each augmented $\vec{V}_i$, rather than D), is also possible. However, the matrix of k-juxtaposed $\vec{V}_i$ for the decode process should not be singular.

[0066] Finally, a corresponding $y_{i,d}$ is associated and added to the input $v_{i,d}$ encoding $\vec{W}_d$ (row d of the secret key matrix W with a length of k) by:

$$y_{d,i} = f_{encode}(\vec{W}_d, v_{d,i}) = \vec{W}_d \cdot \varphi(v_{d,i})$$

[0067] The aforesaid sequence is repeated for all the D dimensions of the descriptor/key set $\vec{V}_i, \vec{W}_d$ and all the $n(T_K)$ f_K-tagged elements of the template for key generation to get $\vec{Y}_i$-augmented T_K: $\{[(x_i, y_i), \vec{V}_i, \vec{Y}_i]\}$. Next, W is altered (minimally, by adding small noise) to arrive at W_c, and similar applications are made to the f_k-untagged portion of the template to get a complete $\{y_{i,d}\}$-augmented T in a way that its components, including $y_{i,d}$ completely blend together across tagged, untagged, chaff, and vascular elements. Multiple fake W's can be produced, each applied to a subset of T_{AK} (subsets with $n(T_{VPD})$ number of elements recommended for added security).

[0068] Note that the above process is noninvertible, i.e., given $y_{i,d}$, one cannot get back to $v_{i,d}$ and $\vec{W}_d$ (for one thing, the calculation of $\vec{\varphi}(x)$ and $y_{d,i}$ are many-to-one functions and noninvertible and, further, that until the time of positive genuine verification one does not know which subset of T_{AK} contains the tagged and thus W-encoded data to solve for it).

[0069] In one observational example, within datasets with a threshold of $k = 40$ (single gaze, single comparison, 2 ROI), a false accept was unable to be produced. That is, within observational limits, no two different users generated the same key and, thus, the entropy is seemingly equal to the key length. However, this does not imply that for a much larger database of users a collision (false accept) at $k = 40$ could not happen, in which case one may simply increase k (albeit at the expense of a possibly higher false reject ratio given the higher threshold). As for empirical false acceptance ratio evaluation, using all the 7 billion population of the earth, one can experimentally guarantee the uniqueness of a biometric key space for up to about only 36 bits ($\log_2(7 \times 10e9) = 36.03$). Given the above, at some arbitrary strict threshold for k, the level of chaff-induced obfuscation of T_{AK} will eventually constitute the limit for key entropy.

[0070] Encoded keys can be changed, replaced, or revoked in multiple different ways, from changing the contents of W or the corresponding $\{Y_i\}$ to changing vectorizing functions. Tagging functions and chaff contents can also be changed to achieve the aforesaid. Some of these methods are applicable at the time of enrollment, whereas others can be applied at any

time. For instance, at any time, each vector key $\overrightarrow{W_d}$ can be revoked or changed in a private, secure, and convenient way by perturbing at least $n(T_A) - k + 1$ elements of $y_{d,i}$ across i, e.g., by adding a small noise vector to all the d[th] elements of $\{Y_i\}$. This changes the solution $\overrightarrow{W_d}$ without revealing its new or old contents, which can be only known upon discovering at least k elements of T_k made possible by a successful verification of the genuine user. In the case of multiple enrollment templates and ROIs, the same key W can be encoded in each template so that the released key from the best/combined comparison(s) remains the same. Note that since the tagged template elements are different across these enrollments, the corresponding $\{V_i, Y_i\}$ will also be different and thus there is no attack vector arising from comparing multiple templates with the same encoded W.

Verification and Decoding

[0071] In one implementation, biometric template verification begins with image capture, segmentation and enhancement, interest point detection and feature extraction, and descriptor scrambling in the same or substantially the same manner as described above with respect to the enrollment process. On the other hand, adding and tagging chaff and key encoding apply only to the enrollment process.

Matching

[0072] During matching, the claimed identity, as represented by the stored enrollment template, can be verified by matching the enrollment template against the verification template in the same scrambled space. If successful, at least k vascular interest points from the enrollment template are correctly found as a result of the positive genuine match. This enables the key-decoding process, which is the inverse of, but is similar to, key-encoding. Decoding enables the discovered subset of T_{AK} with cardinality of k or larger to compute W.

[0073] To mitigate cross-template attacks, where a resourceful attacker compromises a device, its code and logic, and gains access to multiple enrollment templates and tries to cross-match them, the attack can be thwarted by having the chaff contents across different templates within the matching distance of each other (or any significant part of the previous templates when synthesizing the chaff descriptors of each to be added to an enrollment template).

[0074] One implementation of a template matching algorithm is briefly described as follows. (1) An image pyramid is formed for a multi-scale matching process. (2) Points of

interest are found using a vascular point detector. (3) Features are calculated using multi radii LBP (local binary patterns), multi radii CS-LBP (center symmetric LBP), SURF, H-LBP (histogram of LBP), and H-CS-LBP (histogram of CS-LBP) around the aforesaid points. The result is saved as a naive enrollment template (a set of (x, y) vascular point coordinates plus descriptor vectors for the image patches around them, as described above). (4) Descriptors are shortened and de-correlated using pre-calculated PCA loadings, and isometrically scrambled (device-specific salting and re-shuffling of dimensions). Matching is performed in this surrogate private space. (5) Nearest neighbor matches between enrollment and verification template points are found based on Euclidean distances of all descriptors around enrollment-verification point pairs using a weighted sum. Candidate pairs are passed to the following outlier rejection step. (6) RANSAC with affine/non-reflective similarity hypothesis is performed to find outliers under assumed geometrical transform assumption, as well as the related transformation matrix. (7) The final match score is found as a nonlinear function of the correlation of x and y coordinates of the outlier-excluded enrollment-verification matched pairs, number of found pairs (k), and recovered scale and rotation from RANSAC (or other metric summarizing deviation of the transformation matrix from identity beyond reasonable values).

Key Decoding

[0075] In one implementation, the verification template is first matched against the augmented and obfuscated enrollment template to find k or more members of T_{VPD} upon successful genuine match. When using multiple ROIs or enrollment/verification templates for each biometric transaction, the first comparison to hit k matched points or higher can be used for computing the encoded W. One can also take the union of tagged augmented enrollment elements found through such multiple comparisons to achieve a higher k.

