

# Humanly Computable Passwords as Lattice based OTP generator with LWE.

Sławomir Matelski\*

December 2020

## Abstract

For safe resource management - an effective mechanism/system is necessary that identifies a person and his rights to these resources, using an appropriate key, and its degree of security determines not only the property, but sometimes even the life of its owner. For several decades, it has been based on the security of (bio)material keys, which only guarantee their own authenticity, but not their owner, due to weak of static password protection. In the article will be presented the i-Chip an innovative challenge-response protocol, implements the Learning with Rounding (LWR) method as LWE variant, the known NP-hard problem, and based on post quantum lattice cryptography. The i-Chip scheme requires only single (albeit detailed) picture, illustrating the grid of a virtual microchip along with a brief set of rules, describing the way in which such chip operates, thanks to which reverse engineering of such virtual structure is as difficult as similarly operation on a physical microchip, and their diversity is limited only by the fantasy of the users and collaborating researchers, inspired by the analogous features of the physical elements. It will be shown how the i-Chip generates the one-time password (OTP) or whole digital signature, also offline on paper documents.

## 1 Introduction

For many centuries before the age of the computer and the Internet, a handwritten signature was sufficiently secure for document authentication, only occasionally supplemented with an optional seal. Both of these methods leave a permanent ink trail on the document. The shape of the autograph is similar, but not the same each time, and its safety was determined by the degree of difficulty in repeating it so as to be sufficiently similar, but not identical. because the ideal pattern could only be achieved by mechanical action. Unlike autograph, when using a stamp, the identical trace is always expected on the document. An autograph is a material product of behavioral biometrics as well as a projection of identity. It has a stylized individual shape and hence the topography of characteristic elements which is the result acquired ability to duplicate the remembered pattern. Each owner of the identity can create or change his autograph, but, of course, it requires an update to the signature pattern, kept in the appropriate certification authority. The currently used authentication standards also make a mark on electronic documents, using electronic stamps in the form of smart cards, which until recently did not even guarantee their own authenticity, because their technical constraints (e.g. RFID) forced the use of a weak cryptographic protocol. [6].

However, **on what basis are we to believe that this e-stamp is used by the owner and not by another person?**

The only security here is a static password or PIN, which equate the intellectual level of all users to the level of illiteracy, which puts 3 or 4 crosses for authentication instead of a stylized autograph due to the ease of counterfeiting or easy to interception. Other ineffective ways to strengthen security employ next technological gadget, such as a smart phone, which can be used, but less and less often used as an additional information channel. However, a smartphone is as also exposed to malware infection like a laptop or PC, and is increasingly used as the only information channel. Moreover, even in the case of using 2 channels/devices - there is no guarantee that they are safe and under the sole control of the owner, not the adversary.

In order to achieve greater credibility of this method of authentication, sometimes the biometric features of the body are used, but the body is also a material element and is subject to the same laws as a stamp.

---

\*S. Matelski, INTELCO Ltd., (corresponding author)  
e-mail: s.matelski@intelco.pl

Commercially available technology makes it possible to secretly scan such an element and then make a model of it, not to mention the more brutal methods of the criminal world, which does not hesitate to even kill someone, not only needed to scan a part of the body. Moreover, once lost body biometric data means an irreversible loss of control over one's own identity, because the exchange of body elements, such as a finger or an eye, is rather unrealistic. Much more common, however, is the problem with sufficient repeatability of the behavioral pattern or the technical inaccuracy of used scanners.

It seems that the only solution that guarantees the security of authentication is the use of the autograph features mentioned at the beginning, i.e. similarity, but not identity, of which OTP supported by the LWR method is the logical equivalent.

The first attempts to create the Human Generated Passwords (HGP) protocol were published in year 1991 in the seminar work of Matsumoto and Imai [1] and despite the lapse of 30 years to the date on which our article was written, none of the solutions developed so far have reached the commercially expected level that requires simultaneous satisfies of 2 difficult to reconcile properties: acceptable level of safety and acceptable degree of usability.

Schemes closest to this level: Foxtail developed in 2005 by S. Li [11] [2] and Shared Cues developed in 2013/2017 by J. Blocki [4] - we propose as additional variants of the i-Chip protocol, which thanks to its special properties acts as a common platform for them and for creating many new versions which will arise in the future. We hope that the initial state of research presented in this paper will attract the attention of other cryptologists to create own new licensed variants of i-Chip protocol.

The implementation of these schemas has become even more secure and more usable. The finished implementation of the i-Chip protocol can be viewed on a short film or it can be tested in the interactive demo or in the realy e-banking system at [www.i-sign.org](http://www.i-sign.org)

An important element of the i-Chip scheme is the implementation of the LWR method, which, as an LWE variant, is also considered as an NP-hard problem. Earlier, this method was introduced into the Foxtail+ [15] and HB+ [23] schemas, bringing the expected security effects, but at the same time the degree of usability has decreased and authentication now requires more time as the user has to perform additional protocol rounds to compensate for rounds lost to incorrect binary responses. The implementation of the LWR method in i-Chip does not require any additional steps, and the user only randomly selects between several correct answers in the form of decimal numbers, so the procedure is several times shorter and faster.

The i-Chip is already the 4th step in the evolution of the TopoSign (Topographic Signature) scheme, after 0G graphical UserID key, 1G/2G easy OTP generator with entropy reduction, and 3G (Invisible Ink), developed by S. Matelski in 2009-2012. However, only the fusion of the Invisible Ink concept of the pixelation properties to draw in special wizard the private key as photomask of all components of such a "micro chip" which includes the information about them and their logical relations and map of paths conducting the digital signal from input to output, forming its value from Vinp to Vout, brought the expected result, allowing to build a secure generator of OTP.

