Deniability

Dušan Klinec

OpenLab

6. 3. 2015

About me

- Whitebox cryptography, EACirc (ANF), Cube attack, NAT traversal, WSN (ProtectLayer).
- PhD student, with suspended study for 1 year.

Motivation

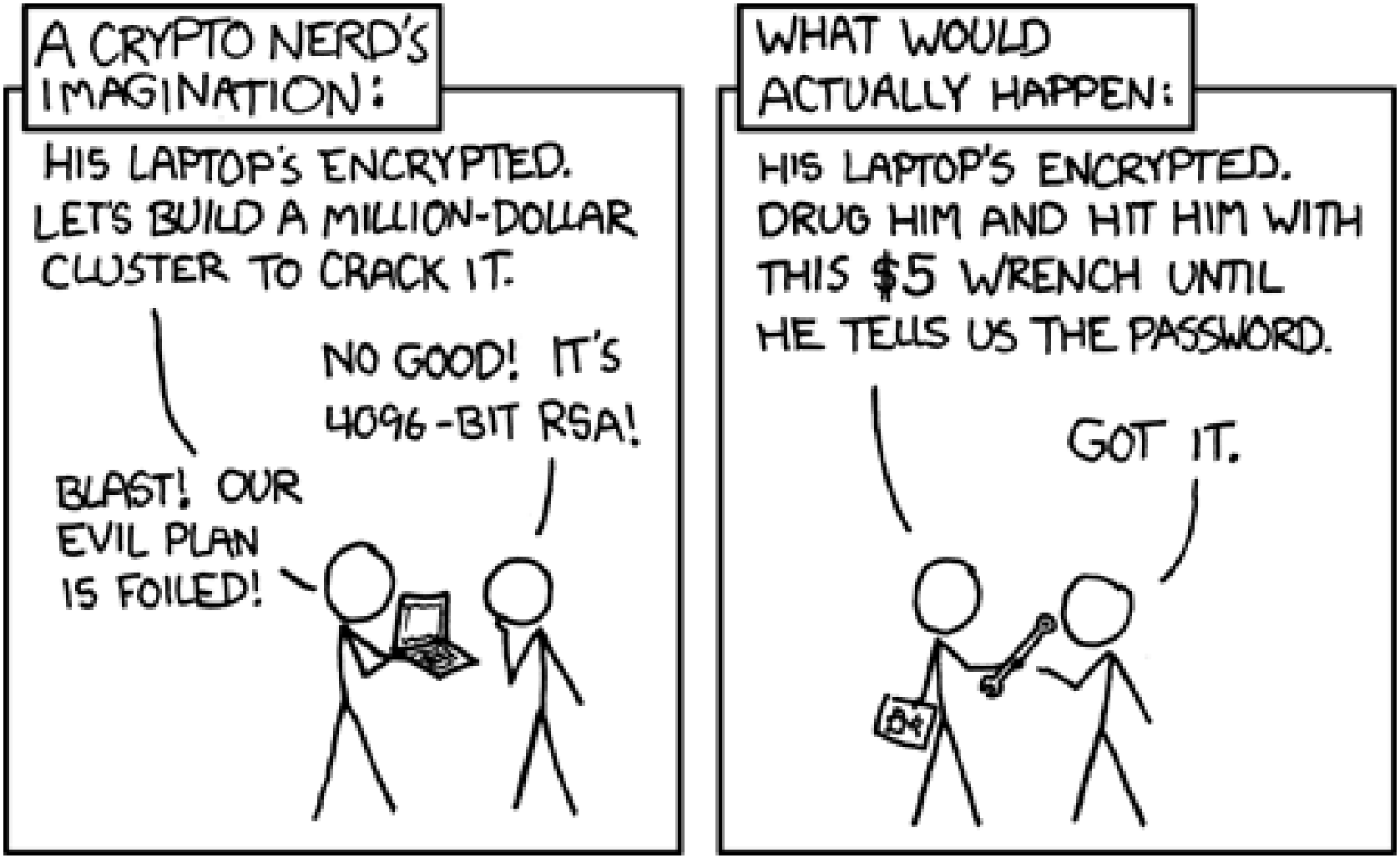
Deniability

Deniability

- Bob does not want Alice (or anybody else) to prove to a third party (e.g., judge) that:
  - Bob said "Assassinate him".
  - Bob ever talked to Alice (assassin).
- Use cases: lawyers, journalists, whistle blowers, disidents, ...
- From crypto view: conflicting requirements with authentication.
- Deniability comes in many different flavours.

