Using Hamiltonian Totems as Passwords

Hervé Chabanne[1,2], **Jean-Michel Cioranesco**[3], **Vincent Despiegel**[1],
Jean-Christophe Fondeur[1],**David Naccache**[3,4]

[1] Morpho (Safran Group)
11, boulevard Galliéni F-92130 Issy-les-Moulineaux, France.
surname.name@morpho.com
[2] Télécom ParisTech
46 Rue Barrault, 75013 Paris
[3] Sorbonne Universités – Université Paris II
12 place du Panthéon, F-75231, Paris Cedex 05, France.
surname.name@etudiants.u-paris2.fr
[4] École normale supérieure, Département d'informatique
45, rue d'Ulm, F-75230, Paris Cedex 05, France.
surname.name@ens.fr

Abstract

Physical authentication brings extra security to software authentication by adding real-world input to conventional authentication protocols. Existing solutions such as textual and graphical passwords are subject to brute force and shoulder surfing attacks, while users are reluctant to use biometrics for identification, due to its intrusiveness. This paper uses Hamiltonian tokens as authentication means. The proposed token structure offers many possible configurations (*i.e.*, passwords) and is small enough to be carried on a physical keychain. After presenting our general idea, we describe an efficient algorithm to produce these tokens. Our procedure was validated by running a recognition campaign on a wide batch of synthetic samples, and experimented on prototypes manufactured using a commercial 3D-printer.

Keywords: Authentication, (Visual) Passwords, Token, 3D, Hamiltonian.

1 Visual passwords

This paper introduces a new user authentication method well-suited for mobile devices such as smartphones. In some sense, the concept described here can been seen as the illegitimate child between biometric recognition and passwords. From the first, it borrows the pattern matching algorithms that handle data, and from the second, their secrecy. The concept, called *visual passwords*, is also less privacy intrusive than biometrics while keeping most of their characteristics. Biometric systems rely on physical characteristics of an individual to identify him amongst a large population [9]. As physical characteristics are generally considered public and impossible to renew, this can raise some privacy issues as the link between the individual and the application using this biometric must be protected.

Our visual password authentication proposal relies on freely chosen objets. The underlying principle of visual passwords is quite simple:

- At registration, the user chooses *something* as password and takes a photography of it. In reference to the movie Inception, we call this image a *totem*. For instance, [19] is a good candidate totem example. The user freely choose something that he has under his hand. Obviously, the choice of the totem has to remain secret. Once chosen, the totem is sent to the authentication service,
- when the user wants to authenticate, he takes another photography of his totem for a comparison image vs image with the reference.

To increase the totem's entropy, totems can be chosen among a great variety of objects [19] or be objects that lend themselves to a sufficient diversity. We decided to take the second option for two reasons: it enables us to develop specialized recognition algorithms for the kind of totems we chose and is that we do not want to get back the passwords drawback with hard to remember totems.

There are other attemps related to our visual passwords scheme. For instance, with graphical passwords [15], the user is asked to select a certain number of images from a set of random pictures. He then must select them among some decoy pictures (see [7] for a detailed analysis of graphical passwords on mobile devices). In [11], the user creates a password by clicking on several arbitrary pixels of an image where some tolerance is accepted to correct his inaccuracy in the selection of pixels. The main difference with our visual passwords scheme is that we rely on an effective image processing algorithm for the verification step, i.e. we want to determine the degree of similarity between the image stocked during the enrollment step and the fresh one taken for the authentication purpose. This enables us to more diversity in the choice of our visual passwords.

The paper is organized as follows: Section 2 describes the Hamiltonian totems which are the main contribution of this article. Section 3 details our recognition algorithms.

2 Hamiltonian Totems

A Hamiltonian circuit is a circuit running through all the vertices of a graph. The high entropy provided by Hamiltonian graphs makes them a very suitable totem candidates.

The problem of finding a Hamiltonian circuit in arbitrary graphs (HAMPATH) is known to be NP-complete. Membership in NP is easy to verify (given a candidate solution, the correctness of a solution can be verified in linear time). We refer the reader to [10] for more information on HAMPATH.

Solving the HAMPATH problem is a special case of the famous traveling salesman problem. However, generating Hamiltonian paths for specific structures, e.g. cubes, can bedone efficiently [2, 6].

At the beginning we have thought of creating a Hamiltonian cube but the recognition algorithm did not detect inner layers when plunging inside the cube. We hence decided to limit the Hamiltonian circuit to the cube's surface, namely to its four vertical faces. Dirac's theorem on Hamiltonian cycles states that an n-vertex graph in which each vertex has degree at least $n/2$, must have a Hamiltonian cycle. In our particular case, the only constraint to obtain a solution to HAMPATH is to choose an even sized cube.