[0076] Next, using the tagging function f_k, k or more of the points from T_K are identified. These points are on the W-encoding function f_{encode} by design. Only k points are needed for an exact solution of the resulting system of equations, thus, the first k (or any other k members of the recovered T_K) from a successful verification process can be used. For each of the aforementioned k members of T_K, the respective $v_{i,d}$ is vectorized into k components using the same vectorizing (splitter) function described in "Vectorizing Functions," below. Along their corresponding $Y_d = [y_{i,d}]$, k-way vectorized $v_{i,d}(i = 1, 2, \ldots k)$ have enough information to

find their corresponding encoded key $\overrightarrow{W_d}(w_{i,d}, i = 1, 2, \ldots k)$ as follows: for each row d, k samples of $v_{i,d}$ (iterated over $i = 1, 2, \ldots k$) are split k ways by vectorizing function φ, above, giving rise to $[\varphi]_{k\times k}$. Key vector $\overrightarrow{W_d}$ is then found using the encoding fact:

$$[\varphi]_{k\times k}\,[w_d]_{k\times 1} = Y_d$$

And thus:

$$[w_d]_{k\times 1} = [\varphi]^{-1}_{k\times k}\,Y_d$$

Again, note that, because the k data points are used for equation-solving, order does not matter, and any subset of T_K with cardinality of k will suffice. Decoding using the light version described above follows a similar logic, but without the vectorizing function.

[0077] An initial security analysis will now be described. The following assumes a compromised device where the template is decrypted, and the biometric authentication code is decompiled. Given that secret key-carrying chaff T_K(with about $n(T_{VPD})$ members) are indistinguishable from the rest of the template elements, the chances of a lucky draw revealing a member of T_K is about $n(T_K)/n(T_A)$. A brute force attack for guessing all the required k points, considering the independent and identically distributed nature of such guesses, to solve the system of equations assuming a stolen and unencrypted enrollment template and program logic, plus availability of a measure of success, is then about $\left(\frac{n(T_K)}{n(T_A)}\right)^k$ because:

$$P(guess_1 \in T_K, guess_2 \in T_K, \ldots guess_k \in T_K) = \prod_{i=1}^{k} \frac{n(T_K) - i}{n(T_A) - i} < \left(\frac{n(T_K)}{n(T_A)}\right)^k$$

[0078] Thus, the effective entropy can be calculated as:

$$Entropy = -k \log_2 \left(\frac{n(T_K)}{n(T_A)}\right)$$

As an example, with $k = 40$ minimum genuine matched points, and typical number of chaff to total template points ratio of 1/5 (about 4 chaff points per vascular interest point), the entropy is larger than 92 bits.

[0079] Note that the capacity of the system, i.e., the size of the key W, is $D \times k \times L$ bits, where L is the length (in bits) of the number system used to encode W. For instance, only using SURF-128 features (the 128-dimensional version of SURF), and using unsigned 64-bit integer format to represent W (63 effective bits after discarding LSB to mitigate round off errors), the key capacity (length) is $128 \times 36 \times 63 = 290{,}304$ bits, or about 35 KB. This is not the entropy of the system, however, as calculated earlier. To enforce a standard length and strength in terms of key bit sequence flow, SHA512 can be applied to each encoded key W_D. Thus, regardless of the size of W_D, there is a pattern-unpredictable 512-bit private key sequence.

Sample Chaff Generation and Tagging Function Implementations

[0080] Tagging and using chaff decouples ensuing functionality from (already scrambled and obfuscated) real template points and descriptors spanned by vasculature, providing added security, privacy, and revocability. The following provide more specific details on various implementations of chaff, its generation, and tagging.

Spatial Placement of Chaff

[0081] The spatially uniform or otherwise non-distinguishable-from-vascular-interest-point "chaff-infusing" can be achieved in several ways to protect stored templates (generally enrollment templates, as verification templates are generated momentarily during matching). In one example, the minimum (outlier-rejected) spatial distance between real (non-chaff) interest points is determined. Chaff points are inserted until the distance between any two points (chaff and/or vascular interest points) is about the same minimum distance. A densely chaff-infused template will offer stronger security on multiple fronts. The downside is the larger size of the chaff-obfuscated template, which can also slow down the matcher.

[0082] Another less extreme implementation is a two-step chaff insertion. More specifically, given a typical tiling over the spatial span of the enrollment template, start with placing the first portion of the chaff (needed to make the average of total template points per area granule, chaff and non-chaff, about equal), using a minimum distance requirement (e.g., three pixels) for this step, known as coarse chaff insertion. The process continues with inserting the rest of the chaff until achieving the desired chaff to non-chaff ratio, typically 3x to 7x, by relaxing the minimum distance threshold (e.g., to one pixel) (fine chaff insertion step).

[0083] A further method for chaff placement includes, using an existing template, replicating the spatial patterns of vascular points in vascular tiles over non- (or almost non-) vascular tiles (in some cases, with small naturally occurring geometric distortions) while inserting chaff at empty locations/neighborhoods, observing continuity of spatial distribution of x, y coordinates of chaff-infused template at tile boundaries, as well as overall uniform spatial density per tile.

[0084] Yet another method includes following the same vascular tree-like structure if the closest dots are too close using an L-system (Lindenmayer grammar for tree-like structures). Then chaff is added, according to L-system generated spatial patterns, to less vascular tiles, until reaching a uniform tile density across template while observing continuity at tile boundaries.

Chaff Descriptor Contents

[0085] In one implementation, the descriptor feature vectors in a template, if considered as signals, are non-ergodic processes. The statistical properties of each feature element in a chaff-infused enrollment template, also with respect to what comes before and after it, in spatial and feature space, should be the same for chaff vs. non-chaff descriptors. The distribution of inter-descriptor distances, as well as their means and covariances matrices within and across chaff and non-chaff should also be similar. The aforesaid can be achieved by PCA projection that renders descriptors (chaff and non-chaff) zero mean and uncorrelated. Within the aforesaid boundaries, chaff descriptors of locations closer to vascular points can be chosen so that they are less likely to match against each other, so that the matching accuracy does not suffer (while remaining within VPD descriptor distribution characteristics). Besides creating chaff descriptor content from existing real point descriptors (e.g., application of a small circular shift plus a small noise to VPD-associated feature vectors), the PCA projection and scrambling function will further flatten any differences between chaff and genuine descriptors. Note that scrambling salts and reorders coordinates in a device specific manner, preserving Euclidean distances for matching purposes in scrambled space only within the given unique software and hardware environment, enabling two-factor authentication during a single biometric eye scan transaction. Optional eigenvalue normalization after eigenvector projections of PCA step creates a whitened stored template that has a close to identity covariance matrix across all its features for further security.

