A User-Friendly Approach to Human Authentication of Messages

Jeff King

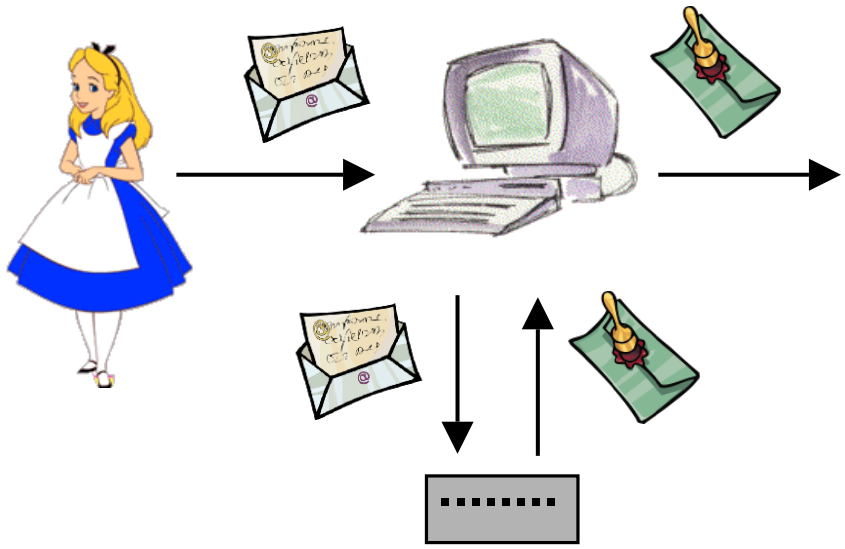
André dos Santos

Georgia Institute of Technology

{peff,andre}@cc.gatech.edu

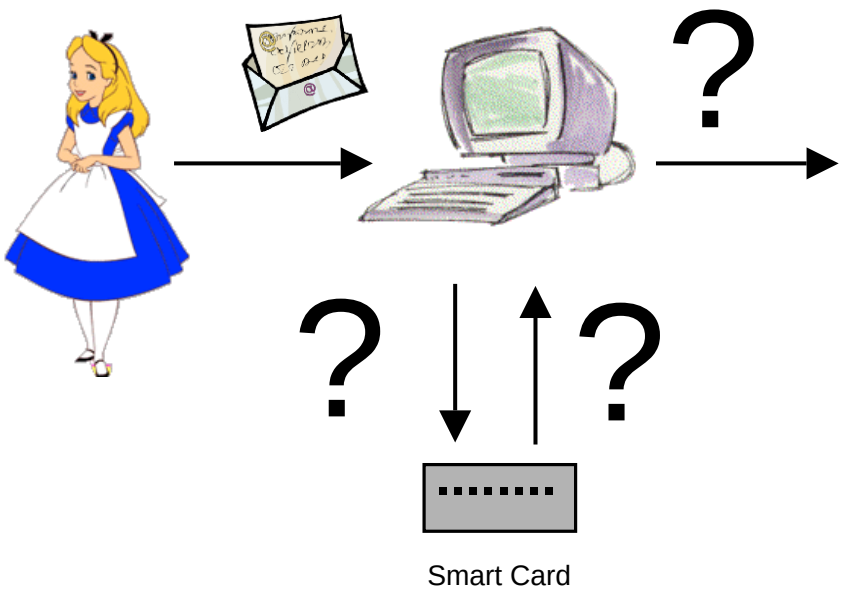
Motivation

- Suppose Alice is on a trip to a computer security conference.
- Alice has to digitally sign an important document to send to a colleague back at the office.
- She uses one of the conference's computers to, using her smart card, sign the document and send through e-mail.



Problem Solved?

- Did she really sign the document?
- What did the smartcard actually receive?
- How does Alice know it's the same thing that was on her screen?
- How does she know if anything was even sent to her card?



The Problem

- How can a human interact with a (remote) Trusted Computing Base using an untrusted computing system?