Puzzle

Puzzle II



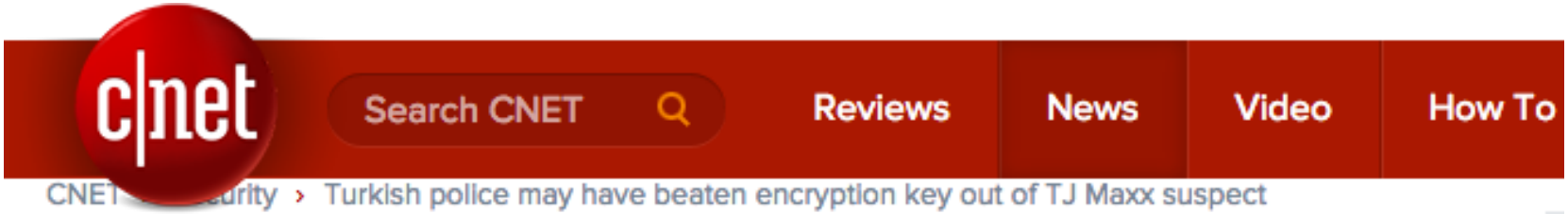
http://xkcd.com/538/

Puzzle solution

Rubber-hose cryptanalysis

- Euphemism for extraction of (crypto) secrets by coercion or torture.
- Not that good for disidents, NSA/CIA/FBI targets, ...

It may happen in real



CNET Security › Turkish police may have beaten encryption key out of TJ Maxx suspect

Turkish police may have beaten encryption key out of TJ Maxx suspect

When criminals turn to disk encryption to hide the evidence of their crimes, law enforcement investigations can hit a brick wall. Where digital forensics software has failed to recover encryption passwords, one tried and true technique remains: violence.

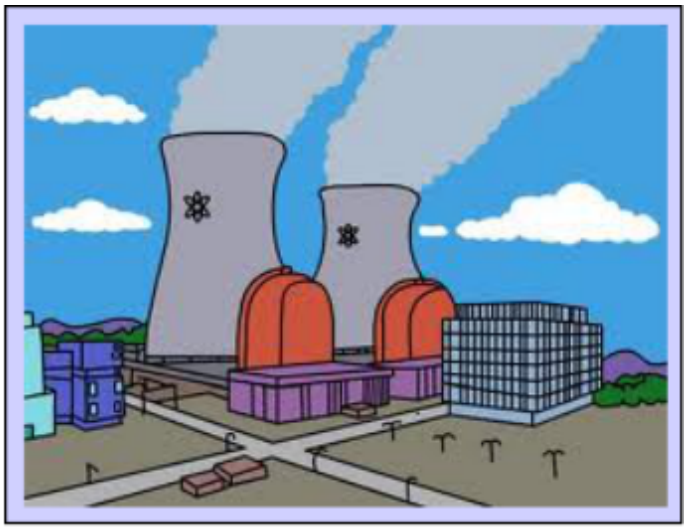
http://www.cnet.com/news/turkish-police-may-have-beaten-encryption-key-out-of-tj-maxx-suspect/

Neuroscience based approach

- Connects neuroscience and authentization.
- Authenticate via learning.
- User password cannot be revealed.