Tagging

[0086] Tagging functions can be implemented in many different ways, such as by using hash functions. For instance, assume x, y coordinates of an interest point and its corresponding feature vectors: (1) x, y coordinates are added with the first eight elements of the local feature vector V corresponding to the respective interest point. (2) The resultant is hashed with SHA512. The resulting bit string is grouped into 64 bytes. (3) To derive tagged (output) coordinates, two sets of sequences are extracted from the aforesaid byte string by considering all odd byte locations as one sequence (Seq1, 32 bytes), and all even locations as second sequence (Seq2, 32 bytes). (4) All the bytes in Seq1 are bit-XORed to get a single byte for a tagged x coordinate. Similarly, all the bytes in Seq2 are XORed to get a single byte as a tagged y coordinate. (5) If there is a chaff point at the aforementioned location, then it will be "tagged." If not, and the nearest chaff is at a radius of r pixels (e.g., one pixel), then the selection moves to the calculated location and is tagged. If none of the above occurs, a tagged chaff point is created at this location. Different rehashing of Seq1 and Seq2 can be implemented if the x, y range is beyond 0-255.

[0087] Another approach is to use mathematical functions for tagging locations. Assume a three-step process (T1, T2, and T3 below) applied in cascade. The (x, y) coordinates of the input template point are transformed as follows:

T1:

$$x_{new} = x \sin(y)$$

$$y_{new} = x \cos(x)$$

T2:

$$x_{new} = \begin{cases} -x & if\ x < 1 \\ x - x_{max} & if\ x > x_{max} \\ 1 & if\ x = 0 \\ x & else \end{cases}$$

$$y_{new} = \begin{cases} -y & if\ y < 1 \\ y - y_{max} & if\ y > y_{max} \\ 1 & if\ y = 0 \\ y & else \end{cases}$$

x_{max} and y_{max} are the maximum values for spatial coordinates in the chaff-infused template.

T3:

$$x_{new} = \begin{cases} x_{max} - x & if\ x\ is\ odd \\ x & else \end{cases}$$

$$y_{new} = \begin{cases} y_{max} - y & if\ y\ is\ odd \\ y & else \end{cases}$$

[0088] Note that tagging functions can be cascaded or re-parameterized to change behavior across different instantiations of the biometric authentication application. Chaff placement can be limited to the ROI mask (more specifically, a union of population ROI masks, in order to hide individual eyelid contours).

Example Algorithm for Chaff Location and Content Synthesis

[0089] One implementation of an algorithm for chaff location and content synthesis is as follows. Consider there are N original (VPD) points along their respective descriptors (currently H LBP, H CS LBP, and SURF), creating the template from an image of size $R \times C$ pixels (where R is the number of rows and C is the number of columns). In one implementation, steps for calculating chaff and tag are as follows:

1. Define chaff to vascular interest point "Ratio" parameter (e.g., approximately 3.5 to 4.5).
2. Insert tagged points for each original point used for Key Generation (Key Tag):
    a. Generate a tag point within the $R \times C$ window using a first tagging function that accepts the location and descriptor information of an original point as its input.
    b. Check if the tagged location is that of an original point:
        i. If yes, do nothing.
        ii. If no, but there is a chaff point within a one pixel radius, move the chaff to the tagged location.
        iii. Otherwise no:
            1. Create a chaff point at the location generated from the first tagging function.
            2. Generate descriptors for the above point using the closest original point.

descriptors (FineChaffDescriptor):

3. Insert tagged points for each original point used for Server HandShake (ServerTag).
    a. Generate a tag point within the $R \times C$ window using a second tagging function with the location and descriptor information of the original point.
    b. Check if the tagged point location is an original point or the KeyTag:
        i. If yes, do nothing.
        ii. If no, but there is a chaff point within a one pixel radius, move the chaff to the tagged location.
        iii. Otherwise no:
            1. Create the point generated from the second tagging function.
            2. Generate descriptors for the above point using the closest original point.

descriptors (FineChaffDescriptor):

4. Divide the $R \times C$ into k tiles of equal size (e.g., $k = 20$, for 4x5 tiles and $R = 80$, $C = 100$, +/– 20). It should be noted that the foregoing values are for purposes of example, and other possible values are contemplated. Certain values can change, for example, based on image sensor (resulting image resolution).
5. Calculate the number of points (Original + KeyTags + ServerTags) in each tile and find the maximum (MaxPoints).
6. Calculate required points and change type per tile:
    a. If Number of points in a tile is less than MaxPoints/2: Do CoarseChaff until MaxPoints/2 followed by FineChaff until total points is equal to MaxPoints +/- 5%. (As used in this example algorithm, +/- X% can refer to a random number within the range of -X to +X).
    b. If Number of points in a tile is greater than or equal to MaxPoints/2: Do FineChaff until total points is equal to MaxPoints +/- 5%.
7. For a random 20% (can be increased for higher chaff count) of the chaff generated in Step 6, create ChaffTagChaff.

a. Generate a tag point within the $R \times C$ window using a third tagging function with the location and descriptor information of the original point.

b. Check if the tagged point location is an original point or KeyTag or ServerTag or Chaff:

i. If yes, do nothing.

ii. If no, but there is a chaff point within a one pixel radius, move the chaff to the tagged location.

iii. Otherwise no:

1. Create the point generated from the third tagging function.

2. Generate descriptors for the above point using the closest original point descriptors (FineChaffDescriptor).

8. If the number of (KeyTag + ServerTag + CoarseChaff + FineChaff + ChaffTagChaff)/Original points is less than Ratio: Create FineChaff.

CoarseChaff

1. Generate a random chaff point within the tile that is at least three pixels away from all points.

2. CoarseChaffDescriptor: Take the closest Original Descriptor (OrigDesc).

3. For SURF descriptors:

a. NewSURFdescriptor = CircularShift(OrigDesc, +/- 30% length) + (0.01% Gaussian noise).

b. If normalized SSD of (OrigDesc, NewSURFdescriptor) < 0.1 goto 3.a.