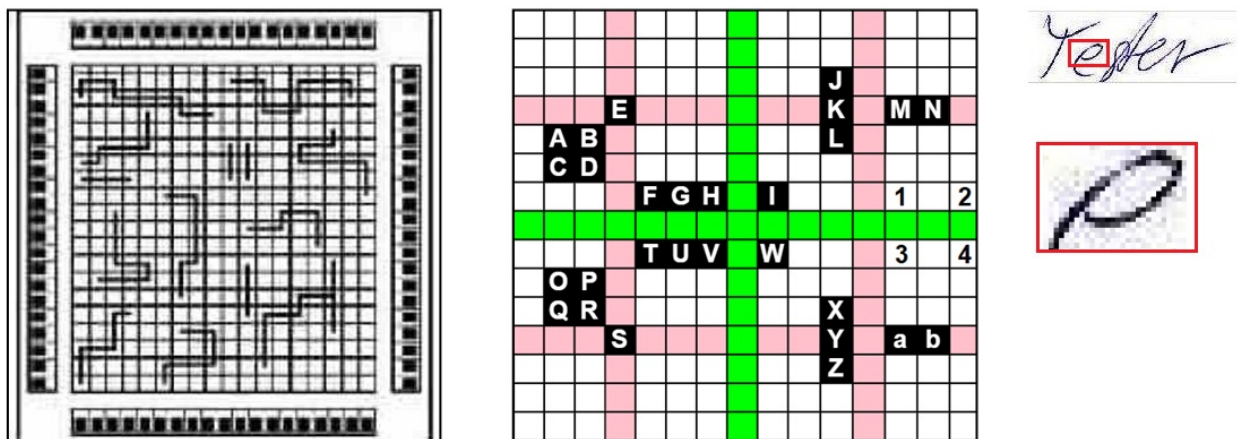


Figure 1: Layout of Programmable Logic Array vs. i-Chip layout and pixel-topography of handwriting.

## 1.1 Notation

Throughout the paper, by  $\mathbb{Z}_n$  we will denote the set  $\{0, 1, \dots, n - 1\}$ . The metric which will be used to measure the distance between two entries (which we will call the *matrix distance*) described by the pair of coordinates from  $\mathbb{Z}_n \times \mathbb{Z}_n$  will be given by Manhattan metric  $d$ , i.e.

$$\forall_{(a,b),(c,d) \in \mathbb{Z}_n \times \mathbb{Z}_n} \quad d((a,b), (c,d)) = |a - c| + |b - d|.$$

Thus pairs of coordinates  $(a,b), (c,d) \in \mathbb{Z}_n$  are said to be *adjacent* if the distance between them equals 1, i.e.,  $d((a,b), (c,d)) = 1$ . It means that the matrix entries described by these coordinates are either horizontally or vertically (but not diagonally) next to each other.

Additionally, for a (finite) sequence  $A := (a_1, \dots, a_n)$  we will write  $a \in A$  whenever there exists index  $i \leq n$  such that  $a_i = a$  (we are well aware that this notation is not strictly coherent with the rest of set-theoretic notions we use, but nevertheless it makes the descriptions of the algorithm more legible).

## 2 i-Chip as personalizable OTP generator

The way of operating of i-Chip protocol is inspired heavily by the properties of integrated circuits. Below we provide a description of the method for generating OTP with i-Chip.

The protocol discussed can have multiple variants with varying levels of complexity (for sake of simplicity and ease-of-use, we consider only decimal system in the description, but i-Chip can be extended on hexadecimal and larger systems without any difficulties). The parameters of the algorithm are denoted by four positive integers  $N, L, B, K \in \mathbb{N}$  which respectively denote

- chip size (the matrices describing both private part of the key and the challenge matrix have size  $N \times N$ );
- parameter describing OTP length,  $L \leq 10$ ;
- maximal number of *blocks* (which will be discussed later), for the sake of clarity and memorability we restrict  $1 \leq B \leq 10$ ;
- maximal block length  $K \leq 10$ ;

### 2.1 Defining private key

Having defined these parameters, the user proceeds to preparing the private part of the key. The secret is the aforementioned *image of a chip* which the user has to memorize. Consider the coordinate set  $Z := \mathbb{Z}_N \times \mathbb{Z}_N$ . The key consists of the several sequences of coordinates, which we will refer to as *escape (output/out) blocks* and *entry (input/inp) blocks*. Each escape block  $E_i := (e_{i,0}, e_{i,1}, \dots, e_{i,m_i})$  has an associated entry block  $J_i := (z_{i,0}, z_{i,1}, \dots, z_{i,l_i})$  where  $e_{i,j}, z_{i,k} \in Z$  for all  $j \leq m_i, k \leq l_i$  and  $m_i, l_i \leq K$  for all  $i \leq B$ . In addition to length constraints, there is also an injectivity condition in the scope of single entry-escape block pair, i.e. for all  $i \leq B$  the following conditions are satisfied:

- for all  $j_1, j_2 \leq m_i, j_1 \neq j_2$  we have  $e_{i,j_1} \neq e_{i,j_2}$ ;
- for all  $j_1 \leq m_i, j_2 \leq l_i$  we have  $e_{i,j_1} \neq z_{i,j_2}$ ;
- for all  $j_1, j_2 \leq l_i, j_1 \neq j_2$  we have  $z_{i,j_1} \neq z_{i,j_2}$ .

Distinct entry-escape block pairs are free to overlap though. Having placed all the escape-entry blocks in his secret, the user finally defines a sequence of  $L$  pairwise distinct coordinates  $G := (g_0, \dots, g_L) \in Z^{L+1}$ ,  $g_{j_1} \neq g_{j_2}$  for  $j_1 \neq j_2$  which do not belong to any of the escape-entry blocks. This sequence of coordinates  $G$  will be called a *generator*. In the right part of the figure below presents the exemplary secret prepared for user, while the left presents the memory representation of the private key.

