OneDigit Schema

January 26, 2016

1 Notation

Let $\mathcal{A}$ be the set of alphabet. We assume that $|\mathcal{A}| = N$. For the case of passowrd generation, $\mathcal{A} = \{A, B, \ldots, Z\}$ and $N = 26$. We denote the set of digits by $\mathcal{D}$, i.e., $\mathcal{D} = \{0, \ldots, 9\}$. Let's $\mathcal{C}$ denotes the set of possible challenges. We denote the i^{th} coordinate of a vector $\vec{u}$ by u_i.

2 OneDigit Schema

. Given a positive integer A, we define $[A] = 0, \ldots, A-1$

2.1 Preprocessing step

- Memorize a a random map $f : \mathcal{A} \to \mathcal{D}$
- Memorize a random string $s = s_1 \ldots s_{d-1} \in \mathcal{D}^{d-1}$

2.2 Processing step

Algorithm 1 OneDigit schema

Input: Challenge $c = c_1 \ldots c_l$

$g \overset{10}{\equiv} f(c_1) + \ldots + f(c_l)$

Output: Response sg

Before stating the main theorem of this note, we define the notion of strong linearly independence.

Definition 1. *We say that set of challenges $\{c_1, \ldots, c_p\}$ is strong linearly independent* (mod 10) *if $\{c_1, \ldots, c_p\}$ is linearly independent* (mod 5) *and* (mod 2)*. Note that a direct consequence of strong linear indepence is linear independence.*

Theorem 2. *Denote the output of OneDigit schema on a challenge c, by $p(c)$. We define $\mathcal{R} = \{p(c) \,|\, c \in \mathcal{C}\}$. For any challenge $c \in \mathcal{C}$ and any response $r \in \mathcal{R}$*

(a) $\Pr[p(c) = r] = \dfrac{1}{10^d}$

Furthermore, assume that we have made k observations $\big(c_1, p(c_1)\big), \ldots, \big(c_k, p(c_k)\big)$. *Then,* $\forall g_{k+1} \in \mathcal{D}$ *and* $\forall c_{k+1} \in \mathcal{C}$ *s.t.* $\{c_1, \ldots, c_k, c_{k+1}\}$ *is strong linearly independent* (mod 10)

(b) $\Pr[p(c_{k+1}) = sg_{k+1} \mid (p(c_1) = sg_1), \ldots, (p(c_k) = sg_k)] = 1/10$

Part (a) is saying that without having any prior information, the probability of guessing the correct response to any single challenge is $1/10^d$. *In other words, for any two responses* r_1 *and* r_2

$$\Pr[p(c) = r_1] = \Pr[p(c) = r_2]$$

Now assume that the adversary has observed k (challenge, response) pairs and she is trying to guess the response to a new challenge c_{k+1}. *After seeing the first (challenge, response) pair, she will know the value of s. So the only unknown part of* $p(c_{k+1})$ *is the single digit* g_{k+1}. *Part (b) is saying that for any new challenge* c_{k+1} *which forms a strong linearly independent set with k previously observed challenges, the adversary can not do better than guessing* g_{k+1} *randomly.*

Proof. (a) For any $c \in \mathcal{C}, r \in \mathcal{R}$. Let $r = r_1 \ldots r_d$

$$\begin{aligned} \Pr[p(c) = r] &= \Pr[p(c)_1 \ldots p(c)_{d-1} = r_1 \ldots r_{d-1}] \Pr[p(c)_d = r_d] \\ &= \Pr[s = r_1 \ldots r_{d-1}] \Pr[p(c)_d = r_d] \end{aligned}$$