4. For HLBP descriptors:

a. NewHLBPdescriptor = CircularShift(OrigDesc, +/- 30% length) + (20% Gaussian noise).

b. If normalized SSD of (OrigDesc, NewHLBPdescriptor) < 0.1 goto 4.a.

5. For HDLBP descriptors:

a. NewHCSLBPdescriptor = CircularShift(OrigDesc, +/- 30% length) + (20% Gaussian noise).

b. If normalized SSD of (OrigDesc, NewHCSLBPdescriptor) < 0.1 goto 5.a.

FineChaff

1. Generate a random point within the tile that is at least 1 pixel away from all Points.
2. FineChaffDescriptor: Take the closest Original Descriptor (OrigDesc).
3. For SURF descriptors:

   3.1. NewSURFdescriptor = CircularShift(OrigDesc, +/-30% length) + (0.01% Gaussian noise).

   3.2. If normalized SSD of (OrigDesc, NewSURFdescriptor) < 0.2 goto 3.1.

4. For HLBP descriptors:

   4.1. NewHLBPdescriptor = CircularShift(OrigDesc, +/-30% length) + (20% Gaussian noise).

   4.2. If normalized SSD of (OrigDesc, NewHLBPdescriptor) < 0.225 goto 4.1.

5. For HDLBP descriptors:

   5.1. NewHCSLBPdescriptor = CircularShift(OrigDesc, +/-30% length) + (20% Gaussian noise).

   5.2. If normalized SSD of (OrigDesc, NewHCSLBPdescriptor) < 0.225 goto 5.1.

Vectorizing Functions

[0090] A simple yet secure and efficient way to split a scalar such as $v_{i,d}$ in k ways is to provide the scalar (or a function of it) to a hash function such as SHA512, and use groups of the produced bit strings as the desired series of numbers. The reasons for using vectorizing functions are as follows: (1) numerical stability of spanned system of linear equations irrespective of descriptor content (which for instance could be very close to zero especially within constraints of the given numerical precision for several locations in a feature vector); (2) larger capacity for multiple or larger key contents, as each vector element can span its own linear mixture equation line; and (3) equation coefficients need to be calculated by the template elements at runtime, rather than just recalled from their stored values, for added security.

[0091] Another example of a vectorizing function is as follows. Other deterministic and secure vectorizing functions that result in stable non-singular solutions for the decode process are also acceptable.

[0092] Seed a pseudo random number generator (PRNG) with a function of $v_{i,d}$ and create a sequence of k pseudo random numbers. For instance, use a cryptographically secure PRNG algorithm denoted by $f_{md\_num\_gen}$ and seed it with

$$f_{seed}(k, v_{i,d}) = \lfloor 2^{31} |\cos(k v_{i,d})| \rfloor$$

[0093] One can use more than one $v_{i,d}$ in this process, e.g., combine $v_{i,d} + v_{i,d+1}$ (or more, effectively, lowering D at the expense of reducing capacity of W) into one for added numerical stability and irreversibility.

[0094] Next, take the resulting first k pseudo random numbers, $rnd\_seq_i, i = 1,2, \ldots k$ as the vectorized output. Thus the vectorizing function is:

$$\overrightarrow{rand\_seq}_{i,d} = f_{md\_num\_gen}\left(f_{seed}(k, v_{i,d})\right)$$

[0095] Optionally, for added security and dynamic range control, one can pass the above $v_{i,d}$ spanned vectors through a nontrivial noninvertible function $\varphi(x)$. One example is as follows. Apply $rnd\_seq_i = (rnd\_seq_i - 0.5) \times 8$ (to linearly project the random sequence to $[-4, 4]$ to produce more unpredictable fluctuations with the following $\varphi(\bullet)$). One example for φ (depicted below) is:

$$\varphi(x) = \tanh(x - 10) \sin\left((x - 10)e^{-\frac{x-10}{2}}\right)$$

[0096] Finally, the corresponding $y_{i,d}$ for the input $v_{i,d}$ and its associated/encoded $\vec{W}_d$ (row d of the secret key matrix W) is given by:

$$y_{d,i} = f_{encode}(\vec{W}_d, v_{d,i}) = \sum_{j=1}^{k} w_{d,j} \varphi(rnd\_seq_d(j))$$

As mentioned, using the earlier noted SHA based vectorization negates the need for these type of vectorizations.

Trust Server Functionality

[0097] In one implementation, the trust server is an optional added layer of security that can be used in conjunction with the local key approach. Another added benefit to the trust server is surrogate remote verification and template/access revocability. For instance, if the server does not recognize the token sent by the device (a unique but re-issuable byproduct of biometric eye scan matching at the time of verification), then it can send a signal to, for example, the concerned online banking service or other service using the biometric authentication, not to honor the particular requested transaction. The details of the present implementation parallels in most part the chaff tagging and template matching processes described above.

[0098] Assume that S_{CHF}, a hash $H(.)$ of the descriptor part of the T_S: $\{\vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d\}, i = 1,2, \ldots, n(T_S) \longrightarrow S_{CHF} = H(\{\vec{V}_i^1, \vec{V}_i^2, \ldots \vec{V}_i^d\}) = \{h_i\}, i = 1,2, \ldots, n(T_S)$, is designated as the master chaff record and stored on the trust server at the time of enrollment (e.g., one master chaff record per enrollment in multi-enrollment systems). At the time of biometric verification, if the trust server validation is desired, the following "handshake" process can take place: the matched subset of template elements T_{VER}, is provided to f_S, a second chaff tagging function similar to f_K but for trust server functionality, yielding $S_{VER} = H(T_{VER})$, which is sent to the trust server at the time of verification. From the properties of the matcher, it is known that for a successful genuine match:

(a) $T_{VER} \subset T_{VPD}$, and

(b) $n(T_{VER}) \geq k$

[0099] That is, a successful match finds at least k of the real vascular interest points, and a failed (e.g., impostor) match does not. Thus, it follows that the following conditions have to be met at the server side to verify the integrity of device-side match:

$$S_{VER} \subset S_{CHF} \text{ and } n(S_{VER}) \geq k$$

[00100] Note that one can also transmit a time-varying hash of S_{VER}, e.g., by nested repetition of SHA512 on S_{VER} n times, with n being a function of a universal time stamp (e.g., a modulus). The trust server will perform the same time-varying hash of its S_{CHF} before any comparisons.

