The Topographic Signature (TopoSign) Protocol

Hassan Jameel Asghar[1], Slawomir Matelski[2], Josef Pieprzyk[1]
[1]Centre for Advanced Computing Algorithms and Cryptography,
Department of Computing, Macquarie University,
NSW 2109, Australia
{hassan.asghar,josef.pieprzyk}@mq.edu.au
[2]INTELCO co. ltd, Poland
s.matelski@intelco.pl

Abstract

This report contains analysis and discussion on different versions of the TopoSign (Topographic Signature) protocol proposed by Matelski.

1 Notation

We denote the size of the challenge by n. Unless otherwise specified, a challenge consists of a grid, which we shall also refer to as a matrix. A challenge shall be represented by the bold lower case letter $\mathbf{c}$. Each grid square (or matrix entry) is indexed by integers 1 through n. Each grid square in the challenge also contains a digit modulo a positive integer d. A grid square-integer pair is called a location. Thus, a location i has both an index and a value. The index of the location i shall be denoted by $\hat{i}$. The index belongs to the set $\{1, 2, \ldots, n\}$. The value of a location i shall be denoted by $|i|$, which takes on values from the set $\{0, 1, \ldots, d-1\}$ together with the symbol ∞. The symbol ∞ is used when a location is considered independent of a challenge, and therefore does not have a value from the set $\{0, 1, \ldots, d-1\}$. A location belonging to a particular challenge $\mathbf{c}$ shall be denoted by $\mathbf{c}[i]$. Furthermore, the index (respectively, value) of $\mathbf{c}[i]$ shall be represented by $\mathbf{c}[\hat{i}]$ (respectively, $\mathbf{c}[|i|]$).

The secret is represented by s, which is a sequence of locations. We denote the size of the secret, $\mathsf{len}(s)$, by k. Then the locations in s can be respresented by $s_1, s_2, \ldots, s_k$. Notice that without reference to any challenge, $|s_i| = \infty$. Given a secret $s = s_1 s_2 \cdots s_k$ and a challenge $\mathbf{c}$, a candidate for the secret location s_i, where $1 \leq i \leq k$, is any location j in $\mathbf{c}$ such that $\mathbf{c}[|j|] = \mathbf{c}[|s_i|]$. In other words, the two locations have the same value. The set of all candidates for the secret location s_i is denoted by C_i.

2 TopoSign 1G

For a better understanding of the reasons behind the design of TopoSign 2G, we describe a stripped version of the protocol, called TopoSign 1G. In TopoSign 1G, or 1st Generation TopoSign, the user chooses k locations as his secret. Upon receiving a challenge, the user simply outputs the values of his k secret locations in order. That is, for the secret $s_1 s_2 \cdots s_k$ and challenge $\mathbf{c}$, the user's response will be $\mathbf{c}[|s_1|]\mathbf{c}[|s_2|]\cdots\mathbf{c}[|s_k|]$. See [1] for a graphical illustration.

3 TopoSign 2G

The problem with TopoSign 1G is that an observer can find the secret locations after observing a very small number of authentication sessions. To quantify the number of authentication sessions required to obtain the secret, we attempt to look at the protocol from an adversarial perspective. To this end, we use the following notation. Let $C_i(j)$ denote the set of all candidates for the ith secret location after j sessions have been observed. Here i ranges from 1 to k. Thus, $C_1(0)$ means the number of possible locations that are candidates for the first secret location when no session has been observed. Clearly, without observing any session the adversary has no knowledge about the secret.[1] This implies that $|C_i(0)| = n$ for $1 \leq i \leq k$.[2]

After the observation of each session, the sizes of the C_i's decrease. We can measure the rate of decrease in the sizes of the C_i's using expected values. Note that the response consists of k digits, each of which is from the set $\{0, 1, \ldots, d-1\}$. Let $X_{j,l}$ denote an indicator random variable, which is 1 if the value of the location l is the same as the user's response (value of the secret location) in the jth session. Let $E[\cdot]$ denote the expected value. Then:

$$\begin{aligned}
E[|C_i(1)|] &= \sum_{l=1}^{n} E[X_{1,l}] \\
&= \sum_{l=1}^{n} \Pr[X_{l,1} = 1] \\
&= \Pr[X_{i,1} = 1] + \sum_{l=1, l\neq i}^{n} \Pr[X_{l,1} = 1] \\
&= 1 + \sum_{l=1, l\neq i}^{n} \frac{1}{d} \\
&= 1 + \frac{n-1}{d}
\end{aligned}$$

[1] We assume that the adversary cannot guess the user's secret locations other than random guess.