Algorithm 1 Random Hamiltonian Circuit Generator

```
 1: Input $n$ size of the cube
 2: Output Hamiltonian Totem
 3: Let $Q = Q_1, ..., Q_v$ be the $v = n(n-1)$ squares of size 2 filling the lattice of $4n(n-1)$ points.
 4: while Card($Q$) $\neq 1$ do
 5:     Choose randomly $\{a, b\} \in Q^2$ with $a \neq b$.
 6:     if $a$ and $b$ have at least one couple of neighboring parallel edges then
 7:         Break a randomly chosen couple of parallel neighboring edges, verify that they form a single Hamiltonian
            circuit and merge $c = a \leftrightsquigarrow b$.
 8:         Let $Q = Q \cup \{c\} - \{a, b\}$
 9:     else
10:         goto line 4
11:     end if
12: end while
```

The Algorithm 1 solves HAMPATH for our structure in a very short time by associating elementary Hamiltonian squares mapped on the four vertical faces of the cube. At each step two different Hamiltonian cycles in adjacent graphs and a new Hamiltonian cycle are created. The process is repeated until only one Hamiltonian cycle remains. We implemented this process in C. The code starts by filling the lattice with 2 × 2 squares, and then associates them randomly. The program ends when only one cycle is left.

The Hamiltonian cycle spreads on the cube's four vertical faces and we place two plates on the upper and lower faces to increase the structure's rigidity. This causes a loss in entropy for recognition but in return, we succeed to create a solid totem structure.

The entropy of a random Hamiltonian circuit generator $\mathcal{G}(n)$ for cubes of surface of size n is difficult to estimate, and is given by the following formula:

$$H(\mathcal{G}(n)) = -\sum_{i=1}^{u_n} p_i \log_2(p_i),$$

where u_n denotes the number of distinct circuits constructible within a cube of size n and p_i is the probability that, when queried, $\mathcal{G}(n)$ will output the circuit number i. However, this definition is of little use since to the best of our knowledge, there are no estimates of u_n in the literature. An efficient security analysis of our solution requires an estimation of the key space, *i.e.*, of the number of possible Hamiltonian circuit configurations.

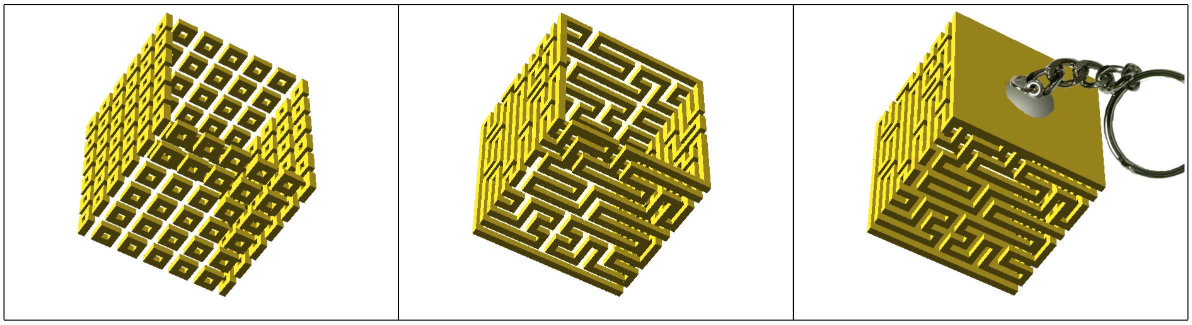

Figure 1. The three steps of the token generation

Figure 1 shows the three steps of the token generation, we used OpenSCAD [18] for generating 3D-printable files of our totems.

We first filled the four vertical faces with elementary squares, then we associated them randomly using the square association algorithm [2]. Finally, we added the upper and lower faces of the cube to improve totem rigidity.

3 Recognition Algorithm

A generic feature extraction (FE) algorithm could be implemented to encode Hamiltonian cubes. A generic FE algorithm classically relies on extracting characteristics points. These points are usually located on high gradient areas. The nature of the descriptors extracted at these points could vary but they must usually have the following properties: relative invariance to scale, rotation, translation and illumination. The literature is extremely rich on this topic, see for instance, [12, 13, 16]. Points must be discriminant and their relative positions must be compared using classical algorithms that compute the deformation between clouds of points.

Nevertheless, better performances should be obtained by resorting to FE algorithms specific to the problem to be solved. Hamiltonian cubes are objects with multiple constraints that could be leveraged to improve robustness to the extraction process.