[00101] Other possible functionalities of the trust server include revoking access to remote service (e.g., in case of a stolen device), as the new enrollment on the new device will create

different S_{VER} and S_{CHF}. Note that server chaff is not identical to key generation chaff and thus this separation provides partial independence and thus added security over several hypothetical attack vectors. Otherwise, verification accuracy and validation security of private key vs. server chaff could be considered to be the same.

[00102] An initial security analysis is as follows. The following scenario assumes a compromised device where the template is decrypted, the biometric authentication code is decompiled, and thus the device-server handshake logic plus template structure are known to the attacker. Given the indistinguishability of chaff and real vascular interest points, the probability of a lucky first draw from the template is at most $\frac{n(T_S)}{n(T_A)}$, i.e., the ratio of tagged chaff by f_S (about the same as $n(VPD)$) divided by the total number of template elements, because:

$$P(guess_1 \in T_S, guess_2 \in T_S, \ldots guess_k \in T_S) = \prod_{i=1}^{k} \frac{n(T_S) - i}{n(T_A) - i} < \left(\frac{n(T_S)}{n(T_A)}\right)^k$$

with the assumption that such guesses are independent and identically distributed.

[00103] The chances for the attacker to be able to collect all the required minimum k of T_S members by guessing is exceedingly minimal. Using typical values of about one tagged chaff for each vascular interest point, and four total inserted chaff for each vascular interest point, and $k = 40$ for a single 2-ROI scan, the chance of success at first try is:

$$\left(\frac{n(T_S)}{n(T_A)}\right)^k = 0.2^{40} = 1.1 \times 10^{-28}$$

If the trust server limits the number of failed attempts, the overall chance of success for such an attack remains very small. Furthermore, if an attacker compromises both the trust server and the user's device and deciphers all the required content, he or she cannot access the vascular interest point portion of the user template by subtracting the server master chaff record from the user device template, as T_S is only a subset of T_{CHF}.

[00104] The terms and expressions employed herein are used as terms and expressions of description and not of limitation, and there is no intention, in the use of such terms and expressions, of excluding any equivalents of the features shown and described or portions thereof. In addition, having described certain implementations in the present disclosure, it will be apparent to those of ordinary skill in the art that other implementations incorporating the

concepts disclosed herein can be used without departing from the spirit and scope of the invention. The features and functions of the various implementations can be arranged in various combinations and permutations, and all are considered to be within the scope of the disclosed invention. Accordingly, the described implementations are to be considered in all respects as illustrative and not restrictive. The configurations, materials, and dimensions described herein are also intended as illustrative and in no way limiting. Similarly, although physical explanations have been provided for explanatory purposes, there is no intent to be bound by any particular theory or mechanism, or to limit the claims in accordance therewith.

What is claimed is:

1. A computer-implemented method comprising:

receiving one or more images;

identifying a plurality of interest points based on the received images;

generating a plurality of obfuscating data points based on the interest points;

creating an obfuscated template based on the interest points and the obfuscating data points; and

storing the obfuscated template.

2. The method of claim 1, wherein the obfuscating data points are generated such that a spatial distribution of the interest points and a spatial distribution of the obfuscating data points are substantially similar.

3. The method of claim 1, further comprising discarding a record of which points in the obfuscated template are the interest points.

4. The method of claim 1, further comprising encoding a key using a subset of at least one of the obfuscating data points and the interest points.

5. The method of claim 4, wherein each point in the subset is determined based on a different one of the interest points.

6. The method of claim 1, wherein the images comprise biometric imagery.

7. The method of claim 6, wherein the images comprise images of a region of an eye, each eye region image comprising a view of a vasculature of the respective eye region, and wherein the interest points comprise vascular interest points.

8. The method of claim 1, further comprising associating one or more real descriptors with each interest point, wherein each real descriptor describes one or more localities surrounding the corresponding interest point.

9. The method of claim 8, further comprising associating one or more synthesized descriptors with each obfuscating data point, wherein each synthesized descriptor comprises a statistical similarity to the real descriptors.

10. The method of claim 9, further comprising:

receiving one or more second images;

identifying a second plurality of interest points based on the received second images;

creating a verification template based on the second plurality of interest points;

comparing the verification template with the obfuscated template to identify a plurality of matching interest points; and

authenticating a user based on the matching interest points.

11. The method of claim 10, wherein the comparing comprises identifying the matching interest points based on one or more of the real and synthesized descriptors.

12. The method of claim 10, further comprising reducing a dimensionality of the real descriptors and the synthesized descriptors.

13. The method of claim 12, wherein the comparing comprises identifying the matching interest points based on one or more of the reduced dimensionality descriptors.

14. The method of claim 10, further comprising isometrically scrambling the real descriptors and the synthesized descriptors.

15. The method of claim 14, wherein the comparing comprises identifying the matching interest points based on one or more of the scrambled descriptors.

16. The method of claim 10, further comprising decoding a key based on at least a subset of the matching interest points.

17. A system comprising:

one or more computers programmed to perform operations comprising:

receiving one or more images;

identifying a plurality of interest points based on the received images;

generating a plurality of obfuscating data points based on the interest points;

creating an obfuscated template based on the interest points and the obfuscating data points; and

storing the obfuscated template.

18. The system of claim 17, wherein the obfuscating data points are generated such that a spatial distribution of the interest points and a spatial distribution of the obfuscating data points are substantially similar.

19. The system of claim 17, wherein the operations further comprise discarding a record of which points in the obfuscated template are the interest points.

20. The system of claim 17, wherein the operations further comprise encoding a key using a subset of at least one of the obfuscating data points and the interest points.

21. The system of claim 20, wherein each point in the subset is determined based on a different one of the points of interest.

22. The system of claim 17, wherein the images comprise biometric imagery.

23. The system of claim 22, wherein the images comprise images of a region of an eye, each eye region image comprising a view of a vasculature of the respective eye region, and wherein the interest points comprise vascular interest points.

24. The system of claim 17, wherein the operations further comprise associating one or more real descriptors with each interest point, wherein each real descriptor describes one or more localities surrounding the corresponding interest point.

25. The system of claim 24, wherein the operations further comprise associating one or more synthesized descriptors with each obfuscating data point, wherein each synthesized descriptor comprises a statistical similarity to the real descriptors.