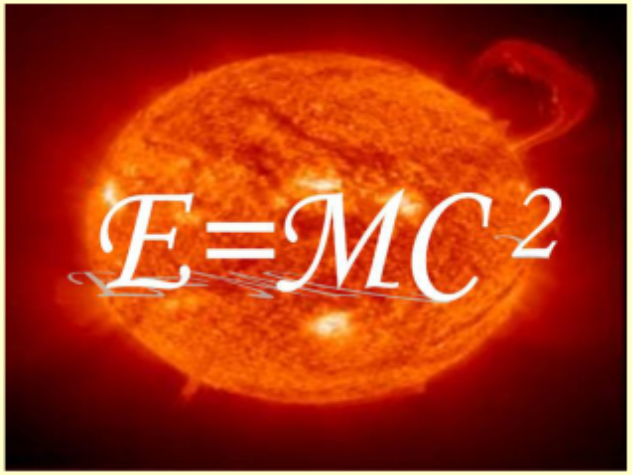
Setup / attacker model

Goal: Passwords that cannot be revealed consciously



https://www.usenix.org/sites/default/files/conference/protected-files/bojinov_usenixsecurity12_slides.pdf

Human memory systems

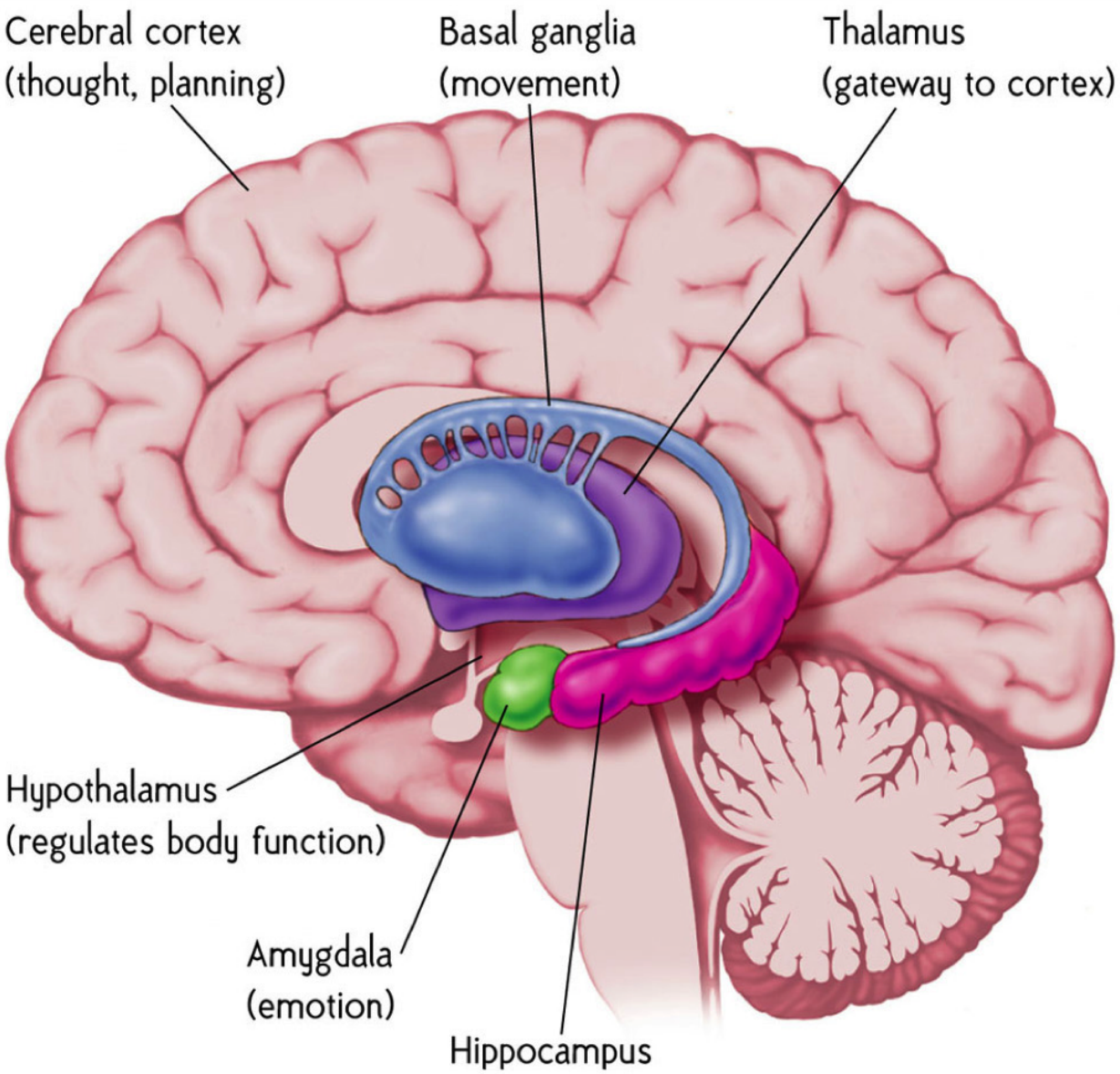


Declarative

Procedural

https://www.usenix.org/sites/default/files/conference/protected-files/bojinov_usenixsecurity12_slides.pdf