## 2.2 Generating OTP

The verifier generates a  $N \times N$  challenge matrix  $C$  of random digits. To generate the OTP, the user has to collate this matrix with his private key and calculate the digits of OTP one at the time. The procedure of obtaining subsequent characters of the OTP goes as follows:

1. Let  $i = 1$ . For each  $0 \leq j \leq L$ :
  - (a) Let  $V_{inp}^j := C[g_j]$  (recall that  $g_j$  is the  $j$ -th pair coordinates of the generator  $G$ );
  - (b) Starting from  $i$ -th entry block (where  $i = 1$  for first OTP character), search  $J_i$  for the coordinates  $z_{i,k}$  such that  $V_{inp}^j = C[z_{i,k}]$ . If no such coordinates are found in  $i$ -th block, move to the subsequent input block. By  $i_j$  denote the index of the entry (*activated*) block in which appropriate (*activation*) element has been found. If search fails, i.e.,  $V_{inp}^j \neq C[z_{i,k}]$  for all  $i \leq B$ , let  $V_{out}^j = V_{inp}^j$ .
  - (c) Let  $V_{out}^j$  be any of the elements of the form  $C[e_{i_j,k}]$ ,  $k \leq m_i$ , i.e.,  $V_{out}^j$  is any of the digits from challenge matrix which are pointed by the  $i_j$ -th escape block.
  - (d) The  $j$ -th digit of the OTP equals  $V^j = (V_{inp}^j + V_{out}^j) \bmod 10$ . Let  $i = i_j + 1$  (or leave  $i$  unchanged if  $i_j$  is not defined).
2. The sequence  $V := (V^0, V^1, \dots, V^L)$  is the generated OTP. To avoid overloading the first blocks, it is recommended to resume the search for  $V_{inp}^j$  from the block next to the last searched.

For additional security, the following exceptions have been added on (b)-(c) elements of algorithm. They increase the non-linearity of the function used to obtain the elements of  $V$  which in turn increase the overall resistance of i-Chip. Let  $z_{i_j,k}$  be the coordinates on which the *activation* element  $V_{inp}^j$  was found first in the challenge matrix.

- (EXC-A) If coordinates adjacent to  $z_{i_j,k}$  belong to some entry block  $i' \neq i_j$  (i.e.,  $z_{i',k'}$  is adjacent to  $z_{i_j,k}$  for some  $k' \leq m_i$ ), add  $C[z_{i',k'}]$  to  $V_{inp}^j$ . This exception can occur up to four times if  $z_{i_j,k}$  is adjacent to four elements of other entry blocks. In the case where  $z_{i_j,k}$  belongs to more than a single block, this exception does not apply – see the subsequent exception.
- (EXC-B) Assume  $z_{i_j,k}$  belongs to more than one entry block, i.e.,  $z_{i_j,k} \in J_{i'}$  for some  $i'$ . The user is free to choose the value  $V_{out}^j$  from any element of the  $C[e_{i',k'}]$  in addition to the range of choice granted in (c).
- (EXC-C) Assume  $z_{i_j,k}$  belongs to a block  $J_i$  for which the corresponding escape block has no elements at all. In such case, the user should treat the value  $C[z_{i_j,(k+1) \bmod |J_{i_j}|}]$  as  $V_{out}^j$  (in the formula above  $|J_{i_j}|$  denotes the length of  $i_j$ -th entry block, i.e., the block in which the sought value was found). This exception can also be used in cases where the escape block is not empty if the condition occurs:  $V_{out}^j = V_{inp}^j$  as special trigger in protocol.

## 2.3 Introductory example

As from the description, the procedure of generating OTP may seem troublesome and difficult. To illustrate its actual simplicity, consider the following example. Let  $C$  be the challenge matrix as in the figure below. The next figure depicts graphically the private key defined by user.

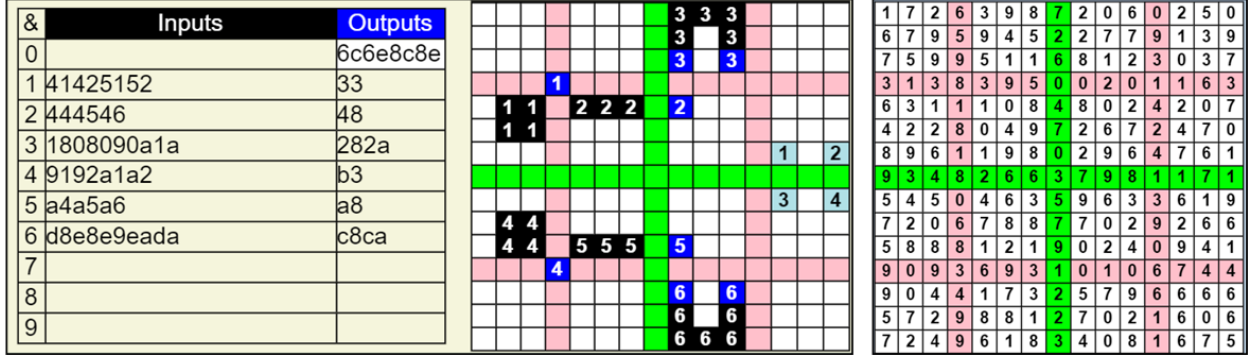


Figure 2: Exemplary secret defined by user which is compliant with algorithm parameters:  $N = 15$ ,  $B = 6$ ,  $L = 4$ ,  $K = 5$ , and randomly-generated challenge matrix of size  $15 \times 15$ . The colouring was introduced to increase legibility.

Based on Figures 2 we will compute the four-digit OTP, which would be valid for the provided challenge matrix.

- The first number we consider is under the coordinates pointed by  $g_0 = (12, 6)$ . Therefore,  $V_{inp}^0 = C[12, 6] = 7$ . We then search the first block for an entry pointing to 7 in challenge matrix. Nevertheless,  $J_0$  does not contain such entry. The problem repeats when we search through  $J_1$ , but finally  $J_2$  contains such coordinates, which are  $(12, 1)$ . Notice, that the output block  $E_2$  paired with  $J_2$  contains 2 elements. This implies that considering any of the variants:  $V_{out}^0 = 2$  or  $V_{out}^0 = 8$  will yield a valid OTP. Considering case where  $V_{out}^0 = 2$ , we calculate the first digit of OTP by adding those two digits modulo 10, which yields