26. The system of claim 25, wherein the operations further comprise:

receiving one or more second images;

identifying a second plurality of interest points based on the received second images;

creating a verification template based on the second plurality of interest points;

comparing the verification template with the obfuscated template to identify a plurality of matching interest points; and

authenticating a user based on the matching interest points.

27. The system of claim 26, wherein the comparing comprises identifying the matching interest points based on one or more of the real and synthesized descriptors.

28. The system of claim 26, wherein the operations further comprise reducing a dimensionality of the real descriptors and the synthesized descriptors, and wherein the comparing comprises identifying the matching interest points based on one or more of the reduced dimensionality descriptors.

29. The system of claim 26, wherein the operations further comprise isometrically scrambling the real descriptors and the synthesized descriptors, and wherein the comparing comprises identifying the matching interest points based on one or more of the scrambled descriptors.

30. The system of claim 26, wherein the operations further comprise decoding a key based on at least a subset of the matching interest points.

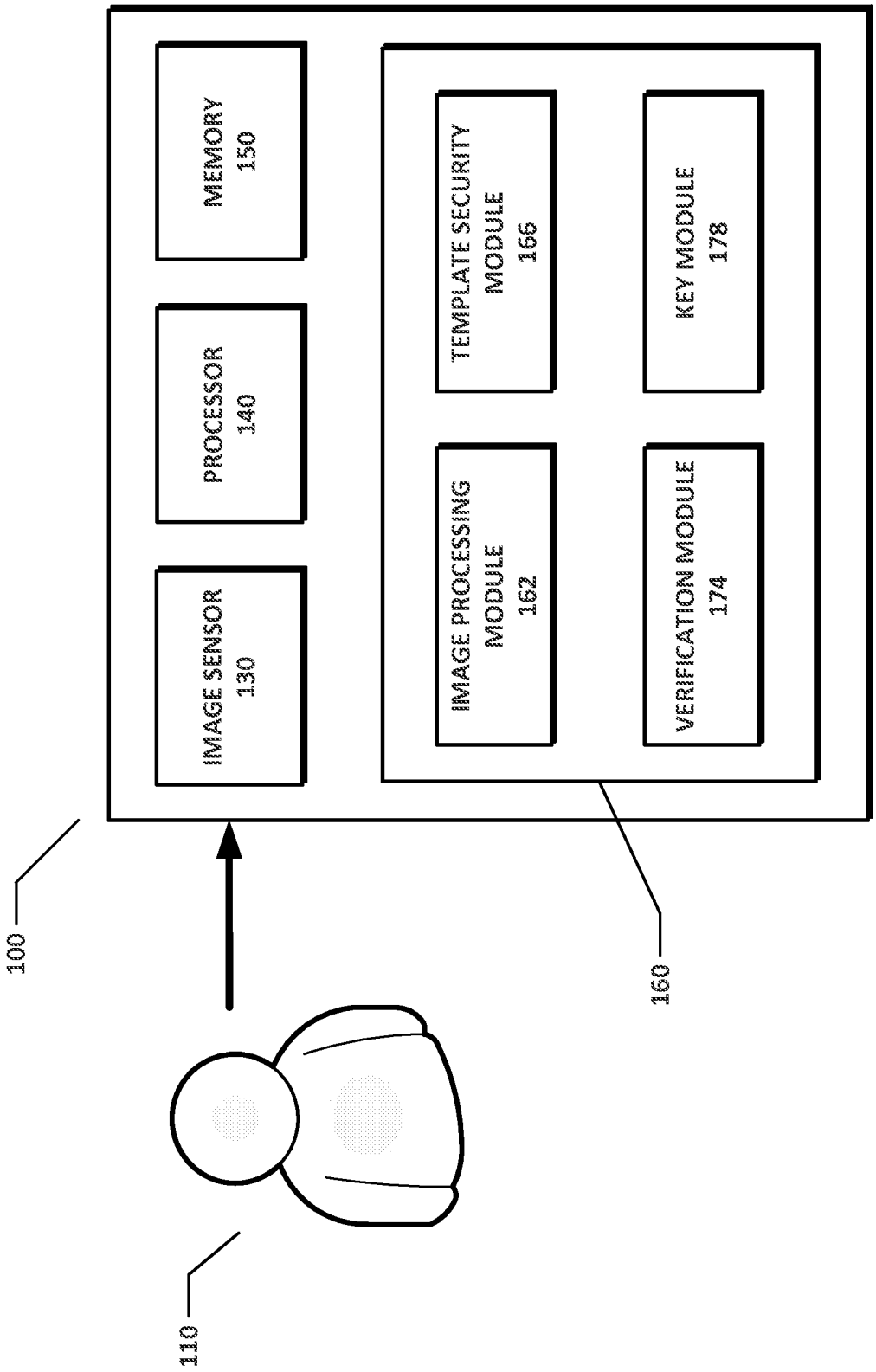
MEMORY 150
PROCESSOR 140
IMAGE SENSOR 130
TEMPLATE SECURITY MODULE 166
KEY MODULE 178
IMAGE PROCESSING MODULE 162
VERIFICATION MODULE 174
100
160
110