The FE problem on Hamiltonian cubes in a video can be decomposed as follow:

- Select of the best representative for each side of the cube and geometry correction.
- Binarize each and every side of the cube
- Error correction considering constraints on the Hamiltonian path

As we are working with a synthetic data (the background is stable, the movement of the cube is regular and controlled), the first FE step is relatively simple. A combination of Hough transform on the gradient map computed by using Sobel filter and a simple frame binarization using Otsu's algorithm [14] combined with a morphological operation (closing filter to remove noise), is used to determine on which frame each face of the cube should be extracted. Similar tricks such as those used for QR codes (patterns easily detected, specific colours used for the corners and the connection on the edges of the cube) [17] could be used to ease the rectification process in real conditions.

As soon as the face's corner is accurately detected, a homograhy is calculated to rectify the face to its frontal view by using DLT algorithm [8].

A grid is set on the cube to try and determine if each grid element of the grid is filled or not. The result of Otsu's algorithm rectified with the homography is used to vote for each element.

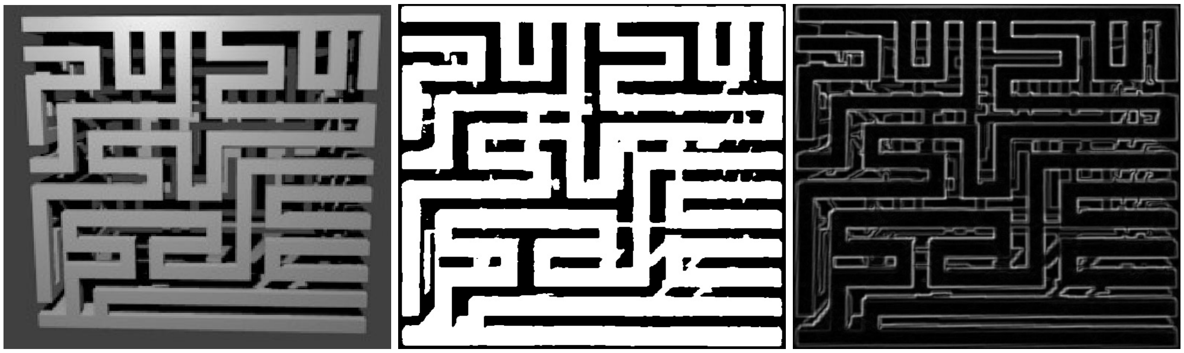

Figure 2. Selected image. Binarized image after rectification. Gradient image after rectification

The Hamiltonian path that we want to extract is not random. This means that not every configuration is possible. Masks of possible configurations are used to correct wrongly detected elements. In particular, every vertex should be connected to exactly two neighbors. A greedy algorithm is used to vote for edges which are unambiguous and determine which remaining edges are impossible and which ones are still possible. Ambiguous edges are determined at the end of the process when only a few possibilities remain. More sophisticated algorithms could employed and the choice could be made to balance between object entropy and robustness of the feature extraction process.

As soon as the Hamiltonian paths are extracted on each face of the cube, we have extracted 4 binary vectors that can be compared using Hamming distance. In real conditions, to tolerate occlusion, the chosen comparison function could be the one used classically to compare iris codes [5], taking only into account only good quality areas where the cube was robustly extracted.

4 Tests on Synthetic Data

We run an acquisition campaign on 100 samples (Fig. 3), to avoid printing each token, we generated videos of the rotating samples. An algorithm has been developed using the opencv function to evaluate the approach's validity.

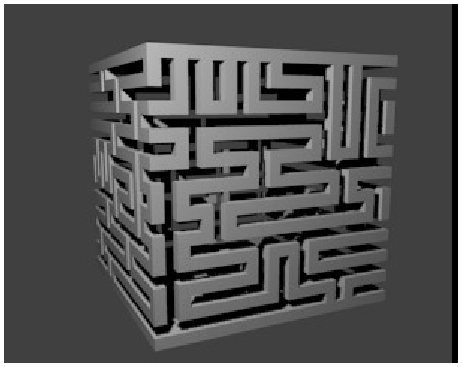

Figure 3. Snapshot from one the acquisition campaign's video

On synthetic data, the algorithm developed in the previous section leads to perfect extraction performances. No error on the Hamiltonian path extracted was observed on a synthetic database of 100 different samples. This mean that in this specific campaign, a threshold of 0 on the hamming distance could be used to separate perfectly genuine and impostor tests and lead to a False Reject Rate of 0% compared to a False Accept Rate of 0%. However, the difficulty of the extraction process should be evaluated in real conditions to determine the real performances and find out how to customize the tokens to improve robustness in difficult real-life acquisitions conditions (*e.g.* uncontrolled lights, non-uniform background, uncontrolled acquistion scenarios).