Solutions

- Use a trusted computing system to interact
  - availability?
  - security perimeter?
- Directly interact with TCB
  - extra hardware?
  - complexity and tamper resistance?
- Require the human to be a trusted computing system!

TCB to Human Secure Channel Requirements

- Human must recognize some unique secret held only by TCB (authenticity)
- Inseparable binding of secret to message contents (integrity)
- Easy for a human to do without computation or memory aid

Outline

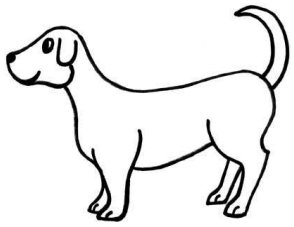
- Motivation
- Definition of Keyed Hard AI Problems (KHAP)
- KHAP Using 3-D images
- KHAP-Based Protocol
- Conclusion
- Future Work

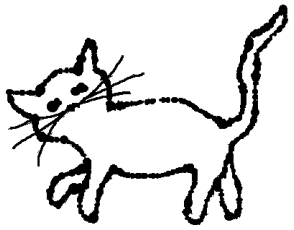
Hard AI Problems

- Informally, something that humans can do easily but computers can't.
- More formally (von Ahn, 2003)
  - S - a set of problem instances
  - f - a function mapping instances to answers
  - For human H, $H(x) = f(x)$ with high probability
  - Security parameters - (α, τ)-hard
  - For any algorithm A running in time τ, $Pr[A(x) = f(x)] \leq \alpha$
- CAPTCHA - Completely Automated Turing Test to Tell Computers and Humans Apart
- Generate random message, transform it, ask human to repeat it