[2] The notation $|\cdot|$ here denotes the size of the set, and is not to be confused with the same notation used for the value of a location.

Since there are n locations, the expected number of locations in $C_i(1)$ is:

$$E[|C_i(1)|] = 1 + \frac{n-1}{d}$$

Recall that $1 \leq i \leq k$. So, the above is true for all k secret locations. For every subsequent challenge-response pair, the expected size of C_i reduces by a factor of d. This follows from the fact that the probability that the value of a location is the same as the value of the secret location in m sessions is d^{-m}. Thus after m sessions:

$$E[|C_i(m)|] = 1 + \frac{n-1}{d^m}$$

Finding a value of m for which the expected size of $C_i(m)$ is equal to 1 is cumbersome, since it requires calculating probabilities of all possible outcomes. We can, however, obtain a good estimate as follows. Observe that if a set of candidates for a secret location contains 2 candidates, one of them is definitely the secret location. Thus we estimate the value of m for which the expected size of the candidate set is less than 2. We get:

$$\begin{aligned} 1 + \frac{n-1}{d^m} &< 2 \\ \Rightarrow \frac{n-1}{d^m} &< 1 \\ \Rightarrow n - 1 &< d^m \\ \Rightarrow \log_d(n-1) &< \log_d d^m \\ \Rightarrow \log_d(n-1) &< m \log_d d \\ \Rightarrow m &> \log_d(n-1) \end{aligned}$$

Denote the above by m_{lb}. Thus, after m_{lb} challenge-response pairs we expect $C_i(m_{\mathsf{lb}})$ to have a single candidate. In other words, there is close to a 50 percent chance that $C_i(m_{\mathsf{lb}})$ has a single candidate. Thus, there is a high probability that the adversary can get at least one of the secret locations after the observation of m_{lb} sessions. With $n = 361, k = 4$ and $d = 10$, we get $m_{\mathsf{lb}} > 2.56$. This means only 3 sessions are enough to find the secret with high probability.

This can be verified with simulations. For instance, after 10,000 runs, we found that the secret can be revealed after the observation of 3 sessions, 70 percent of the time. On the other hand, the secret can be revealed with *probability* 1, after about 3.32 sessions on average. While this number is slightly higher than 3, it should be noted that the adversary can still get the secret after the observation of 3 sessions with a significantly high probability (70%). Thus, the above method of obtaining an estimate is reasonable and *safe* from a security standpoint.

If we use larger values of parameters, that is $n = 3610$ and $k = 10$, the situation does not improve much. Now we obtain $m_{\mathsf{lb}} > 3.56$, which is only a marginal improvement over the previous case (by a single session).

3.1 Changes in TopoSign 2G

TopoSign 2G is constructed so that the sizes of the C_i's decrease much slowly. To achieve this, two innovations are introduced:

- Instead of changing the value of each location uniformly at random in each challenge, only the values of a select few are changed. Details follow.
- Fuzzy responses are tolerated. To be more precise, if say d_0 is the value of one of secret locations, then the user can respond with either d_0, $(d_0 + 1) \bmod d$, or $(d_0 - 1) \bmod d$.

We first describe the version of TopoSign 2G that uses only the first change mentioned above. We begin by describing the method of generating challenges.

3.2 Generating Challenges in TopoSign 2G (First Variant)

Let s denote the secret, and for simplicity assume that $\mathsf{len}(s) = k = 1$, i.e., the secret consists of only one location. Under the new scheme, the challenges are constructed using the procedure described below. Here $\mathbf{c}$ denotes the current challenge being generated, and $\mathbf{c}_{\mathsf{prev}}$ denotes the challenge from the previous session.[3] Initially, $\mathbf{c}_{\mathsf{prev}}$ is empty.