$$V^0 = 7 + 2 \pmod{10} = 9.$$

- We read the value pointed by second coordinates from the generator. Thus  $V_{inp}^1 = 1$ . Starting from the entry block  $J_3$ , we search for coordinates  $z$  pointing on value 1. Since  $J_3$  fails to satisfy desired condition, we move to  $J_4$ . Luckily, two out of three coordinates from  $J_4$  point on 1. Since the corresponding escape block is a singleton, thus  $V_{out}^1$  equals 0. Finally,  $V^1 = 1 + 0 \pmod{10} = 1$ .
- Next element starts by identifying the value hidden under coordinates  $(12, 8)$  which is  $V_{inp}^2 = 6$ . We then search the subsequent entry block for coordinates pointing to a matrix entry equal to 6. Unfortunately  $J_5$  does not contains such coordinates. Since this block was the last one, we start from the beginning. Neither  $J_0$  and  $J_1$  contains any entry pointing on 6. Such coordinates are found in  $J_3$ . Once again we are left with the choice between two distinct values in the escape block. This time let us choose  $V_{out}^2 = 8$  (but replacing 8 with 2 is perfectly fine too). After this choice,  $V^2 = 6 + 8 \pmod{10} = 4$ .
- We are left with the last digit in OTP to be calculated. Last coordinates from generator point on value 9. Unfortunately, 9 is not an element pointed on by even a single entry block. Therefore, a special case described in point 2.b) is applied and  $V_{inp}^3 = V_{out}^3 = 9 + 9 \pmod{10} = 8$ .

Since the OTP was defined to be 4 digit-long, the procedure stops. Putting these values together in appropriate sequence yields password of the form 9 1 4 8. As we have mentioned throughout this description, this is not the only valid OTP for this example, as some escape blocks had multiple coordinates to choose from.

### 2.4 Examples of using exceptions (A-C)

The  $V_{inp}^j$  value is searched sequentially in all entry blocks. The entry block that contains the element with searched value we called "activated by activation element", what implies the following exceptions:

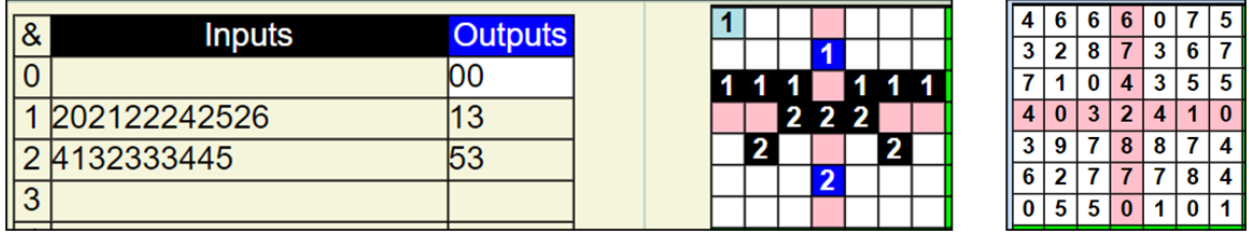


Figure 3: Fragment of key with exc-A and challenge matrix

Since the value of '4' is in the generator element, the activation element is at position (3,4) in block 2 for which the output element is (5,3) having the value '7'. But since the activation element is adjacent to element (2,4) having the value '3', then  $V^0 = (4 + 3 + 7) \bmod 10 = 4$ .

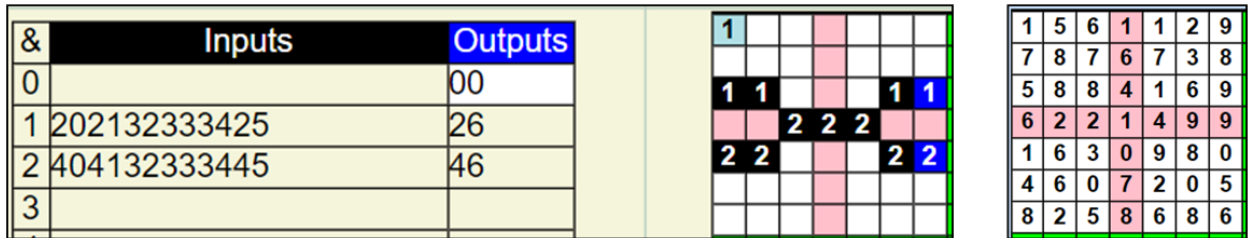


Figure 4: Fragment of key with EXC-B and challenge matrix

In above example, the input elements of blocks 1 and 2 overlap in the matrix segment at positions (3,2), (3,3), (3,4), and the value of the generator element is '1', so the element activating block 1 is in position (3,3), which is also occupied by the element of block 2. the output of these blocks combined are treated as common, then  $V_{out}^0$  takes the value from either of them, that is, from position (2,6) or (4,6). So, for example,  $V^0 = (1 + 9) \bmod 10 = 0$ , but it could be well also  $V^0 = (1 + 0) \bmod 10 = 1$ .

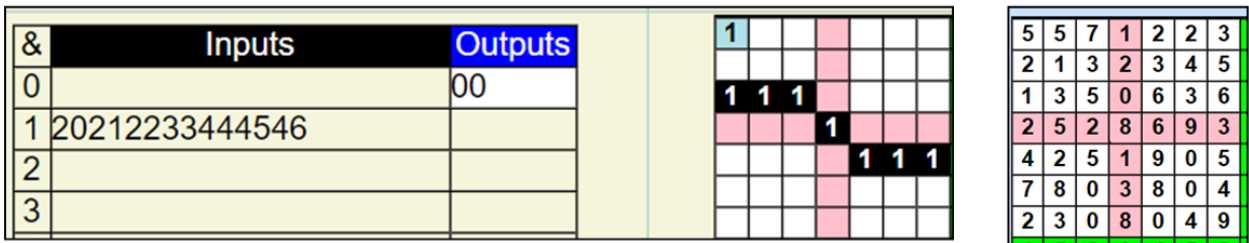


Figure 5: Fragment of key with EXC-C and challenge matrix

Since there is a value of '4' in the generator element, the activation element is at position (3,4) in block 2, for which the output element is (5,3) having the value '7'. But since the activation element is adjacent to element (2,4) having the value '3', then  $V^0 = (4 + 3 + 7) \bmod 10 = 4$ .

In this case, the order of the elements of this block is important, i.e. (2,0), (2,1), (2,2), (3,3), (4,4), (4,5), (4,6). Since in the generator element at position (0, 0) there is the value '5', which also appears in the block (activating the block) at position (2, 2), the value '8' from the next element at position (3, 3) is assumed as  $V_{out}^0$ . Hence  $V^0 = (5 + 8) \bmod 10 = 3$ . If the element activating this block were the last element in block, e.g. (4,6), then the first element of the block is assumed at position (2,0).

