

The user 126 provides an input pattern made up of a sequence of discrete graphical choices to a graphical interface 112, which is a component of the authentication system 1410. The display parameter 124, which in one embodiment is an image file, may help the user 126 to recall his secret pattern. In FIG. 14, the input pattern is processed by components of the enrollment system 110 including a converter 114, a summer 1420, in conjunction with a sequence of thresholds in a memory element 1430, a codeword generator 116, a hasher 1440, and a comparator 1450, in conjunction with a hash in a memory element 1430, to determine whether the input pattern corresponds to a visual password established during enrollment.

[0131] Entry of an input pattern that corresponds to a visual password established during enrollment will enable the user to access a resource 1650. In embodiments, the resource 1460 is a computer system, a database, or other resource that the user 126 desires to employ. In another embodiment, the resource 1460 provides computational resources or data that the user 126 would like to access. In another embodiment, the resource 1460 is a physical location or entity that the user 126 desires to access or use, such as a room, a locked automobile, or the locked ignition mechanism for an automobile.

[0132] A memory element 1430 contains at least a portion of the visual password. The memory element may be RAM or ROM. The memory element may be a component of the authentication system 1410 or a separate element in signal communication with the authentication system 1410. A memory element 1430 in various embodiments is a CD-ROM, a network device, a smartcard, a personal digital assistant (PDA), a magnetic strip which may be attached to a card the approximate size of a credit card, or a bar code.

[0133] In one embodiment, the display parameter 124, a converter 114, a summer 1420, in conjunction with a sequence of thresholds in a memory element 1430, a codeword generator 116, a hasher 1440, and a comparator 1450, in conjunction with a hash in a memory element 1430, are implemented as a single software application program executing on a general purpose computer system. In various embodiments, the summer 1420 and the hasher 1440 are individually implemented as software modules, programs, or objects, such as objects implemented in the C++ programming language. In other embodiments, one or more of the summer 1420 and the hasher 1440 are combined in a hardware device or integrated chip, such as an ASIC (application-specific integrated circuit).

[0134] The graphical interface 112, converter 114, and codeword generator 116 and comparator 1450 are generally as described with respect to FIG. 1 and FIG. 2.

[0135] The converter 114, the codeword generator 116, the security calculator 118, and the comparator 120, in conjunction with a threshold value 122, are all used by the authentication system 1410 to perform the authentication process. Their individual functions are described in more detail with respect to the steps, illustrated in FIG. 15, which are part of the authentication process.

[0136] FIG. 15 shows a flowchart of an authentication process in accordance with one embodiment of the invention. STEPS 1510, 1520, 1530, and 1570 of FIG. 15 are similar to STEPS 1310, 1320, 1330, and 1370 of FIG. 3. An

input pattern is received (STEP 1510) from the user 126 on a graphical interface 112 (FIG. 14). The input pattern may be a sequence of discrete graphical choices, as described above. Each discrete graphical choice in the sequence of discrete graphical choices that form the input pattern is converted to an input value (STEP 1520). A sequence of input values is thereby produced from the input pattern.

[0137] The summer 1420 (FIG. 14) retrieves a sequence of offsets (STEP 1524). In one embodiment, the sequence of offsets is retrieved from a memory element 1430 associated with the summer 1420 and recovered from the authentication system 1410 itself. In another embodiment, the sequence of offsets is retrieved by signal communication with a resource 1460 (FIG. 14) that is not part of the authentication device 1410.

[0138] Each offset  $\delta$  reveals relative information about a value  $x$ , specifically the differences between the value  $x$  and the associated codeword, but not any absolute information about the value  $x$ . As illustrated in the geometric analogy of FIG. 9, an offset  $\delta$  reveals the location of a value  $x$  relative to the associated codeword  $c_3$ , but does not reveal any information about the absolute location of the committed codeword  $c_3$  or the value  $x$  on the  $u$ - $v$  plane. Thus, in embodiments in which the sequence of codewords is concealed by a hash function  $h$ , such as illustrated in FIG. 15, the only information that the an offset  $\delta$  effectively reveals about a value  $x$  is that it takes the form  $(u+170, v-95)$  for some points  $(u, v)$ . Subject to this constraint, the value  $x$  could otherwise lie anywhere in plane.

[0139] Referring again to FIG. 15, the summer 1420 (FIG. 14) sums each offset in the sequence of offsets with the corresponding value in the sequence of input values (STEP 1526). A sequence of intermediate values is thereby generated from the sequence of input values. Referring to FIG. 9, in that geometric analogy, the input value  $x'$  corresponds to a discrete graphical choice from the input pattern and is represented by a point on the  $u$ - $v$  plane with the coordinates  $(40, 550)$ . The summing step (STEP 1526) corresponds to the translation of the input value  $x'$  by the offset 8 (FIG. 9), just as the value  $x$  was translated to reach the committed codeword  $c_3$ . The intermediate value  $i$ , which is treated as a corrupted codeword, is represented in FIG. 9 as a point on the  $u$ - $v$  plane with the coordinates  $(210, 455)$ . In mathematical notation,  $i=x'-\delta$ .

[0140] Referring again to FIG. 15, a codeword generator 116 (FIG. 14) selects a codeword for each intermediate value in the sequence of intermediate values (STEP 1530). A sequence of codewords is thereby generated from the sequence of intermediate values. Embodiments of the invention require the decoding function  $f$  associated with, but not necessarily used in, the enrollment process to generate an appropriate codeword for each intermediate value. The decoding function  $f$  decodes the intermediate value, which is treated as the corrupted codeword  $i$ , into the nearest codeword. During the authentication process, a codeword is not selected at random from the set of codewords associated with an error-correcting code because the goal of the authentication process is to match each codeword that was selected during enrollment. Again using the geometric analogy of FIG. 9, the decoding function selects codeword  $c_3$  for the intermediate value  $i$  because the intermediate value  $i$  is closer to codeword  $c_3$  than any other codeword.