1: **if** $\mathbf{c}_{\mathsf{prev}} = \emptyset$ **then**
2: Assign a digit uniformly at random from the set $\{0, 1, \ldots, d-1\}$ to $\mathbf{c}[|s|]$ (i.e., the secret location). Denote the digit by d_1.
3: Pick a fraction $\hbar$ of locations in $\mathbf{c}$ and assign them the same digit as $\mathbf{c}[|s|]$ (i.e., d_1).
4: Assign digits uniformly at random from the set $\{0, 1, 2, \ldots, d-1\} - \{d_1\}$ to all the remaining locations, i.e., locations not picked in the above two steps.
5: Assign $\mathbf{c}_{\mathsf{prev}} \leftarrow \mathbf{c}$.
6: **else**
7: Assign $\mathbf{c} \leftarrow \mathbf{c}_{\mathsf{prev}}$ (i.e., the new challenge is the same as the previous challenge at first).
8: Assign a digit uniformly at random from the set $\{0, 1, \ldots, d-1\}$ to $\mathbf{c}[|s|]$. Denote the digit by d_2.
9: Let $d_1 \leftarrow \mathbf{c}_{\mathsf{prev}}[|s|]$ (The value of the secret location in the previous challenge).
10: For all locations i, if $\mathbf{c}_{\mathsf{prev}}[|i|] = d_1$, then assign $\mathbf{c}[|i|] \leftarrow d_2$.
11: Pick a fraction $1 - \hbar$ of locations in $\mathbf{c}$ (excluding the secret location) and assign them digits uniformly at random from the set $\{0, 1, 2, \ldots, d-1\} - \{d_2\}$.

□

The essence of the above procedure is that even though the value of the secret location changes uniformly at random in each challenge, the number of locations

[3] Each session in TopoSign 2G consists of a single challenge-response pair.

having the same value as the secret location's is higher than what is expected when all locations are assigned digits uniformly at random. Consequently, the size of C, the set of candidates of the secret location,[4] decreases much slowly.

3.3 Calculating the Size of C

In a manner similar to the one depicted in Section 3 we can see that the expected value of C after m sessions is:

$$E[|C(m)|] = 1 + (n-1)\,\hbar^m$$

To estimate the value of m that would give a unique solution, we use the same procedure as before:

$$\begin{aligned} & 1 + (n-1)\,\hbar^m < 2 \\ \Rightarrow\ & (n-1)\,\hbar^m < 1 \\ \Rightarrow\ & \hbar^m < \frac{1}{n-1} \\ \Rightarrow\ & \log_d \hbar^m < \log_d (n-1)^{-1} \\ \Rightarrow\ & m > -\frac{\log_d (n-1)}{\log_d \hbar} \end{aligned}$$

Where the last inequality holds since $0 < \hbar < 1$, and therefore $\log_d \hbar$ is negative. Denote the above bound by m_{lb}. With $n = 361$ and $d = 10$, this means that $m_{\mathsf{lb}} = 14.16$. Thus, we expect to find a unique secret after about 15 sessions. This is a considerable improvement over TopoSign 1G (3 sessions). However, this increase comes with a price of an increased probability of success of random guess. We shall discuss this later in Section 3.7.

3.4 The Second Change in TopoSign 2G

The second change in TopoSign 2G is the introduction of fuzzy responses. More precisely, if say d_0 is the value of the user's secret location, then the user can respond with either d_0, $(d_0 + 1) \bmod d$, or $(d_0 - 1) \bmod d$. Abusing notation, we shall say that the user responds with $d_0 \pm 1$ when we mean that he replies with d_0, $(d_0 + 1) \bmod d$ or $(d_0 - 1) \bmod d$. Since the adversary does not know which is the exact digit, the expected size of C is much bigger then in the case of TopoSign 1G. To be more precise, the expected size is:

$$E[|C_i|] = 1 + \frac{3}{d} \cdot (n-1)$$

[4] We have dropped the subscript i from C since we assume that the secret consists of a single location.