## 2.5 Advanced example

The previous example was designed as not to arise many nuisances. The following one is meant to present the full extent of safety offered by the proper design of secret. Consider the following private part of the key, described in Figure 6.

&	Inputs	Outputs
G		625364849382
1	242526	28
2	1a0a0b0c1c	2a2c
3	4b5b6b7b8b	9b
4	78797a7b7c	7d
5	56676869	
6	96878889	
7	c4c5c6	c8
8		
9		

Figure 6: A secret for  $N = 15$ ,  $B = 7$ ,  $L = 5$  and  $K = 6$  along with its memory representation on the left, written in hexadecimal. Pay attention to the fact, that blocks 3 and 4 overlap, while fifth and sixth input blocks have their corresponding output blocks empty.

Let us start with the first digit of the OTP password. Since block  $J_1$  contains no corresponding value to  $V_{inp}^0 = 2$ , we browse through the elements pointed by  $J_2$ . There we are able to find an element of such value, thus we are able to select an element pointed by escape block  $E_2$ . There are two such elements, choice of which results in  $V_0$  being equal to 9 or 1. Let us pick the first possibility, i.e.  $V_{out}^0 = 7$ , thus  $V_0 = 9$ .

Moving on to the next digit we search block  $J_3$ . Here,  $V_{inp}^1 = 4$  is found under coordinates adjacent to the element belonging to block 4. Thus, we apply (EXC-A) and with  $V_{out}^j = 7$  we obtain  $V_j = 7 + 4 + 0 \text{ mod } 10 = 1$ .

The similar pattern goes for the next digit of OTP. In particular, the coordinates at which  $V_{inp}^2 = 3$  can be found are adjacent to both fifth and sixth entry block. Thus, we sum up the values at the lower and upper adjacent cells of challenge matrix with  $V_{inp}^2$  and  $V_{out}^2$  pointed by the escape block  $E_4$  to obtain  $V_3 = 3 + 0 + 5 + 6 \text{ mod } 10 = 4$ .

The value attained by fourth generator cell equals 9. We search  $J_5$  in attempt to find it, but we fail. The same situation follows in sixth block and we are able to pinpoint this value in  $J_7$ . The corresponding value pointed by  $E_7$  equals 8, thus  $V_4 = 9 + 8 \text{ mod } 10 = 7$ .

The next value from the generator,  $V_{inp}^4 = 1$  cannot be found within the scope of the first block, but is pointed by the second block. Thus, the user is able to (once again) choose either 7 or 9. Letting  $V_{out}^4 = 9$ , we obtain  $V_4 = 1 + 9 \text{ mod } 10 = 0$ .

The last digit  $V_{inp}^5 = 0$  lies on the intersection of the third and fourth entry blocks. As a consequence, only exception (EXC-B) applies for this situation. Therefore, we can calculate  $V^5$  as  $0 + 7 = 7$ .

This results in the exemplary OTP for this challenge equal to:  $V = [9, 1, 4, 7, 0, 0]$ .

While this example might be a bit intimidating, it was meant to show the full capabilities of the discussed password generating protocol. In cases where the users are less willing to learn all the tricks and exceptions related to i-chip, the administrator may skip particular rules of OTP generation by enforcing additional directives during users secret creation process.

### 3 Usability of i-Chip

The composition of the secret  $S$ , thanks to the numerous symmetries offered by the pattern of elements in array and the scheme of block elements, can be remembered after only a few minutes or a quarter of an hour of repeating attempts with authentication, especially when the user has built the secret himself in the wizard, and refreshing the memory consists in wandering through all  $B$  blocks in the challenge to find and compute each of the  $L$  digits of OTP at each authentication. The time for this process, depending on the composition of  $S$  and the user's skill, ranges from 16 to 40 seconds for 4 digits response. Visual perception, after several times of scanning the table, goes into the parallel analysis mode, i.e. the location of an element from  $V_{in}$  does not run in the element-by-element mode, but in blocks, just like reading a text for a skilled reader, it is performed with whole words, not letters after letter or syllables. The user can without any special effort create any long mapping blocks, without struggling to remember them with the help of mnemonics, or remember the entire diagram of his secret in the form of only 1 mnemonic, e.g. draw the contours of a known object: plane, car, house, an envelope, or a short text or a part of it: HI, CAT, FIX, TOP, NET, ETC. People, who find it difficult to imagine the construction and operation of an electronic microchip, can design a structure and use analogies of physical components of a completely different object, such as a fragment of a city map, an architectural plan, an object made of LEGO bricks, etc. This protocol is understandable even for children from 8 years of age who associate this scheme with a board game, e.g. monopoly, and are able to add small numbers in the range  $[0,20]$  quite efficiently.

### 4 Applying i-chip to a digital signature

The challenge matrix  $C$  created by the pseudorandom number generator, is modified by adding to each of the  $C_n = |C|$  elements, one or more successive bits of the result  $H$ , from hashing with the hash function  $h$ , e.g. SHA-256, messages  $M$  and  $C$  as  $H = h(M, C)$ , according to the formula  $C'(i) = (C(i) + H_2[i]) \bmod 10$ . The number of  $H_2$  bits must be a multiple of the number of  $C_n$  elements, or their excess is ignored. The user performs his signature by entering the OTP on keyboard or writing OTP digits on the document containing this printed challenge  $C'$ , and QR code specifying the document identifier in the repository for automatic signature verification. The document may optionally also contain a TopoSign 0G diagram to mark 3 - 6 elements, as an User ID. Signature certificate includes:  $C, OTP, H$  and  $UID$ . SHA-224 can be applied to an array with  $|C| = 225 = 15 \times 15$  elements, excluding the central element, or assume that the array has 256 elements and dimensions  $16 \times 16$ , then to each element is added 1 bit from result of hashing by SHA-256 or 2 bits from result of SHA-512.