5 Prototyping

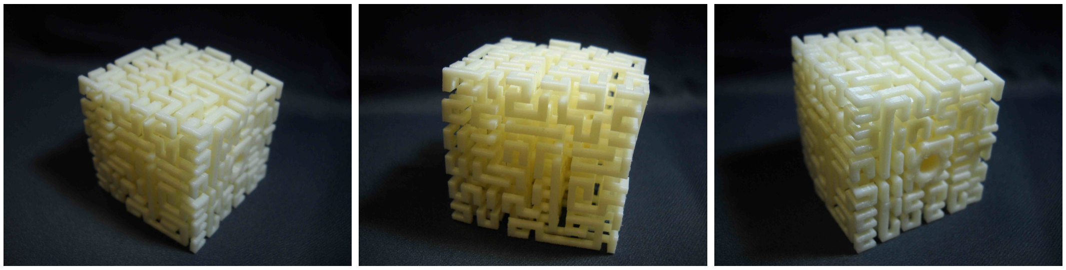

Figure 4. Some examples of totem prototypes generated by our 3D-printer

6 Further Research

The experiments done so far merely use the cube as a collection of four independent QR-codes. This does not exploit all the potential complexity of 3D structures. The authors are currently working on how to design hamiltonian cubes to be able to extract more information than the surface. This could be done by playing on the transparency, the thickness and the color of the edges as a function of the depth.

Further ideas could be the generation of opaque keys containing a complex 3D form that could be detected using x-rays or other 3D scanning techniques. We can also think to extend our work from Hamiltonian totems to artifical fingerprints [3, 4] (not necesseraly with a realistic fingerprint texture, *i.e.* which corresponds to a real human fingerprint) to exploit the fingerprint sensor technology which comes with new smartphones. Whenever the texture artificially generated (ridges frequency, precense of bifurcations and endings) is sufficiently similar to real fingerprints, then the underlying feature extractor and comparison algorithm will enable us to replace fingerprint authentication by authentication with these new totems. We here assume that we can find a suitable material [1] in order to counter the integrated liveness detection technology to have an image of the totem inside the smartphone of sufficient quality.

References

[1] C. Barral, A. Tria: Fake Fingers in Fingerprint Recognition: Glycerin Supersedes Gelatin. Formal to Practical Security 2009: 57-69. 2008

[2] S. Briais, S. Caron, J.-M. Cioranesco, J.-L. Danger, S. Guilley, J.-H. Jourdan, A. Milchior, D. Naccache, and T. Porteboeuf. 3D Hardware canaries. In *CHES*, Lecture Notes in Computer Science. Springer, 2012. Full version in ePrint Archive, Report 2012/324 (`http://eprint.iacr.org/2012/324/`).

[3] R. Cappelli, D. Maio, D. Maltoni, A. Erol: Synthetic Fingerprint-Image Generation. ICPR 2000: 3475-3478.

[4] R. Cappelli, D. Maio, D. Maltoni: Synthetic Fingerprint-Database Generation. ICPR (3) 2002: 744-747.

[5] J. Daugman, How iris recognition works, circuits and systems for video technology, IEEE Transactions on Circuits and Systems for Video Technology, 14, 1 (2004), pp. 21–30.

[6] A. Dharwadker, *The Hamiltonian Circuit Algorithm*, Proceedings of the Institute of Mathematics (Gurgaon), 2011 (`http://www.dharwadker.org/hamilton/`).

[7] P. Dunphy, A. P. Heiner and N. Asokan, A closer look at recognition-based graphical passwords on mobile devices, In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS 2010.

[8] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press 2000.

[9] A. K. Jain, P. Flynn and A. A. Ross, *Handbook of Biometrics*, Springer 2007.

[10] R.M. Karp, Reducibility among combinatorial problems, In *Complexity of Computer Computations*, New York Plenum (1972), pp. 85–103.

[11] D. Kirovski, N. Jojic and P. Roberts, Click passwords, In SEC 2006, pp. 351–363.

[12] D. Lowe, Distinctive image features from scale-invariant keypoints, Int. Journal of Computer Vision, 60, 2 (2004), pp. 1150-1151.

[13] K. Mikolajczyk and C. Schmid, Scale & affine invariant interest point detectors, Int. Journal of Computer Vision, 60, 1 (2004), pp. 63–86.

[14] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man and Cybernetics, 9, 1 (1979), pp. 62–66.

[15] X. Suo, Y. Zhu and G. S. Owen: Graphical passwords: a survey, In *Proceedings of 21st Annual Computer Security Applications Conference*, Tucson 2005, pp. 463-472.

[16] Z. Tu and A. L. Yuille, Shape matching and recognition: using generative models and informative features, In *Proceedings of the European Conference on Computer Vision, ECCV*, vol. 3 (2004), pp. 195–209.

[17] ISO/IEC 18004:2006 Information technology – Automatic identification and data capture techniques – QR Code 2005 bar code symbology specification.

[18] `http://www.openscad.org/`

[19] `http://www.madmoizelle.com/rubriques/vis-ta-vie/la-main-dans-le-sac`.