FIG. 1

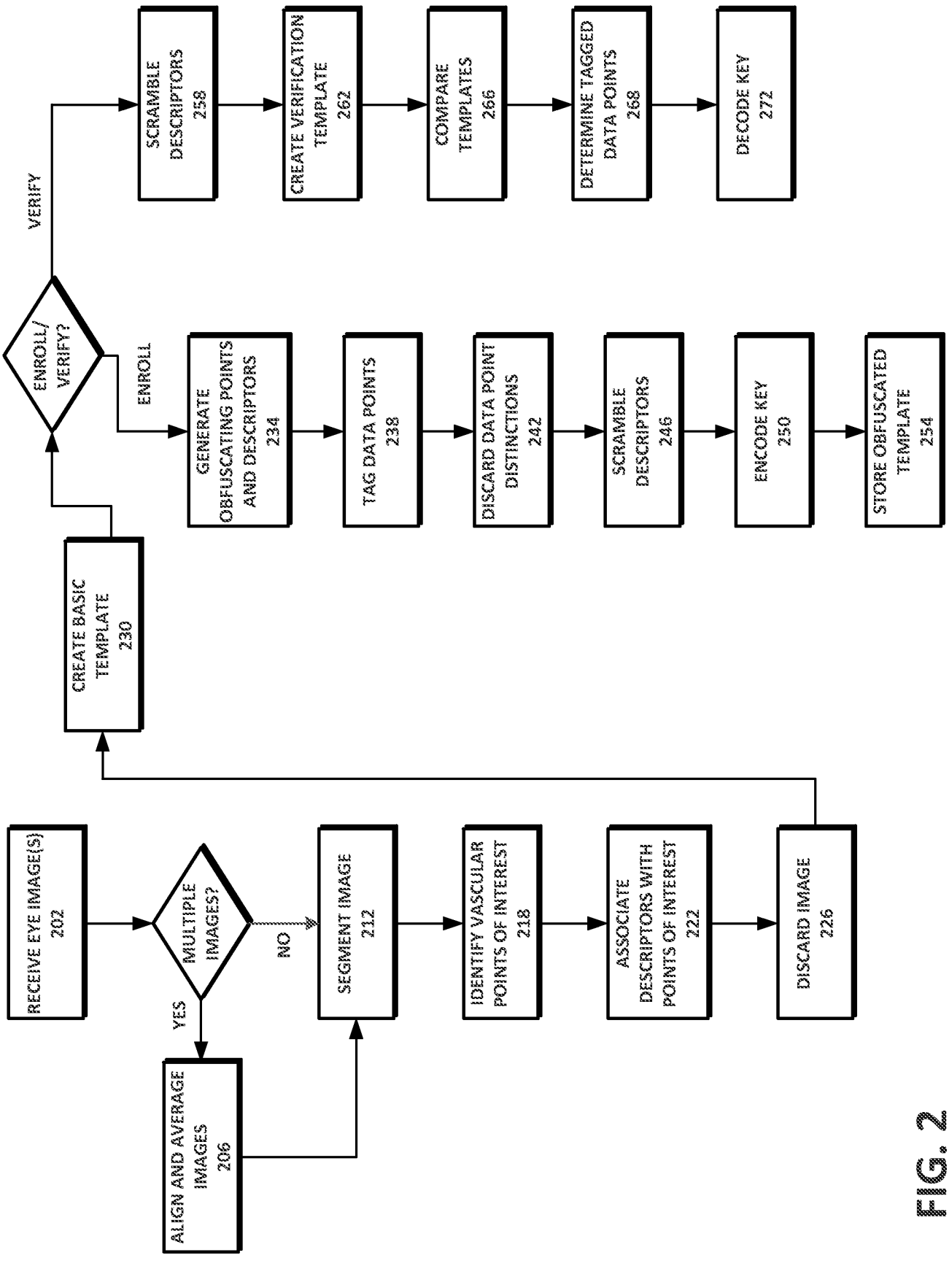
RECEIVE EYE IMAGE(S) 202
MULTIPLE IMAGES?
YES
NO
ALIGN AND AVERAGE IMAGES 206
SEGMENT IMAGE 212
IDENTIFY VASCULAR POINTS OF INTEREST 218
ASSOCIATE DESCRIPTORS WITH POINTS OF INTEREST 222
DISCARD IMAGE 226
CREATE BASIC TEMPLATE 230
ENROLL/ VERIFY?
ENROLL
VERIFY
GENERATE OBFUSCATING POINTS AND DESCRIPTORS 234
TAG DATA POINTS 238
DISCARD DATA POINT DISTINCTIONS 242
SCRAMBLE DESCRIPTORS 246
ENCODE KEY 250
STORE OBFUSCATED TEMPLATE 254
SCRAMBLE DESCRIPTORS 258
CREATE VERIFICATION TEMPLATE 262
COMPARE TEMPLATES 266
DETERMINE TAGGED DATA POINTS 268
DECODE KEY 272

