Visual Image Authentication

Michael Stephen Fiske

Aemea Institute
mf@aemea.org

IHIET-AI
Lausanne, Switzerland
April 22, 2022

Table of contents

Limitations of Malware Detection

- Malware enables breaching of computer systems.
- Cybersecurity solutions focus on malware detection.[1]
- Detection faces fundamental limits in Computer Science:
  1. No computer algorithm can detect all malware.[2]
  2. NP problems[3] (traveling salesman) hide the malware.[4]

---

[1] Nwokedi Idika and Aditya P. Mathur. A Survey of Malware Detection Techniques. Technical Report. Purdue University. February 2007.

[2] E. Filiol. Computer Viruses: From Theory to Applications. Springer, 2005.

[3] Stephen Cook. The P versus NP Problem. Clay Math Institute, 2000.

[4] Eric Filiol. Malicious Cryptology and Mathematics. Cryptography and Security in Computing. Chapter 2. Intech, March 7, 2012.

Use Hard AI Problems that People Understand

- Use hard AI problems that are easy for people to understand.
- Human visual intelligence still surpasses AI visual recognition.[5]

1. Distracting Objects.
2. Visual Occlusions.
3. Undefined or Ambiguous Problems.

[5] Maithilee Kunda. AI, visual imager, and a case study on the challenges posed by human intelligence tests. PNAS. Volume 117. No. 47. pages 29390–29397, November 24, 2020.

Distracting Objects

If **train** is the object, then clouds and birds are distracting objects.

If **bird** is the object, then clouds and a train are distracting objects.

If **clouds** are the object, then birds and a train are distractions.

Visual Occlusions

If the object is **cowboy hat**, then the helicopter occludes the hat.

If the object is **gorilla**, then the bottle occludes the gorilla.

An Ambiguous Image using Distraction and Occlusions

In the image below, is the object an animal that flies?

A machine that flies? An animal that walks?

An object that cuts?

Visual Image One-Time Passcodes[6]

Alice requests to login to her bank account.

Alice receives an image passcode via email or SMS:

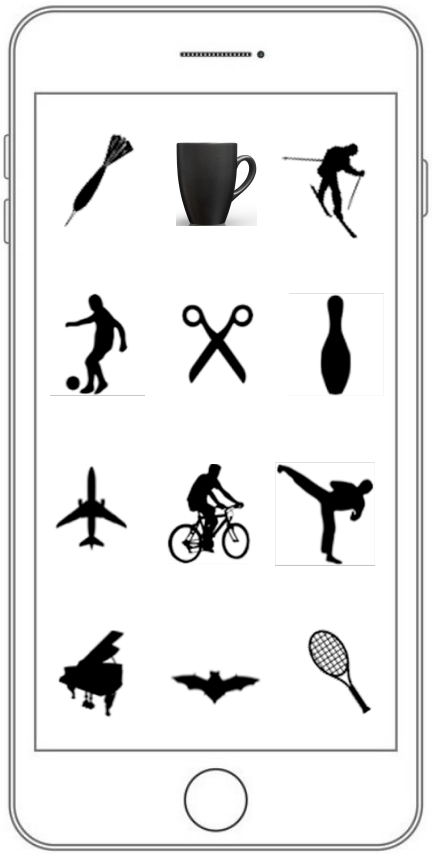
[6] Michael Stephen Fiske. US 10,592,651. March 17, 2020.

Display Screen 1

Alice must select the elephant with her fingers or mouse:

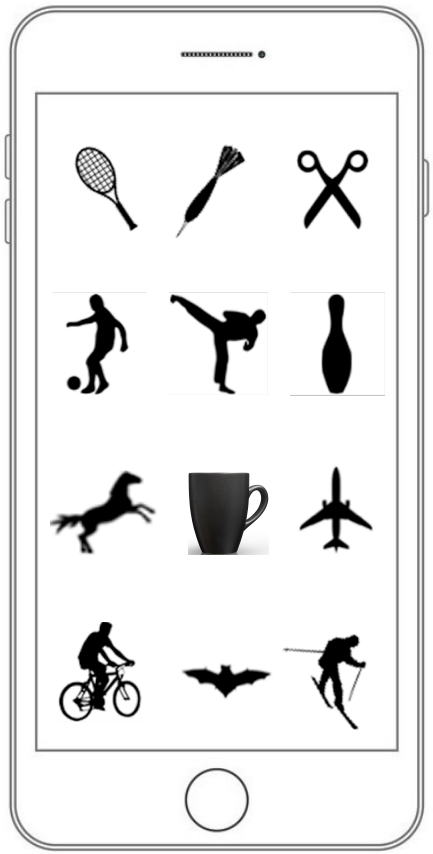
Display Screen 2

Alice must select the piano with her fingers or mouse:



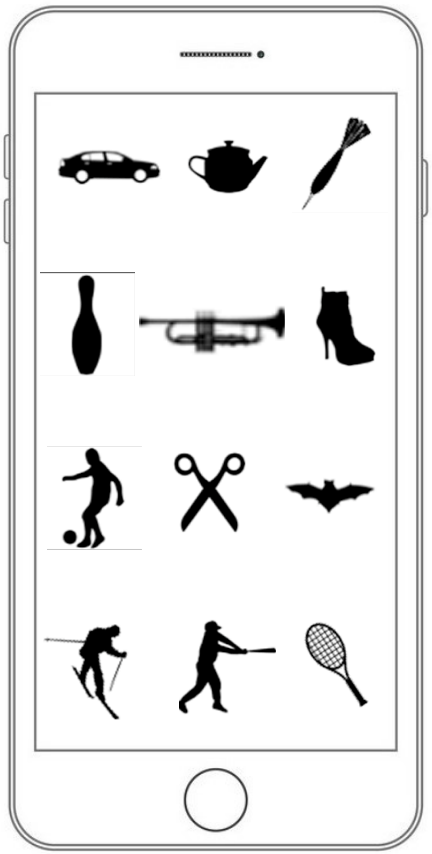
Display Screen 3

Alice must select the horse with her fingers or mouse:



Display Screen 4

Alice must select the trumpet with her fingers or mouse:



Real-time Visual Image One-Time Passcodes

In subsequent visual image OTPs, object categories can change.

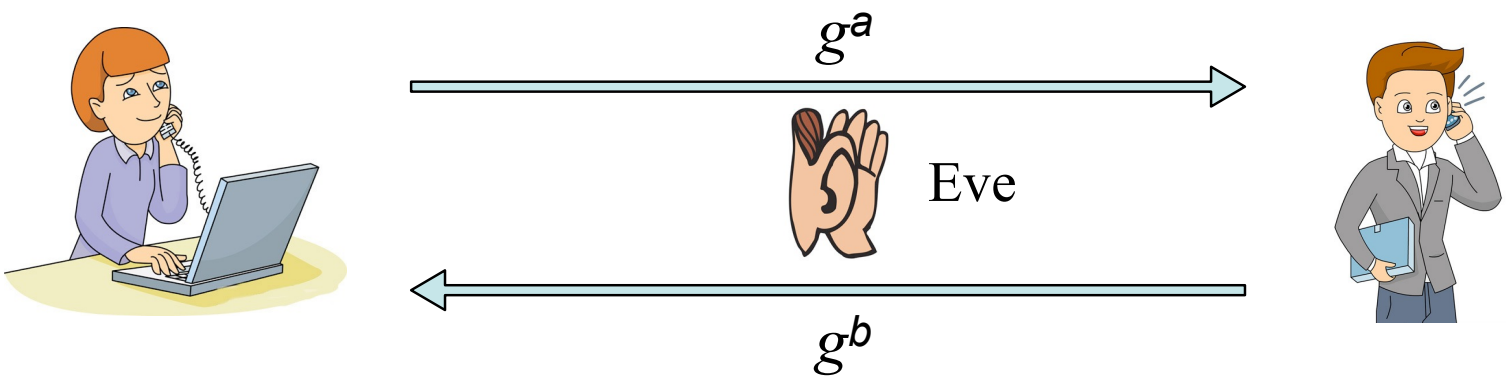
Below is a visual OTP using outlines, rotations and occlusions:

Real-time Demonstration: `https://nps.cryptografx.eu`

Detecting Eve-in-the-Middle Attacks

Alice and Bob perform a key exchange.[7]

Alice and Bob do not trust a central authority.