Procedural memory system



Use procedural memory

http://media.gamerevolution.com/images/misc/guitar%3Dhero.jpg

Why?

Why?

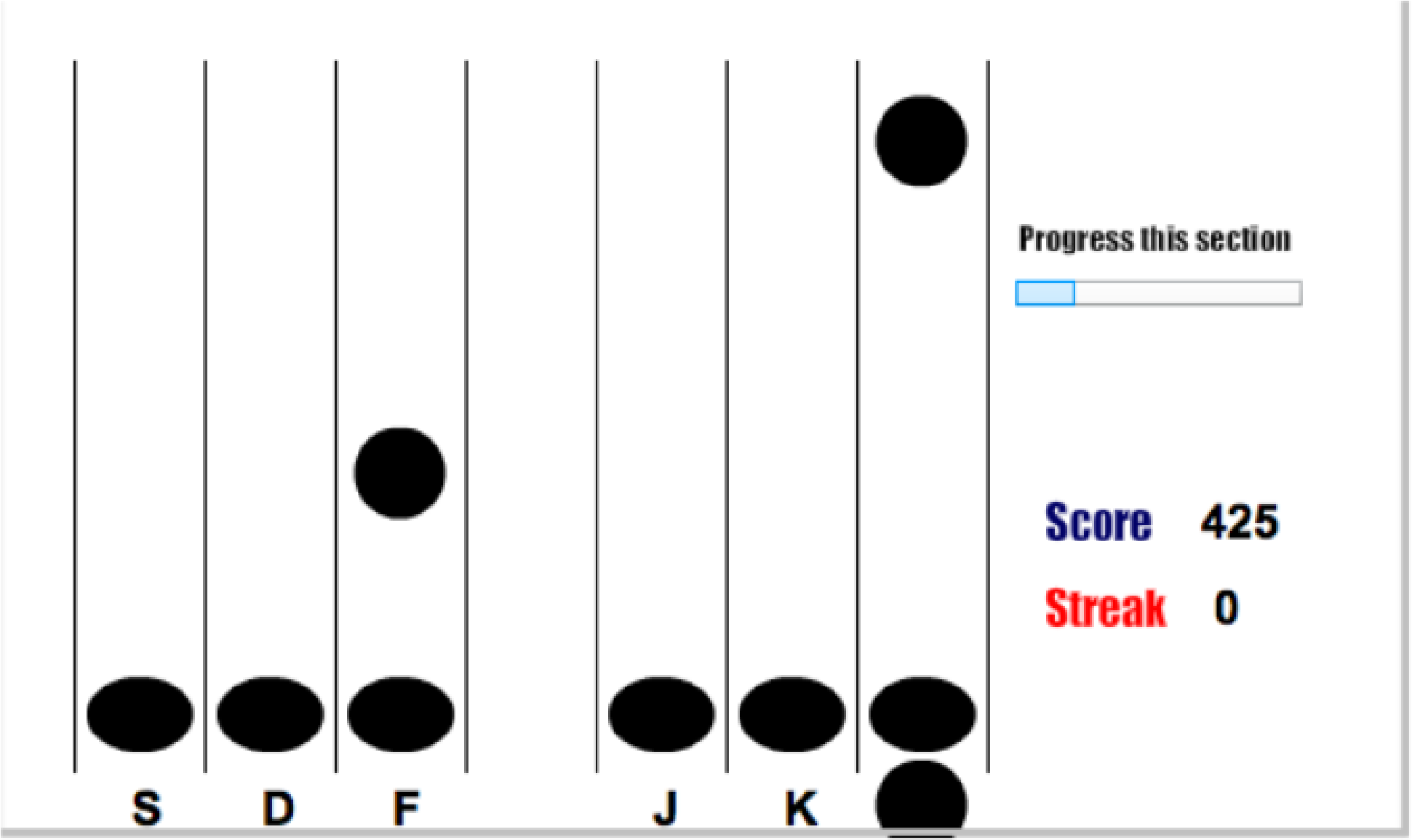
- Ability to learn rapidly long sequences (80 items).
- Not affected by noise.
- Sequences not consciously remembered.
- Goal: covertly embed non-reportable crypto data in human brain.