Since each digit of string s is chosen independently at random, the above formula is equal to

$$\Pr[s_1 = r_1] \ldots \Pr[s_{d-1} = r_{d-1}] \Pr[r(c)_d = r_d]$$

The first $d-1$ probabilities appearing above are each equal to $1/10$. Thus we only need to compute $\Pr[r(c)_d = r_d] = \Pr[f(c_1) + \ldots + f(c_l) \equiv r_d \pmod{10}]$. One way to compute this probability is to count the number of maps f that satisfy

$$f(c_1) + \ldots + f(c_l) \equiv r_d \pmod{10} \tag{1}$$

and divide it by the total number of maps $f : \mathcal{A} \to \mathcal{D}$. What is the number of maps f that satisfy Eq. 2 ? One can choose $f(c_1), \ldots, f(c_{l-1})$ arbitrarily, then $f(c_l)$ will be chosen uniquely by $f(c_l) \equiv r_d - \sum_{i=1}^{l-1} f(c_i) \pmod{10}$. So the total number of choices of f will be 10^{N-l} for the letters that are not present in c, 10^{l-1} for the first $l-1$ letters in c and 1 for the last letter in c. So the total number of choices is $10^{N-l}10^{l-1} = 10^{N-1}$. Note that the total number of maps $f : \mathcal{A} \to \mathcal{D}$ is 10^N. This leads to

$$\Pr[r(c)_d = r_d] = \Pr[f(c_1) + \ldots + f(c_l) = r_d] = \frac{10^{N-1}}{10^N} = \frac{1}{10}$$

Consequently, accounting for the fixed string s

$$\Pr[r(c) = r] = \frac{1}{10^{d-1}} \frac{1}{10} = \frac{1}{10^d}$$

(b) Now assume that the adversary have observed k (challenge, response) pairs $\big(c_1, p(c_1) = sg_1\big)$, $\ldots$,$\big(c_k, p(c_k) = sg_k\big)$, and we want to compute

$$\Pr[\big(p(c_{k+1}) = sg_{k+1}\big) \mid \big(p(c_1) = sg_1\big), \ldots, \big(p(c_k) = sg_k\big)]$$

This is equal to

$$\Pr[\big(p(c_{k+1})_d = g_{k+1}\big) \mid \big(p(c_1)_d = g_1\big), \ldots, \big(p(c_k)_d = g_k\big)]$$

which is equal to

$$\frac{\Pr[\big(p(c_{k+1})_d = g_{k+1}\big), \big(p(c_1)_d = g_1\big), \ldots, \big(p(c_k)_d = g_k\big)]}{\Pr[\big(p(c_1)_d = g_1\big), \ldots, \big(p(c_k)_d = g_k\big)} \tag{2}$$

We start by computing the value of denominator. The nominator value can be achieved similarly. In order to compute $\Pr[\big(p(c_1)_d = g_1\big), \ldots, \big(p(c_k)_d = g_k\big)]$, we should count the number of mappings f that satisfy

$$\begin{cases} f(c_{11}) + \ldots + f(c_{1l}) \equiv g_1 \pmod{10} \\ \vdots \\ f(c_{k1}) + \ldots + f(c_{kl}) \equiv g_k \pmod{10} \end{cases} \tag{3}$$

Lemma 4 shows that the number of solutions to above k linear equations is 10^{N-k}. Therefore, the value of the ratio (2) is equal to

$$\frac{10^{n-k+1}}{10^{n-k}} = \frac{1}{10}$$

□

The main component of lemma 4 is to count the number of solutions to a system of linear equations mod a prime number p. Lemma 3 achieves this result.

Lemma 3. *Given a prime number p, matrix $C_{k\times N}$ with k linearly independent rows* (mod p)*, and vector $g_{k\times 1}$, the number of solutions to $Cx \overset{p}{\equiv} g$ is p^{N-k}.*

Proof. By assumption, rows of matrix C are linearly independent (mod p), thus there must be k columns $\{C^{j_1}, \ldots, C^{j_k}\}$ that are linearly independent (mod p). Let's denote the j^{th} coordinate of vector x by x_j. We claim that for any set $\mathcal{X}_{N-k} = \{x_j \in \{0, \ldots, p-1\} : j \notin \{j_1, \ldots, j_k\}\}$, there will be a unique set $\mathcal{X}_k = \{x_j \in \{0, \ldots, p-1\} : \ j \in \{j_1, \ldots, j_k\}\}$ such that $x = \mathcal{X}_k \cup \mathcal{X}_{N-k}$ is a solution for system $Cx \overset{p}{\equiv} g$.