Alice computes $(g^b)^a$.

Alice's private key a.

Bob computes $(g^a)^b$.

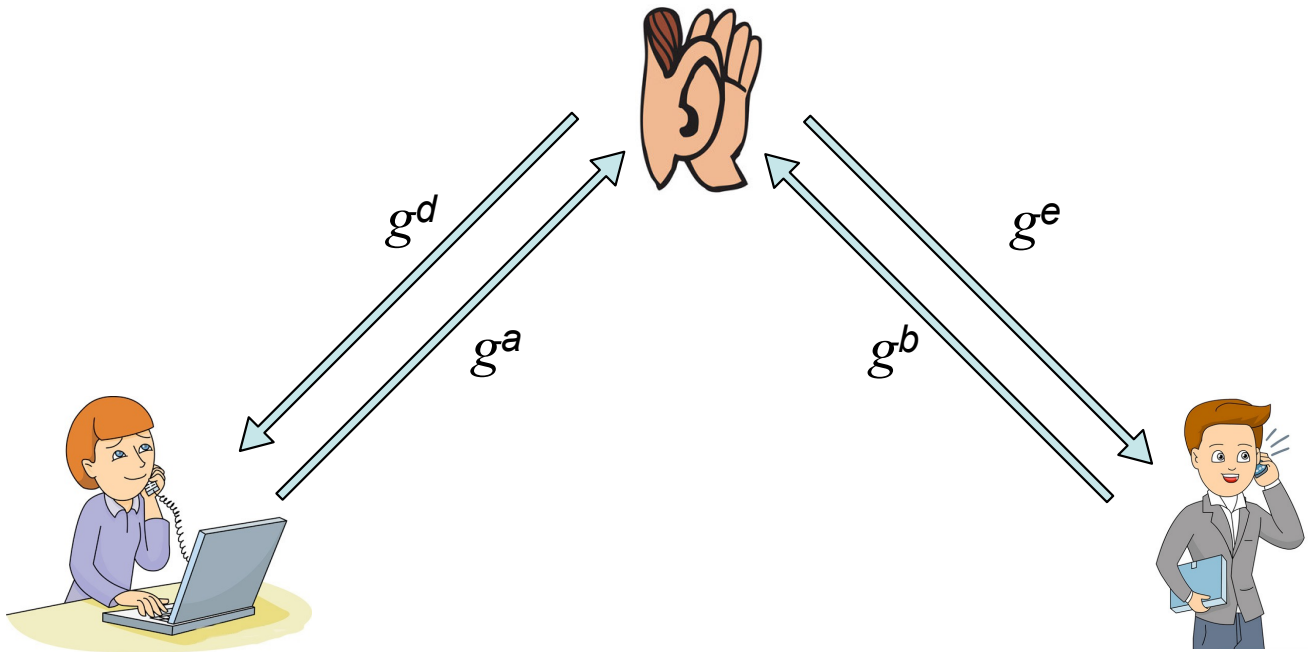
Bob's private key b.

[7] g is an element of an abelian group, agreed upon by Alice and Bob.

An Eve-in-the-Middle Attack on a Key Exchange[8]

Eve intercepts Alice's public key g^a and Bob's public key g^b.

Eve sends public key g^d to Alice and public key g^e to Bob.



Eve and Alice share secret g^{ad}. Eve and Bob share secret g^{be}.

[8] A. C. Geary. Analysis of a Man-In-The-Middle-Attack on the Diffie-Hellman Key Exchange. Protocol. Naval Postgraduate School, 2009.