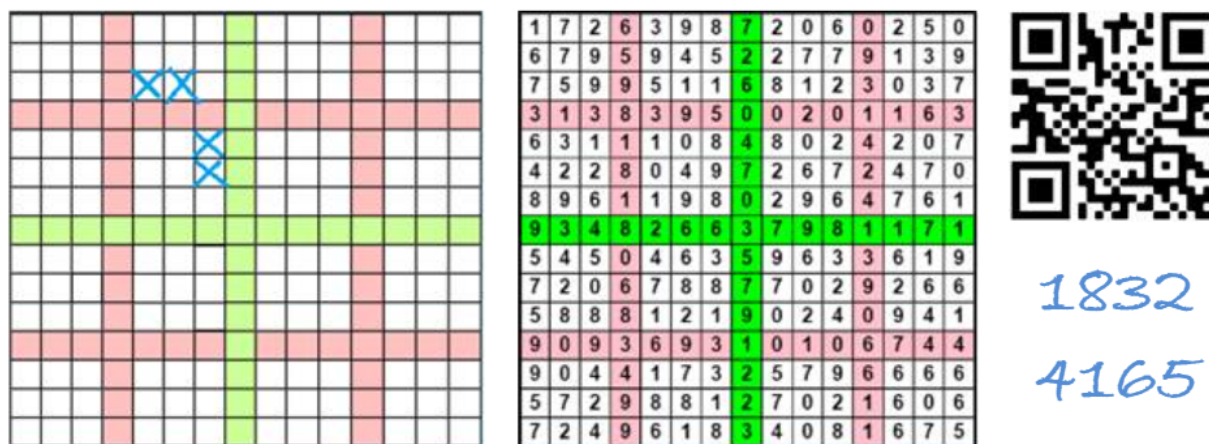


Figure 7: Offline digital signature example: User ID, hashed message as challenge, Transaction ID, OTP.



## 5 The i-Chip as common platform for other HCP protocols

The universal nature of the i-Chip protocol enables simulation or implementation of other Humanly Computable Password (HCP) protocols, as next variants of host protocol, which, however, requires specifying of special triggers that control the choice of the appropriate guest subprotocol. We will show further the implementation of the best known Foxtail and Shared Cues schemas into the i-chip OTP generator as well as new variant of blocks we called Temporary Blocks. Falsifying a handwritten signature is difficult because each person has his own handwriting style, which also presents a greater or lesser degree of difficulty. By analogy, extending the i-Chip schema to many variants is very beneficial for increasing its resistance, and paradoxically makes the scheme more user-friendly, who only needs to know the variants he chose to create his private key. When choosing, for example, 5 out of 50 variants, the cracker has to check additional 2,118,760 combinations of protocol settings, which must first be coded in such cracker, and it will continue to grow, despite the fact, that a slight subtle modification of the schema is already sufficient. We invite other researchers to compose their own licensed variants of HCP protocol, based on this common platform.

### 5.1 Implementation of Foxtail protocol

Schema proposed by Li and Shum in 2001/2002 (published as an IACR ePrint in 2005) [11].

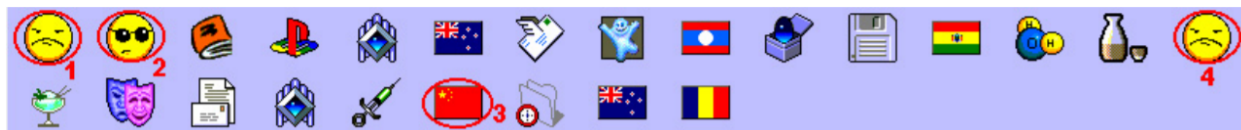


Figure 8: Example of challenge for 1 bit response

Text characters can be used instead of pictures or pass-objects location in the challenge array. There are 4 pass-objects in the challenge above, so the user compute response as  $R = 4 \bmod 2 = 0$ . The response for each challenge is defined by only 1 bit, so for the standardized 4 to 6 digits OTP length it is necessary to repeat this procedure for 16 to 20 rounds.

Properties:

Password  $P = k$  pass-objects out of  $n$  objects

Challenge  $C = l$  objects ( $l \leq n$ )

Response  $R$ :

- Count pass-objects  $P$  in  $C$ :  $\#C(P)$

- Response  $R = f_R(\#C(P))$ , e.g.  $R = \#C(P) \bmod 2$

Our proposal to implement the Foxtail protocol on the i-Chip platform is as follows: We use selected entry blocks as a chain of challenge objects window in the Foxtail schema. For binary response of Foxtail, we use the last escape block in the whole key, which must contains at least 2 (or 3 when LWR method used) elements appropriate for the expected response 0, 1 Therefore, instead of 4 protocol rounds for each bit of OTP digit, only 1 round is needed.

Objects pool and challenge can be designed in several different ways, e.g.: text characters instead of digits and additional secret of pass-characters, or by highlighting of random elements with the background color, but the best solution is to define the trigger-element, whose value (digit) specifies the counted pass-objects (digit), because it does not require any graphical changes to the original i-Chip scheme.

As trigger can function an additional element of the generator block, or we assume that if the value searched for  $V_{inp}^j$  is found in the first entry block, then all entry blocks are treated as challenge for the Foxtail scheme. Otherwise, all key blocks are treated according to the i-Chip rules.