FIG. 2

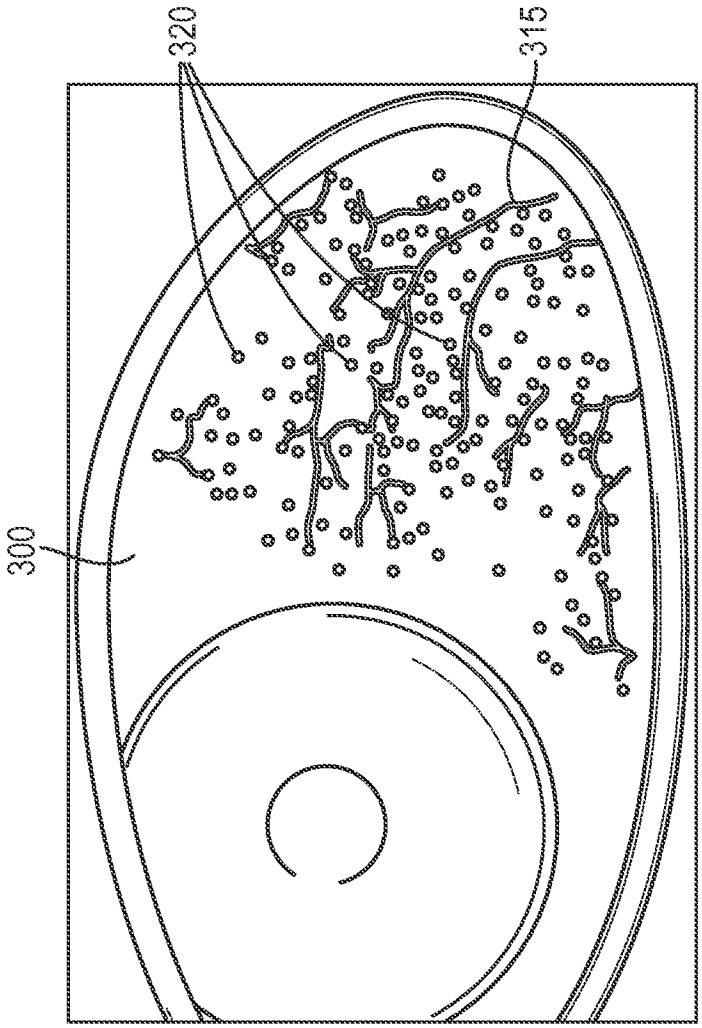

FIG. 3

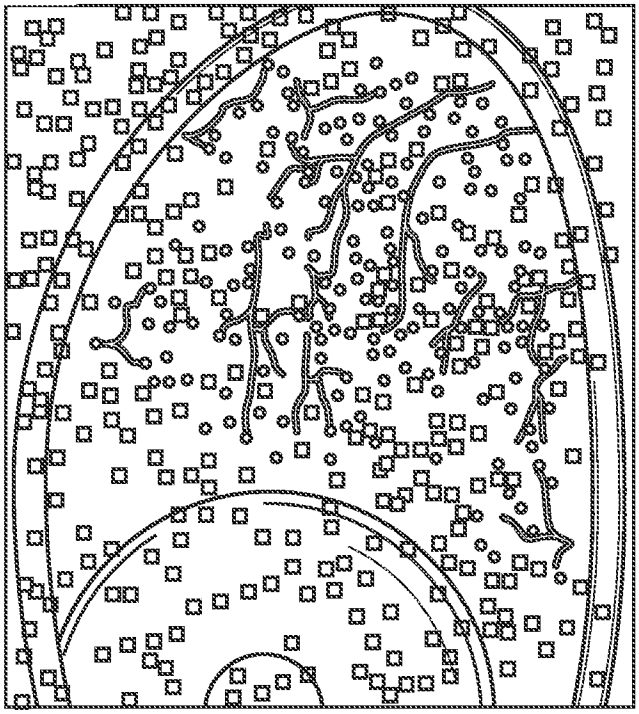

FIG. 4B

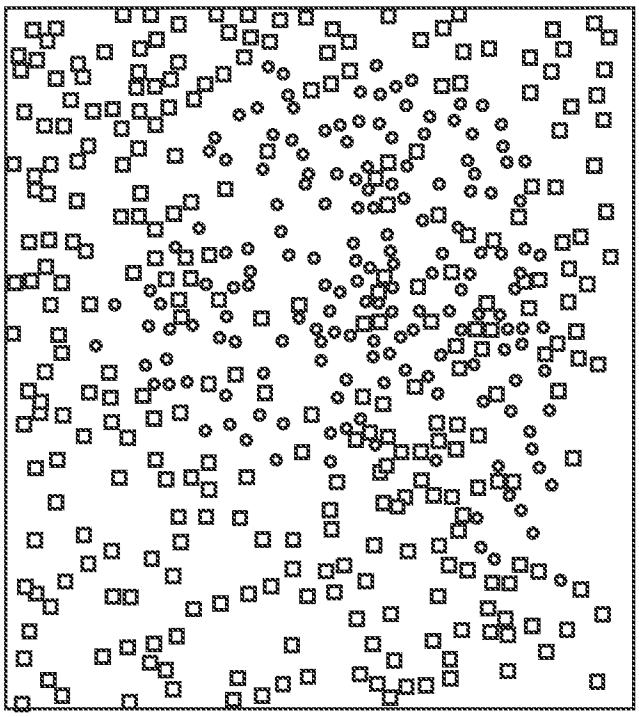

FIG. 4A

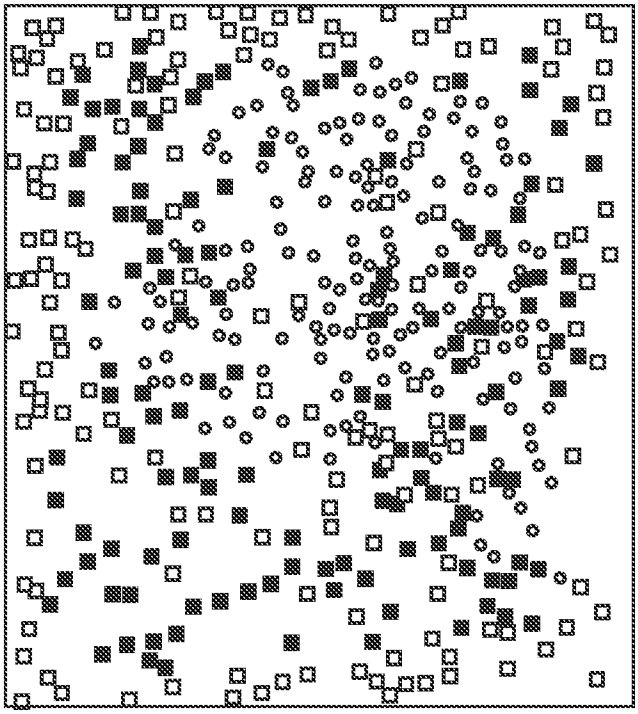

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/055826

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06K9/00 G06F21/32 H04L9/08
ADD.

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)
G06K G06F H04L

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)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	MOHAMED KHALIL-HANI ET AL: "Securing cryptographic key with fuzzy vault based on a new chaff generation method", HIGH PERFORMANCE COMPUTING AND SIMULATION (HPCS), 2010 INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 28 June 2010 (2010-06-28), pages 259-265, XP031731476, ISBN: 978-1-4244-6827-0 the whole document ----- -/--	1-30

[X] 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

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
3 December 2014	12/12/2014

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	de Bont, Emma

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/055826

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Ann Cavoukian, Alex Stoianov: "Biometric Encryption" In: Hank C.A. van Tilborg, Sushil Jajodia: "Encyclopedia of Cryptography and Security", 2011, Springer, XP002733187, ISBN: 978-1-4419-5905-8 pages 90-98, the whole document -----	1,3,6, 17,19,22
X	MEENAKSHI V S ET AL: "Security Analysis of Hardened Retina Based Fuzzy Vault", ADVANCES IN RECENT TECHNOLOGIES IN COMMUNICATION AND COMPUTING, 2009. ARTCOM '09. INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 27 October 2009 (2009-10-27), pages 926-930, XP031564092, ISBN: 978-1-4244-5104-3 the whole document -----	1-10, 17-26
X	UMUT ULUDAG ET AL: "Fuzzy Vault for Fingerprints", 28 June 2005 (2005-06-28), AUDIO- AND VIDEO-BASED BIOMETRIC PERSON AUTHENTICATION; [LECTURE NOTES IN COMPUTER SCIENCE;;LNCS], SPRINGER-VERLAG, BERLIN/HEIDELBERG, PAGE(S) 310 - 319, XP019013285, ISBN: 978-3-540-27887-0 the whole document -----	1-6, 8-11, 14-17, 22, 24-27, 29,30

1