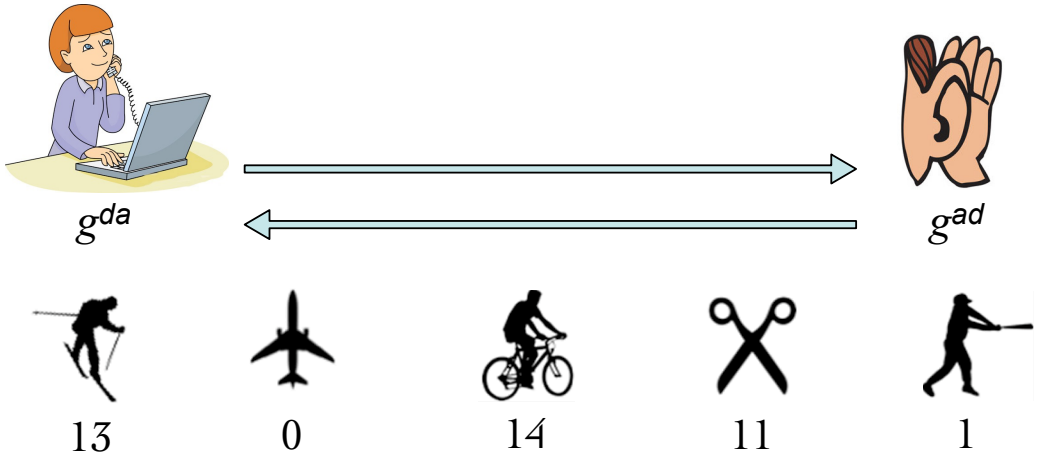
An Eve-in-the-Middle Detection Algorithm

- ϕ is a one-way hash function.[9]
- Visual encoding $f(b_1 \ldots b_n)$ chooses the ith image as b_i mod 16. If $b_1 = 7$, then the first image is horse.

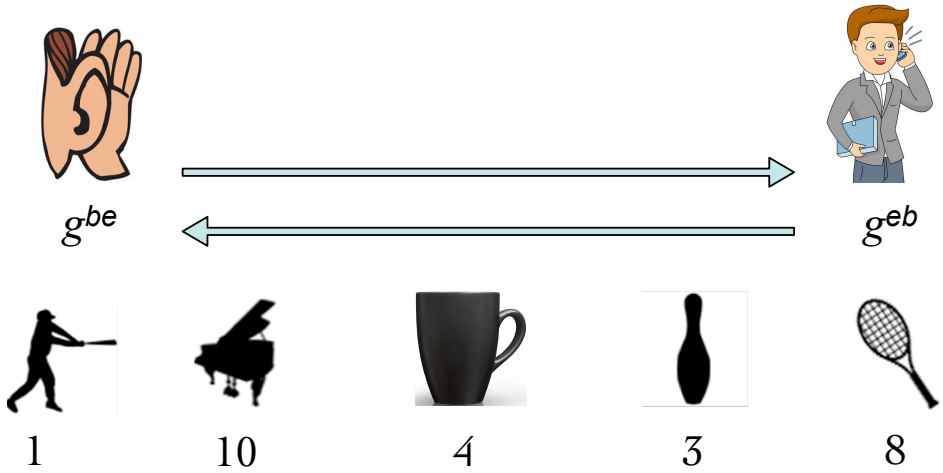
- Alice and Bob compare their visual images (typically, 5 images) derived from $f \circ \phi(g^{ab})$.

[9] NIST. FIPS-180-4. Secure Hash Standard. March 2012.

Example: Alice and Bob Compare Their Images



They don't match: an Eve-in-the-Middle attack occurred.



Protected with Standard Cryptographic Primitives

Our cryptographic library is coded in ANSI C:

- Elliptic curve 25519 implements our public key exchange and digital signatures.[10] [11]
- Our one-way hash functions are implemented with SHA-512.[12]
- Our symmetric cryptography is implemented with AES-256.[13]

[10] Daniel Bernstein. Curve25519: new Diffie-Hellman speed records. Public Key Cryptography. LNCS 3958. New York, Springer. 207-228, 2006.

[11] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, Bo-Yin Yang. High-speed high-security signatures. Journal of Cryptographic Engineering. 2, 77-89, 2012

[12] NIST. FIPS-180-4. Secure Hash Standard. March 2012.

[13] NIST. Advanced Encryption Standard (AES), FIPS 197. November, 2001.

Acknowledgments

We thank Darren Sullivan for his outstanding software development efforts.

We thank Mark Tuttle for sponsoring Darren's software development efforts.