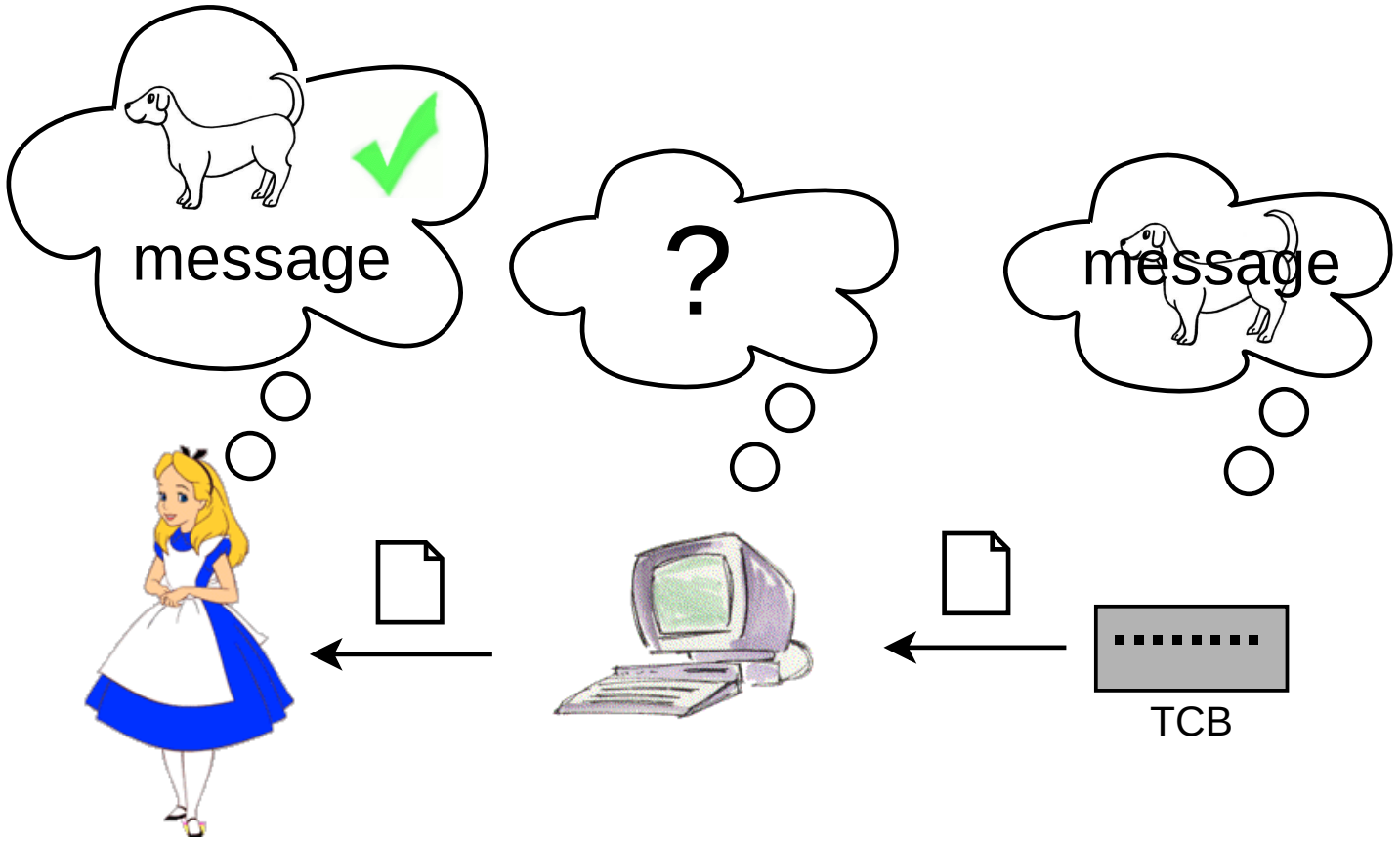
KHAP: Keyed Hard AI Problems

- A transformation problem that includes a shared secret key
- Instances generated with different keys are **distinguishable**
- Computers can't steal keys from messages
- Formalisms (simplified):
  - $H_d(m, m')$ - human distinguishes between messages with different keys
  - $|k - k'|$ - difference between two keys (quantifiable?)
  - Security parameters - $(\alpha, \varepsilon, \tau)$-hard
  - Given $|k - k'| > \varepsilon$, $Pr[H_d(m, m')] > \alpha$
  - For any algorithm A running in time τ, $Pr[H_d(m, A(m', m))] > \alpha$