Why?

Why?

- Ability to learn rapidly long sequences (80 items).
- Not affected by noise.
- Sequences not consciously remembered.
- Goal: covertly embed non-reportable crypto data in human brain.

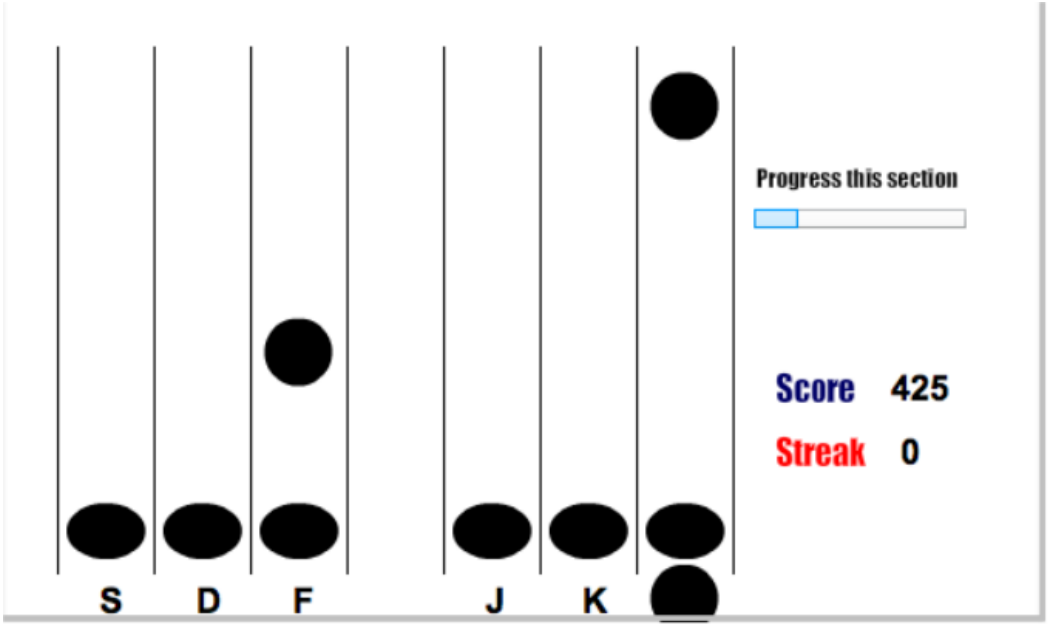
Authentication game



https://www.usenix.org/sites/default/files/conference/protected-files/bojinov_usenixsecurity12_slides.pdf

Authentication game

- Game: Serial Interception Sequence Learning
- Develop sensitivity to structured information w/o being aware of it.
- Fast game tempo, keep user at his/her limits (70% success).
- Falling balls = sequence of $\{s, d, f, j, k, l\}$
- Learned sequence entropy: ~ 37.8 bits.



How does the learning works?

1. Repeat 3 times secret sequence of 30 items + 18 randomly selected items (noise). Total 108 items.
2. Repeat 5 times, gives 540 items.
3. Pause.
4. Repeat 540 item sequence 6 times.

Results:

- Time needed for training (password set): 30–45 minutes.
- Authentication time: 5–6 minutes.

How does the learning works?