Given an arbitrary set $\mathcal{X}_{N-k}$, let's substitute values of x_j for $j \notin \{j_1, \ldots, j_k\}$ in x and simplify the equation $Cx \overset{p}{\equiv} g$:

$$[C^{j_1}, \ldots, C^{j_k}] \begin{bmatrix} x_{j_1} \\ \vdots \\ x_{j_k} \end{bmatrix} \overset{p}{\equiv} \begin{bmatrix} g'_1 \\ \vdots \\ g'_k \end{bmatrix}$$

Where $g'_l = g_l - \sum_{i \notin \{j_1,\dots,j_k\}} C_{li} x_i$ for all $l \in \{1,\dots,k\}$.

Since matrix $[C^{j_1},\dots,C^{j_k}]$ is full rank (mod p), the above linear equation has unique solution x (mod p). So far we have proved that for every set $\mathcal{X}_{N-k} = \{x_j \in \{0,\dots,p-1\} : j \notin \{j_1,\dots,j_k\}\}$, there is a unique set $\mathcal{X}_k = \{x_j \in \{0,\dots,p-1\} : j \in \{j_1,\dots,j_k\}\}$ such that $x = \mathcal{X}_k \cup \mathcal{X}_{N-k}$ is a solution for system $Cx \overset{p}{\equiv} g$. This concludes that the number of solutions to $Cx \overset{p}{\equiv} g$ is equal to the number of possible sets $\mathcal{X}_{N-k}$ which is equal to p^{N-k}.

□

Lemma 4. *Given a function $f : \mathcal{A} \to \mathcal{D}$ and set $\{c_1,\dots,c_k\} \subseteq \mathcal{C}$ strong linearly independent and $g_1,\dots,g_{k+1} \in \mathcal{D}$, the system of linear equations 3, has 10^{N-k} solutions.*

Proof. Assume there is an ordering $a_1,\dots,a_N$ on elements of $\mathcal{A}$. Let's define the N-dimensional column vector f such that $f_i = f(a_i)$. Given challenges $\{c_1,\dots,c_k\}$, for every challenge c_i, we define the N-dimensional row vector C_i as follows. The i^{th} coordinate of C_i, is the number of occurrence of a_i in c. In this vector setting, the last system of equations will be equivalent to

$$\begin{cases} C_1 \cdot f \overset{10}{\equiv} g_1 \\ \vdots \\ C_k \cdot f \overset{10}{\equiv} g_k \end{cases} \Rightarrow \begin{bmatrix} C_1 \\ \vdots \\ C_k \end{bmatrix} f \overset{10}{\equiv} \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix} \Rightarrow Cf \overset{10}{\equiv} g \tag{4}$$

Defining

$$C = \begin{bmatrix} c_1 \\ \vdots \\ c_k \end{bmatrix}, \quad g = \begin{bmatrix} g_1 \\ \vdots \\ g_k \end{bmatrix}$$

We want to count the number of solutions to $Cf \overset{10}{\equiv} g$. Let's denote F_2 to be the set of solutions to $Cf \overset{2}{\equiv} g$ and similarly F_5 to be the set of solutions to $Cf \overset{5}{\equiv} g$. Using lemma 3, $|F_2| = 2^{N-k}$ and $|F_5| = 5^{N-k}$. We claim that for every $x \in F_2$ and $y \in F_5$ there is a unique vector z (mod 5) such that z is a solution for $Cz \overset{10}{\equiv} g$. Furthermore, for every such vector z (mod 10), there is a unique pair $x \in F_2$ and $y \in F_5$.