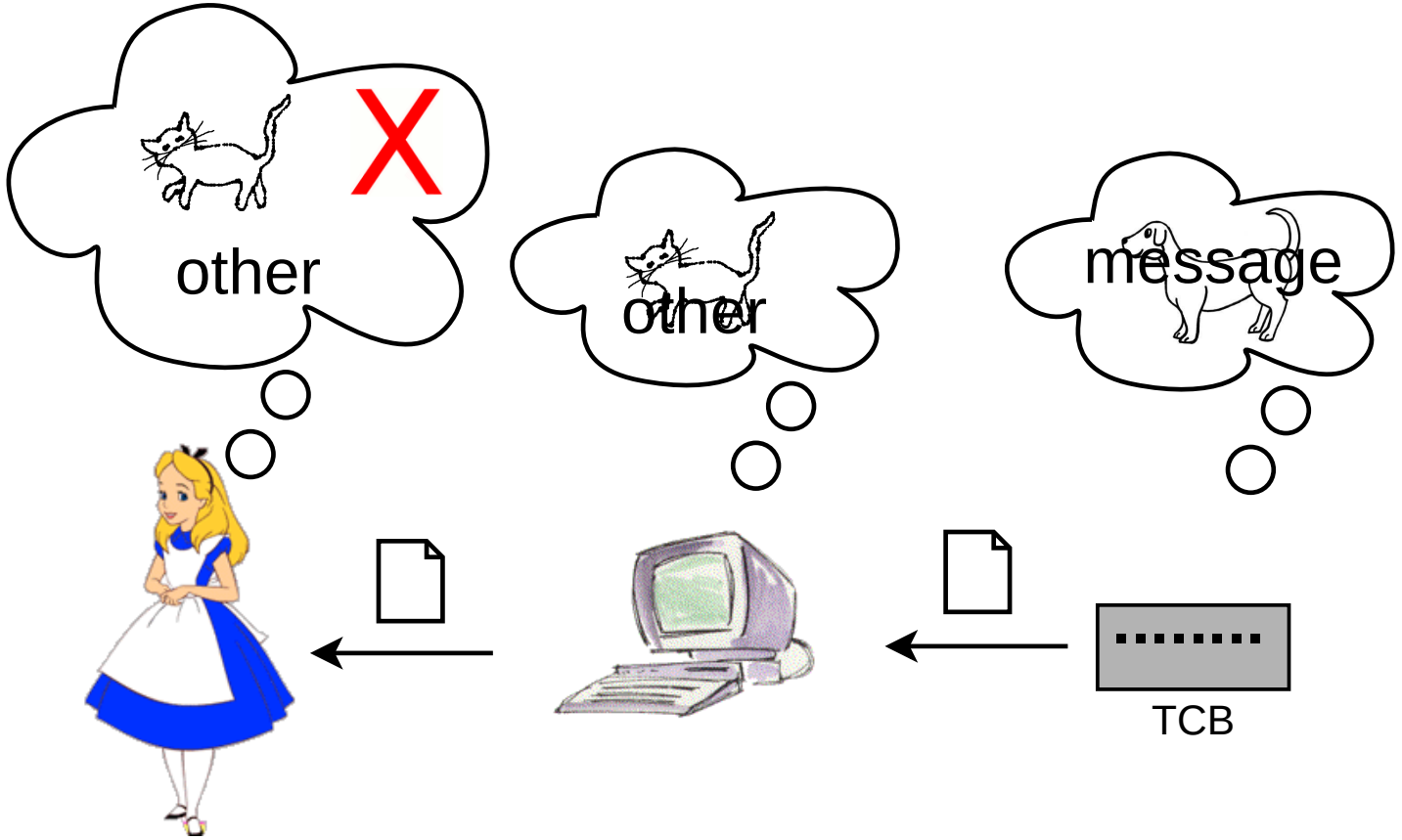




KHAP: Checking Message Authenticity



KHAP: Checking Message Authenticity



KHAP: Parameters

- Desirable:
  - Easy for a human to understand the message.
  - Easy for a human to distinguish messages with different keys.
  - Difficult for a computer to “break”
    - Change message meaningfully
    - Extract keys
    - Extract message?
- Parameters define difficulty and easiness
- Different applications have different parameter requirements
- Problem: how to evaluate parameters for a given problem
- Solution: empirical testing

3-D Keyed Transformation

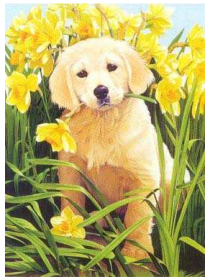
- Render text and objects in a 3-D scene to 2-D image (raytrace)
- Randomize parameters (lighting, position, rotation, size, colors)
- Human can read text from 2-D image
- Key is appearance of certain objects
- Human looks for particular objects in scene
- Scene is hard to modify in a meaningful way (shadows, reflections, finding objects)
- Provide authenticity (presence of keys) and integrity (modifications can be detected by human)

3-D Example

Attacks

- Key Guessing
- Convert from 2-D to 3-D
- Extract Key
  - Only one perspective
- Modify 2-D message
- Replays
- Human Adversary
  - The most powerful
  - May not be able to describe key and/or modify scene in time
  - Beauty of the approach: The intended recipient does not have to describe the key!