And to obtain a unique solution, we get:

$$\begin{aligned} & 1 + \left(\frac{3}{d}\right)^m (n-1) < 2 \\ \Rightarrow\ & n - 1 < \left(\frac{d}{3}\right)^m \\ \Rightarrow\ & \log_d(n-1) < m(\log_d d - \log_d 3) \\ \Rightarrow\ & \frac{\log_d(n-1)}{(\log_d d - \log_d 3)} < m \end{aligned}$$

Again, denote the above lower bound by m_{lb}. With the values of $d = 10$, $n = 361$ and $k = 4$, we get $m_{\mathsf{lb}} > 4.89$. This is still not good enough, especially because the adversary's chance of a successful random guess has increased (since he can respond with $d_0 \pm 1$). Thus, we use the aforementioned two changes in conjunction.

3.5 Generating Challenges in TopoSign 2G (Second Variant)

Let s denote the secret, and again for simplicity assume that $\mathsf{len}(s) = k = 1$, i.e., the secret consists of a single location. Under the new scheme, the challenges are constructed using the under-mentioned procedure. The procedure is similar to the one described in Section 3.2, with slight changes to compensate for fuzzy responses. Once again, $\mathbf{c}$ is the current challenge being generated, and $\mathbf{c}_{\mathsf{prev}}$ denotes the previous challenge. At first $\mathbf{c}_{\mathsf{prev}}$ is empty.

1: **if** $\mathbf{c}_{\mathsf{prev}} = \emptyset$ **then**
2: Assign a digit uniformly at random from the set $\{0, 1, \ldots, d-1\}$ to all the locations.
3: Assign $\mathbf{c}_{\mathsf{prev}} \leftarrow \mathbf{c}$.
4: **else**
5: Assign $\mathbf{c} \leftarrow \mathbf{c}_{\mathsf{prev}}$ (i.e., the new challenge is the same as the previous challenge at first).
6: Let r_{prev} be the response to the challenge $\mathbf{c}_{\mathsf{prev}}$.
7: Assign a digit uniformly at random from the set $\{0, 1, \ldots, d-1\}$ to $\mathbf{c}[|s|]$. Denote the digit by d_2.
8: Let $d_1 \leftarrow r_{\mathsf{prev}}$ (The response of the user in the previous challenge).
9: For each location j such that $\mathbf{c}[|j|] = d_1$, assign $\mathbf{c}[|j|] \leftarrow d_2$. For all locations j such that $\mathbf{c}[|j|] = (d_1 + 1) \bmod d$, assign $\mathbf{c}[|j|] \leftarrow (d_2 + 1) \bmod d$, and for all locations j such that $\mathbf{c}[|j|] = (d_1 - 1) \bmod d$, assign $\mathbf{c}[|j|] \leftarrow (d_2 - 1) \bmod d$.[5]
10: Randomly pick a fraction $1 - \hbar$ of locations in $\mathbf{c}_2$ and assign them random values from $\{0, 1, \ldots, d-1\} - \{d_1 \pm 1\}$.

[5] The order of the last two assignments can be reversed.

□

If there are more than one secret locations, then for this procedure to be consistent, each secret location should have a different challenge. This can be easily done sequentially since the user responds to each location in order.

3.6 Estimating the size of C

Once again we estimate the size of C (the number of candidates for the secret) in the same way as before. It is evident from our earlier discussion that after the observation of the first session:

$$E[|C(1)|] = 1 + \frac{3}{d} \cdot (n-1)$$

And it is easy to see that for every challenge (session) $m \geq 1$, we have:

$$E[|C(m)|] = 1 + \frac{3}{d} \cdot (n-1) \cdot \hbar^{m-1}$$

Table 1 shows the size of C against different values of $\hbar$. The values of the parameters used in the calculations are $n = 19 \times 19 = 361$ and $d = 9$. Note that if all the digits in the challenge are chosen uniformly at random, then $\hbar = \frac{3}{9} \approx 0.33$.[6] As can be seen from the table, the size of C decreases much slower with an increase in $\hbar$.