Given $x \in F_2$, and $y \in F_5$, we prove that there exists a k-dimensional vector z such that $z \equiv x$ (mod 5) and $z \equiv y$ (mod 2). Consider the following k systems of simultaneous congruences:

$$\begin{matrix} z_1 \overset{5}{\equiv} x_1 & & z_k \overset{5}{\equiv} x_k \\ z_1 \overset{2}{\equiv} y_1 & \cdots & z_k \overset{2}{\equiv} y_k \end{matrix}$$

Using Chinese Remainder Theorem, there exist $z_1,\dots,z_k$ satisfying the above congruences. Therefore

$$\begin{matrix} z \overset{5}{\equiv} x & \Rightarrow Cz \overset{5}{\equiv} Cx \overset{5}{\equiv} g & \\ & & \Rightarrow Cz \overset{10}{\equiv} g \\ z \overset{2}{\equiv} y & \Rightarrow Cz \overset{2}{\equiv} Cx \overset{2}{\equiv} g & \end{matrix}$$

Furthermore, for all $i \in [k]$, $z_i \pmod{10}$ is unique. Therefore $z = [z_1, \ldots, z_k]^T$ will be the unique solution to $Cf \overset{10}{\equiv} g$. So far we have shown that for every pair $(x, y) \in F_5 \times F_2$ there is one and only one vector z satisfying $Cz \overset{10}{\equiv}$. Therefore, the number of solutions to $Cz \overset{10}{\equiv}$ is equal to $|F_2||F_5| = 10^{N-k}$ □

Lemma 5. *(Chinese Remainder Theorem) Suppose $n_1, ..., n_k$ are positive integers that are pairwise coprime. Then, for any given sequence of integers $a_1, ..., a_k$, there exists an integer x solving the following system of simultaneous congruences.*

$$\begin{cases} x \equiv a_1 \pmod{n_1} \\ \quad \vdots \\ x \equiv a_k \pmod{n_k} \end{cases}$$

Furthermore, any two solutions of this system are congruent modulo the product $N = n_1 \ldots n_k$. Hence, there is a unique (non-negative) solution less than N.

2.3 HUM

In order to calculate the HUM, we first need to write the steps of Alg. 2.2 in more details

Algorithm 2 OneDigit schema

Input: Challenge $c = c_1 \ldots c_l$
Set $i = 1$, SUM$= 0$
While not EndOfChallenge :

Compute $f(c_i)$	(Applying the map)
SUM $\overset{10}{\equiv}$ SUM $+ f(c_i)$	(Add to the running sum)
$i = i + 1$	(shift pointer)

Print fixed string s
Print SUM

HUM = 2× (initialization) + l× (while loop condition) + l× (map) + l× (add) + l× (shift pointer) + (end while) + 2× (print) = $4l + 5$

2.4 Security: Q value

Theorem. 2 is saying that as long as a new challenge forms a strong linearly independent set with already observed challenges, the adversary can not predict the response to this new challenge. In order to compute the value of Q for OneDigit schema we should answer the following question:

- ◇ Given a set of challenges $\mathcal{C}$, at each round, a new challenge $c \in \mathcal{C}$ is chosen uniformly at random. Let $\mathcal{C}_i$ be the set of challenges chosen till round i. What is the maximum i such that $\mathcal{C}_i$ is a strong linearly independent set?

To answer the above question, we ran the following experiment. We chose the challenge set $\mathcal{C}$ to be the set of all the valid website names. At each iteration, our program choses a challenge c uniformly at random from $\mathcal{C}$ and checks if c along with challenges chosen so far, forms a strong linearly independent set. If yes, it saves the number of iterations as the Q value. Otherwise, it will continue. We repeated this procedure for 1000 times and took the average of all the saved Q values. The following table shows the result:

Number of trials	Q
1000	~ 18