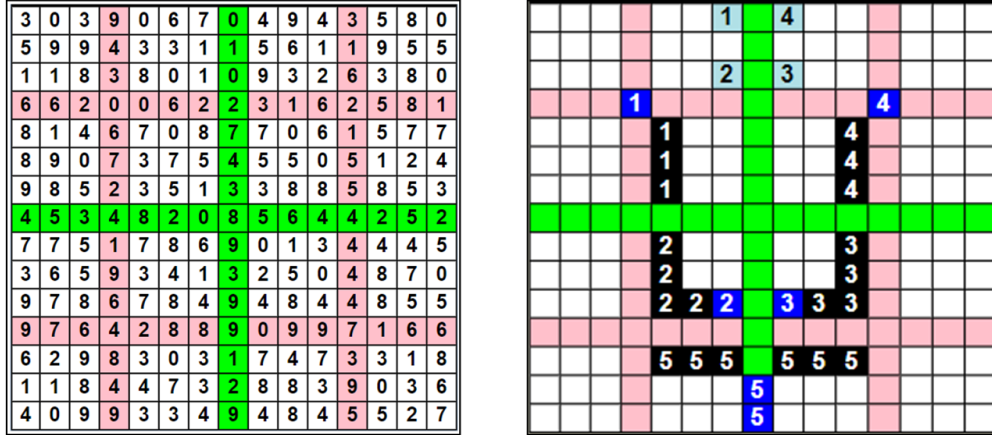



Figure 9: Example of challenge and secret for Foxtail variant of i-Chip scheme


Since the first element in the generator block  $V_{inp}^0 = 7$  appears in the first entry block, then all entry blocks are treated as a Foxtail challenge. We count all entries with digit '7', there is 5 such locations [(4, 4), (8, 4), (a, 4), (c, 8), (c, a)], so the response for Foxtail is  $5 \bmod 2 = 1$ . However, according to the new i-chip rule, we use this response to addressing of proper element from 2 locations [(d, 7), (e, 7)] in the escape block, so finally  $V_{out}^1 = (7 + 9) \bmod 10 = 6$ . The next digits in the generator block no longer appear in the first block, so i-chip rules apply to them, and whole OTP = [6, 2, 8, 8].


## 5.2 Implementation of Shared Cues protocol


Schema proposed by J. Blocki et al. in 2017 [4].

Challenge contains 14 images from c1 to c14. The first top line contains 4 pictures that the user maps on the fig. 10, respectively to: 9, 3, 4, 5. Under this line there are 2 equations that the user has to solve by substituting digits for pictures. The exception being the 1st digit in equation #2, which must be found in the code table [0, 9]. User calculated equation #1:  $(c1 + c2) \bmod 10 = (9 + 3) \bmod 10 = 2$ . The result of '2' is found in the code table, and the picture next to '2' is assigned '7', which is inserted after x to equation #2:  $(x + c3 + c4) \bmod 10 = (7 + c3 + c4) \bmod 10 = (7 + 4 + 5) \bmod 10 = 6$ .

**9**  


**3**  



**4**  



**5**  



**Computed Response:**


$$\sigma \left( \text{Lightning} \right) + \sigma \left( \text{Dog} \right) = 9+3 \bmod 10 = 2$$


$$\sigma \left( \text{Eggs} \right) + \sigma \left( \text{Golf} \right) + \sigma \left( \text{Goat} \right) = 7 + 4 + 5 \bmod 10 = 6$$


**0** 


**1** 


**2** 


**3** 

**4** 

**5** 

**6** 

**7** 

**8** 


**9** 

Figure 10: Example of Shared Cues challenge



## 6 Cryptanalysis

Since the author of this work is an electronics engineer, neither a cryptologist or even a mathematician, the analysis is not done professionally, however the theoretical conclusions and empirical results should be approximately correct. We would be deeply grateful for necessary corrections and completing the article through professionally cryptanalysis.

- The resistance to a randomize attack depends on the number of OTP digits calculated by the user. Their number  $L$  is arbitrary and depends on the needs of the authentication system. By default, it is  $L = 4$  (like a static PIN for a smartcard), or  $L = 6$  (as OTP in some e-banking systems), or  $L = 8$  (in some systems for e-documents signing). The  $L$  number does not depend on the number of generator block elements as they can be used multiple times, or never. However, if used for login, it is suggested to add the static part of the password to the calculated OTP.
- The i-chip's resistance to man in the middle attacks is ensured by the method of hashing (signing) the authenticated message with the private key, as the user-generated OTP is valid only for the signed message (see Chapter 4)
- The challenge in the i-chip protocol is generated full at random, so it is fully immune both against RIFA and against RDFA.
- The resistance against brute-force and Grover quantum algorithm provide a huge keys space, which we estimated using two methods:

1. Method to estimate of whole keys number using i/o mapping used in brute force

For the secret on figure 2, the i/o mappings table define a  $(4 \times 1 + 3 \times 1 + 5 \times 2) \times 2 = 34$  constraints i/o in the  $N \times N = 225$  matrix, so finally keys number =  $225^{(4+68)} = 225^{72} = 2e+169$ . However, the size of the challenge matrix recommended for smartphones is  $19 \times 19$ , and for 12" HD screens, there may be 18 such modules, which increases the key space considerably to  $3e+274$ .

2. Method for determining the number of meaningful keys, but without EXC-C, which increases the key space considerably, because it respects the order of the all entry block elements, calculates the following formula:  $\frac{N^2!}{(N^2-L)!} \cdot \sum_{i=B}^{B+b_0} \left( \sum_{d=1}^k \binom{N^2-L}{d} \cdot \sum_{j=C}^{C+c_0} \binom{N^2-L}{j} + \sum_{j=C}^{C+c_0} \frac{(N^2-L)!}{(N^2-j-L)!} \right)^i$

where:

$[B, B + b_0]$  = number of blocks

$[C, C + c_0]$  = length of entry blocks

$[O, k]$  = length of escape blocks

We tested the resistance of the protocol against finding the secret key with an advanced genetic algorithm, which ran for  $m=1000$  or  $m=10.000$  samples over several days on a computer with 18-core processor. The secret key in the basic configuration, without exceptions, was found after 2 hours of operation. After turning on the exception B, the search was successful only for microparameters for  $n \times n = 25$ . With the simultaneous inclusion of  $A + B$ , the 2 days search did not give a correct result even for microparameters.

## 7 Conclusions

The i-Chip can be used as common platform to implement other HCP protocols, such as Shared Cues, Foxtail, HB. The inclusion of variants in the protocol that use the LWR properties together with protocol conditions that introduce a very strong nonlinearity, empirically confirmed that they are an NP-hard problem, because several days of genetic algorithm breaking tests on a computer with an 18-core processor - failed, even for the microparameters of a  $7 \times 7$  array. On the other hand, the huge key space ensures security even when using Grover's quantum algorithm.

## 8 Acknowledgments and contributions

The authors would be deeply grateful for completing the article through cryptanalysis.