3.7 Compensating for Random Guess

Due to the introduction of the above mentioned innovations, the probability of success of random guess is different in TopoSign 2G as compared to it's first generational version. To see this, suppose $\mathsf{len}(\mathsf{s}) = k = 1$. A naive form of random guess is to sample an integer uniformly at random from $\{0, 1, \ldots, d-1\}$. The probability of success of this attack is $\frac{1}{d}$, which is the same for TopoSign 1G. A better version is to randomly choose a location and then respond with the value of that location. The probability of success of this version of random guess in TopoSign 1G is:

$$1 \cdot \frac{1}{n} + \frac{1}{d} \cdot \frac{n-1}{n} = \frac{1}{n} + \frac{n-1}{n} \cdot \frac{1}{d}$$

The above is greater than $\frac{1}{d}$ for any integer $d > 1$. However, since n is considerably large, this is a marginal improvement. For instance, if $d = 10$ and $n = 361$, this gives 0.1024. The probability of success of the first random guess attack is 0.1000 with these values. The situation is different in the case of TopoSign 2G. The probability of success of random guess in the first variant of TopoSign 2G is:

$$\hbar$$

[6] This can be verified as follows. For the expected value in Section 3.4 to be the same as this one, we should have $(\frac{3}{d})^m = \frac{3}{d} \cdot \hbar^{m-1}$. This implies that $\hbar = \frac{3}{d}$

Table 1: Expected size of $|C(i)|$ against different values of p_{d_0}.

pairs	$\hbar = 33\%$	$\hbar = 50\%$	$\hbar = 66\%$	$\hbar = 75\%$
0	361	361	361	361
1	120.33	120.33	120.33	120.33
2	40.11	60.17	80.22	90.25
3	13.37	30.08	53.48	67.69
4	4.46	15.04	35.65	50.77
5	1.49	7.52	23.77	38.07
6	0.50	3.76	15.85	28.55
7		1.88	10.56	21.42
8		0.94	7.04	16.06
9			4.69	12.05
10			3.13	9.03
11			2.09	6.78
12			1.39	5.08
13			0.93	3.81
14				2.86
15				2.14
16				1.61
17				1.21
18				0.90

As an example, if $\hbar = 0.50$ then the probability of random guess is 0.50. This is considerably higher than 0.1024 for Toposign 1G. The recommended size of the secret in TopoSign 1G is $k = 4$. This means that the success of the random guess is approximately 0.0001, or one in ten thousand. To achieve a similar level, $k = 10$ can be used in TopoSign 2G. With $k = 10$, we get $(0.50)^{10} \approx 0.0009$.

For the second variant of TopoSign 2G, the success probability of random guess is:

$$\frac{3}{d}$$

If $d = 10$, this gives 0.3. To compensate for this increase, a higher value of k can be used yet again. For instance, $k = 8$ yields 0.00006.

3.7.1 Random Guess Based on the Set of Candidates

So far we have mentioned random guess without reference to the set of candidates obtained after observing several sessions. As mentioned before, after one or more sessions have been observed, the size of the set of candidates, C, decreases from n. So, the adversary can instead choose a random location from C as a better form of random guess. In TopoSign 1G, we can see that the

probability of success of this version of random guess is:

$$1 \cdot \frac{1}{|C(m)|} + \frac{|C(m)|-1}{|C(m)|} \cdot \frac{1}{d} = \frac{1}{|C(m)|} + \frac{|C(m)|-1}{|C(m)|} \cdot \frac{1}{d}$$

where m denotes the number of sessions observed. When $m = 0$, $|C(m)| = |C(0)| = n$, which is the same as observed before. However, as m increases $|C(m)|$ decreases rapidly in TopoSign 1G. We would like to use the protocol for a certain number of sessions m, such that $|C(m)| \geq 10$. As we have seen before, in TopoSign 1G the size of C decreases as:

$$\frac{n-1}{d^m}$$

From this, we can see that:

$$\begin{aligned} \frac{n-1}{d^m} &> 10 \\ \Rightarrow n-1 &> 10d^m \\ \Rightarrow \log_d(n-1) &> \log_d 10d^m \\ \Rightarrow \log_d(n-1) &> \log_d 10 + m \log_d d \\ \Rightarrow m &< \log_d(n-1) - \log_d 10 \end{aligned}$$

Denote this upper bound on m as m_{10}. If $n = 361$ and $d = 10$ we get $m_{10} = 1.56$, which means that the expected size of C is below 10 after the observation of just 2 sessions. In effect we can only use TopoSign 1G for a single authentication session if the success rate of random guess is required to be below the marker described above.