Easy to guess keys?



vs

Tradeoffs

Easy for humans

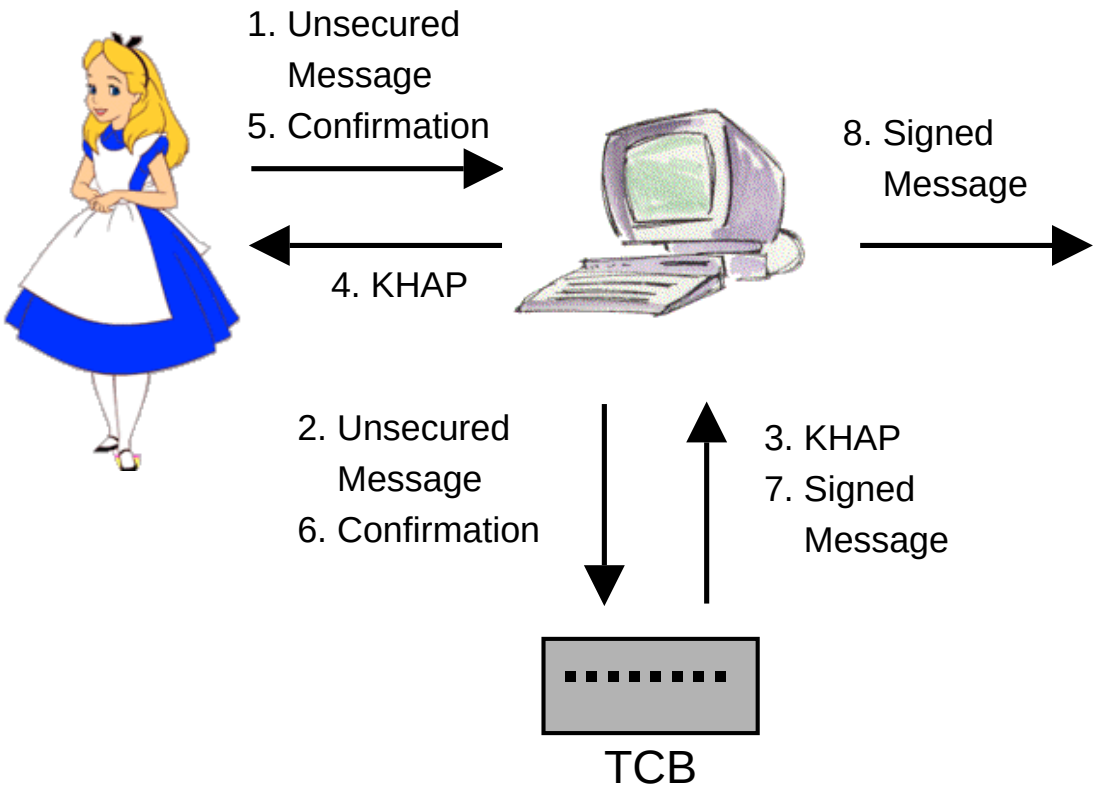
Difficult for Computers

- Easy for humans to recognize message
  - Choice of parameters (text size, fonts, colors, etc) needs to be bounded
  - Message length limitations
  - Maybe by itself static 3-D is not a good domain (animation?)
- Difficult for computers to fake image or extract key
  - Mirror reflections and shadows make cut and paste difficult
  - Obstructions of line of sight make it difficult to reconstruct 3-D keys
  - Text embedded in objects makes cut and paste difficult
- Recognizing fake image
  - How close is close enough? Small changes (1 char)?

Pluggable Problems

- Hard AI problems are “pluggable” into applications
- 3-D KHAP (already discussed)
- Speech KHAP
  - message is speech-synthesized audio clip
  - key is voice parameters; user recognizes voice (parameters selected randomly at key generation)
  - audio distortion used to increase difficulty of analysis
- Handwriting KHAP
  - message is rendered using handwriting sample
  - key is writing style; user recognizes (too hard?)
  - visual distortion used to increase difficulty of analysis