## References

- [1] Matsumoto, T., Imai, H. Human Identification through Insecure Channel. In EUROCRYPT pages 409–421, 1991.
- [2] Asghar, H. J., Steinfeld, R., Li, S., Kaafar, M. A., Pieprzyk, J. (2015). *On the Linearization of Human Identification Protocols: Attacks Based on Linear Algebra, Coding Theory, and Lattices*. IEEE Transactions on Information Forensics and Security, 10(8), 1643–1655.
- [3] Blocki, J., Blum, M., Datta, A. (2013) *Naturally rehearsing passwords*. In: Sako, K., Sarkar, P. (eds.) *ASIACRYPT 2013, Part II*. Lecture Notes in Computer Science, vol. 8270. Springer, Heidelberg
- [4] Blocki, J., Blum, M., Datta, A. (2017). *Toward human computable passwords*. Innovations in Theoretical Computer Science, ITCS 2017.
- [5] Blum, M., Vempala, S. (2015). *Publishable humanly usable secure password creation schemas*. In: AAAI Conference on Human Computation and Crowdsourcing, HCOMP, 32–41
- [6] Golebiewski, Z., Majcher, K., Zagorski, F., Zawada, M. (2008). *Practical Attacks on HB and HB+ Protocols*. In: Cryptology ePrint Archive, Report 2008/241.
- [7] van Heesch, M., van Adrichem, N. L., Attema, T., Veugen, T. (2019). *Towards Quantum-Safe VPNs and Internet*. IACR Cryptol. ePrint Arch., 2019, 1277.
- [8] Huang, Y., Huang, Z., Zhao, H., Lai, X. (2013). *A new One-time Password Method*. IERI Procedia, 4, 32-37.
- [9] Kumbhare, K. K., Warkar, K. V. (2016) *A Review on Noisy Password, Voiceprint Biometric and One-Time-Password*. Procedia Computer Science, 78(C), 382-386.
- [10] Lamport, L. (1981). *Password authentication with insecure communication*. Communications of the ACM, 24(11), 770-772
- [11] Li, S., Shum, H.-Y. (2005). *Secure Human-Computer Identification (Interface) Systems against Peeping Attacks: SecHCI*. IACR’s Cryptology ePrint Archive: Report 2005/268
- [12] Matelski, S. (2011). *Identyfikacja i wierzycielnianie tozsamości metodą podpisu topograficznego*. In: Hołyst, B., Pomykała, J. (eds.) *Metody Biometryczne i kryptograficzne w zintegrowanych systemach bezpieczeństwa*, Wydawnictwo Wyższej Szkoły Menedżerskiej, Warszawa (in Polish)
- [13] Matelski, S. (2020). *The i-Chip as One-Time Password (OTP) & digital signature generator*. IACR’s Cryptology ePrint Archive: Report 2020/1264
- [14] Mijin, K., Byunghee, L., Seungjoo, K., Dongho, W. (2009). *Weaknesses and improvements of a one-time password*. International Journal of Future Generation Communication and Networking 2(4), 29-38
- [15] Monteiro, M., Kahatapitiya, K., Hassan, J., Thilakarathna, K., Rakotoarivelo, T., Kaafar, D., Li, S., Steinfeld, R., Josef Pieprzyk (2020). *Foxtail+: A Learning with Errors-based Authentication Protocol for Resource-Constrained Devices*. IACR’s Cryptology ePrint Archive, Report 2020/261.
- [16] Weinshall, D. *Cognitive authentication schemes safe against spyware* 2006 IEEE Symposium on Security and Privacy (SP’06), Berkeley/Oakland, CA, USA, 2006, pp. 6 pp.-300, doi: 10.1109/SP.2006.10.
- [17] Asghar, H. J., Pieprzyk, J., Wang, H. (2010). *A New Human Identification Protocol and Coppersmith’s Baby-Step Giant-Step Algorithm* Applied Cryptography and Network Security, 349-366.
- [18] Ashgar, H., Matelski, S., Pieprzyk, J. (2012). *The Topographic Signature (TopoSign) Protocol*, IACR’s Cryptology ePrint Archive: Report 2020/1220

- [19] Patil, S., Mercy, S., Ramaiah, N. (2018). *A brief survey on password authentication*. International Journal of Advance Research, Ideas and Innovations in Technology, 4(3), 943-946
- [20] Samadi, S., Vempala, S., Kalai, A. T. (2017). *Usability of humanly computable passwords*. arXiv preprint arXiv:1712.03650.
- [21] Sandvoll M., Boyd C., Larsen B.B. (2015). *PassCue: The Shared Cues System in Practice*. In: Mjøl-snes S. (eds) *Technology and Practice of Passwords*. PASSWORDS 2014. Lecture Notes in Computer Science, vol 9393. Springer, Cham.
- [22] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. Lecture Notes in Computer Science, 2248, 2001.
- [23] Ari Juels and Stephen A. Weis. Authenticating Pervasive Devices with Human Protocols, volume 3621. November 2005.

## A Appendix

Below are the keys tested with the Genetic Algorithm on PC with 18-core micropocessor. Results and tools for i-Chip cracking are available here <http://i-sign.org/isign/public/ga.zip>

&	Inputs	Outputs
0		00044440
1	010203	12
2	414243	3133

1	1	1	1	2
		1		
	2	2		
4	2	2	2	3

Figure 12: Microkey with LWR case

&	Inputs	Outputs
0		00066660
1	020304	15
2	626364	51
3	2122232434	20
4	3242434445	46

1		1	1	1	2
					1
3	3	3	3	3	
4		4	3		4
		4	4	4	4
	2				
4	2	2	2		3

Figure 13: Microkey with EXC-A

&	Inputs	Outputs
0		00066660
1	202132333445	46
2	262534333241	40
3	020304	13
4	626364	53

1		3	3	3	2
			3		
1	1				2
2	2	2	2	2	2
2	2			1	1
			4		
4	4	4	4		3

Figure 14: Microkey with EXC-B

&	Inputs	Outputs
0		00066660
1	202132333425	26
2	404132333445	46
3	2212020304	14
4	4252626364	54

1		3	3	3	2
		3	3		
1	1	3			1
2	2	2	2	2	2
2	2	4			2
		4	4		
4	4	4	4		3

Figure 15: Microkey with EXC-AB