TopoSign 2G is much better in this regard due to a slower decrement in the size of C. First note that the probability of success of random guess in the first variant of TopoSign 2G is:

$$\frac{1}{|C(m)|} + \frac{|C(m)|-1}{|C(m)|} \cdot \hbar$$

And the size of $|C(m)|$ decreases as:

$$(n-1)\hbar^m$$

This gives:

$$m_{10} = \frac{\log_2 10 - \log_2(n-1)}{\log_2 \hbar}$$

With $n = 361$ and $\hbar = 0.50$, this gives $m_{10} = 5.17$. Thus, this can be used for 5 sessions. Similarly, for the second variant of TopoSign 2G, we get:

$$m_{10} = \frac{\log_2(10d) - \log_2(3(n-1))}{\log_2 \hbar} + 1$$

Again with $n = 361$ and $\hbar = 0.50$, this gives $m_{10} = 4.43$. This means that this variant of TopoSign can be used for 4 sessions.

We can increase n to 3610 [1] to increase the number of sessions. Table 2 in Appendix A shows the decrease in the size of C when $n = 3610$ and $\hbar = 0.75$ in the second variant of TopoSign 2G. With $n = 3610$ and $\hbar = 0.50$, we get $m_{10} = 8.50$ for the first variant of TopoSign 2G and $m_{10} = 7.76$ for the second variant. This means 8 and 7 sessions respectively.

3.8 Discussion

TopoSign 2G uses the following assumption for its security. Namely, it is not feasible for the adversary to make a frequency list of values of the candidate locations in $C(i)$ in a challenge in *real-time*. In other words the adversary is unable to find the values occuring the most in a challenge within a threshold amount of time τ. We can reasonably choose $\tau = 20$ seconds. Note that it does not matter if the adversary can obtain the statistic from the values offline. Such statistic will only help the adversary to further decrease the size of the $C(i)$'s, but it will not allow him to impersonate the user directly in the next session, in which a new challenge awaits. This functionality can be achieved through an implementation which perhaps uses CAPTCHAs.

An open question is whether this idea can be extended to pairs or even triples of secret locations. In other words, instead of individual locations, pairs of locations are generated with a skewed distribution. If this can be achieved, then TopoSign 2G can be used for even a higher number of sessions. This follows from the fact that, if pairs are considered, then the whole search space increases to n^2 instead of n. The same reasoning goes for triples.

From a usability point of view, TopoSign 2G is very simple as it does not require the user to do any computation other than simply typing the value of a location. Even though the number of sessions for which TopoSign 2G can be used does not seem very high, it is acceptable for such a simple authentication scheme. Other authentication systems in literature which attempt to increase the number of sessions do so at a severe price of usability.

References

[1] Slawomir Matelski. Identification and authentication of identity by topographical signature protocol. Biometric and cryptographic methods in integrated security systems (WSM), ISBN 978-83-7520-067-6, 2011.

A Expected Value of $|C|$ when $n = 3610$

Table 2 shows how the size of C decreases with increasing number of observed sessions. Note that it is not until the 26th session before the secret can be uniquely revealed. However, the number of sessions before the expected size reaches below 10 is 17. With the first variant of TopoSign 2G, this increases to

Table 2: Expected size of $|C(i)|$ when $n = 3610$ and $\hbar = 75\%$.

pairs	$\hbar = 75\%$
0	3610
1	1203.33
2	902.50
3	676.88
4	507.66
5	380.74
6	285.56
7	214.17
8	160.63
9	120.47
10	90.35
11	67.76
12	50.82
13	38.12
14	28.59
15	21.44
16	16.08
17	12.06
18	9.05
19	6.78
20	5.09
21	3.82
22	2.86
23	2.15
24	1.61
25	1.21
26	0.91

20 sessions. While it is tempting to use this value of $\hbar$, it should be noted that the probability of random guess is considerably low. Thus, for instance, if no session has been observed, the probability of random guess in the first variant of TopoSign is $0.75^{10} = 0.056$ with $\hbar = 0.75$ and $k = 10$.