Protocol: Human sends to TCB

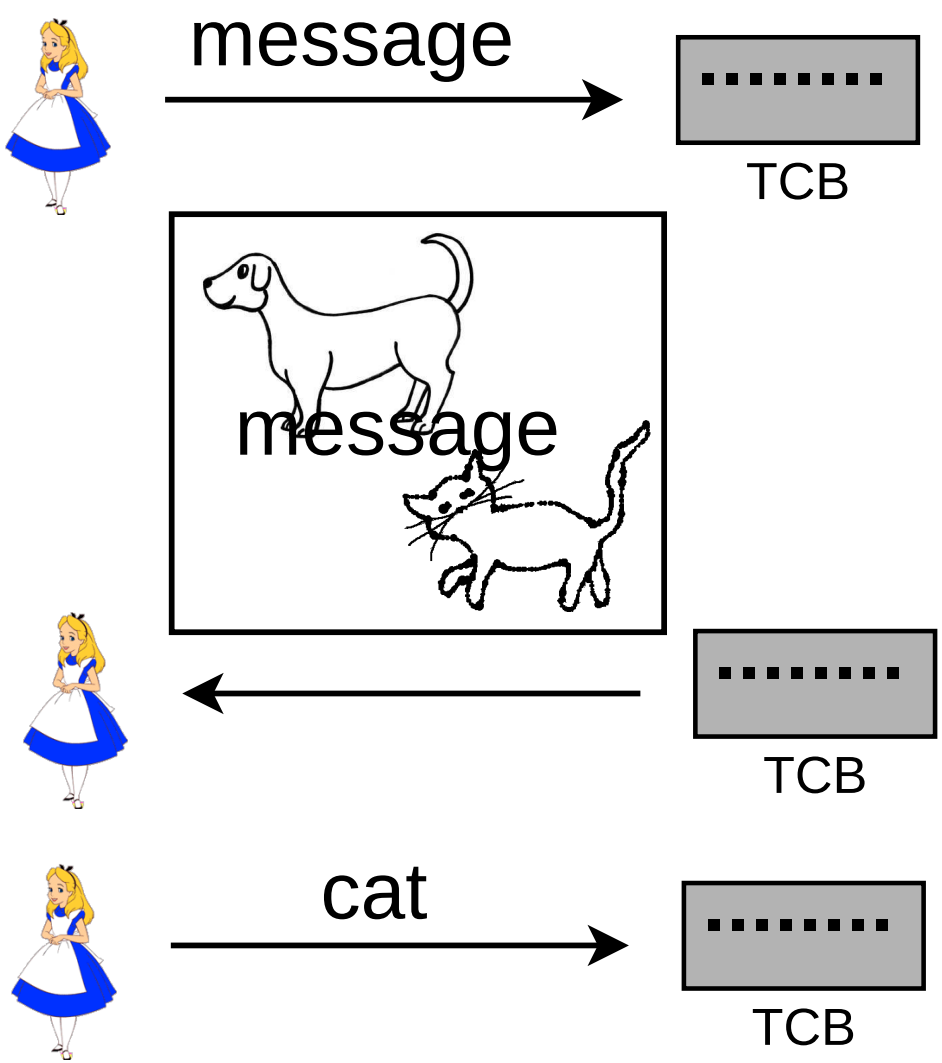


How can step 5 be performed?

Human to TCB Confirmation

- Only one bit is needed.
- Insertion or extraction of a portable device
  - Does not work for remote trusted server
  - Extraction: timing depends on application
  - Insertion to confirm is awkward
- Confirmation word/object
  - Pre-arranged (requires memory aid!) or type/click element from KHAP
  - Requires no special devices
  - Human adversary can be effective

Confirmation Word Example



Conclusions

- Approach is general (mobile device, network, etc)
- Many of the techniques map directly to what humans to do insure security in pencil and paper world
  - Nobody signs a document that has patches covering some words
  - People authenticate each other on the phone by voice characteristics
- Secure
  - Security depends on AI problem parameters
  - Advances in AI break problems (as factoring breaks RSA)
- Easy to use
  - Avoid computation, memory aids: ask humans to do what they do best
  - Some problems are intuitive (e.g., recognizing voice)

Future Work

- Develop specific KHAP problems
  - Evaluate usability (empirical studies)
  - Evaluate security (empirical + expert opinion)
- Collaboration with AI, graphics, speech, human-computer interactions, OCR
- Performance issues for low-power devices
- Key generation and re-keying
- Analyze human attacks (general and specific applications)

Questions?