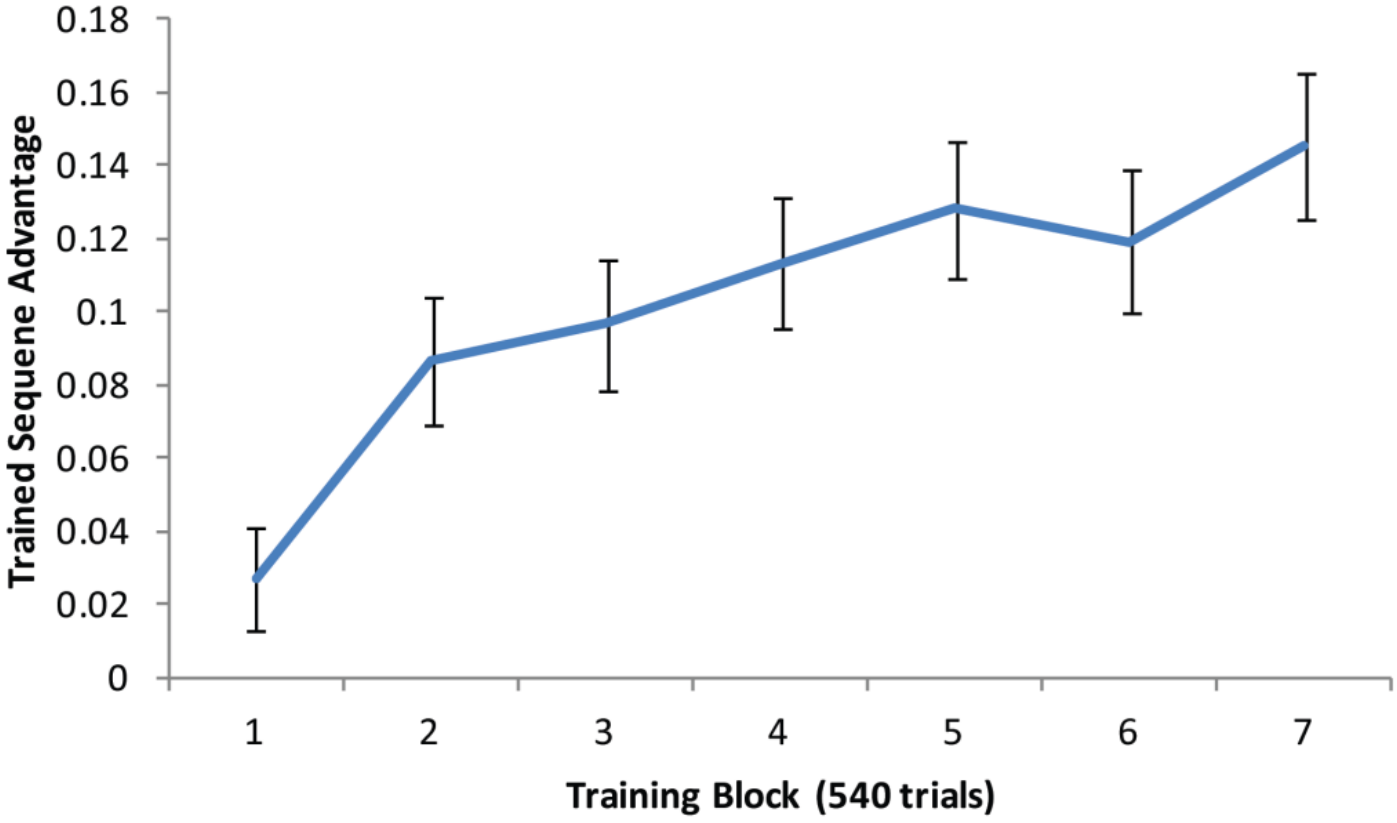
1. Repeat 3 times secret sequence of 30 items + 18 randomly selected items (noise). Total 108 items.
2. Repeat 5 times, gives 540 items.
3. Pause.
4. Repeat 540 item sequence 6 times.

Results:

- Time needed for training (password set): 30–45 minutes.
- Authentication time: 5–6 minutes.

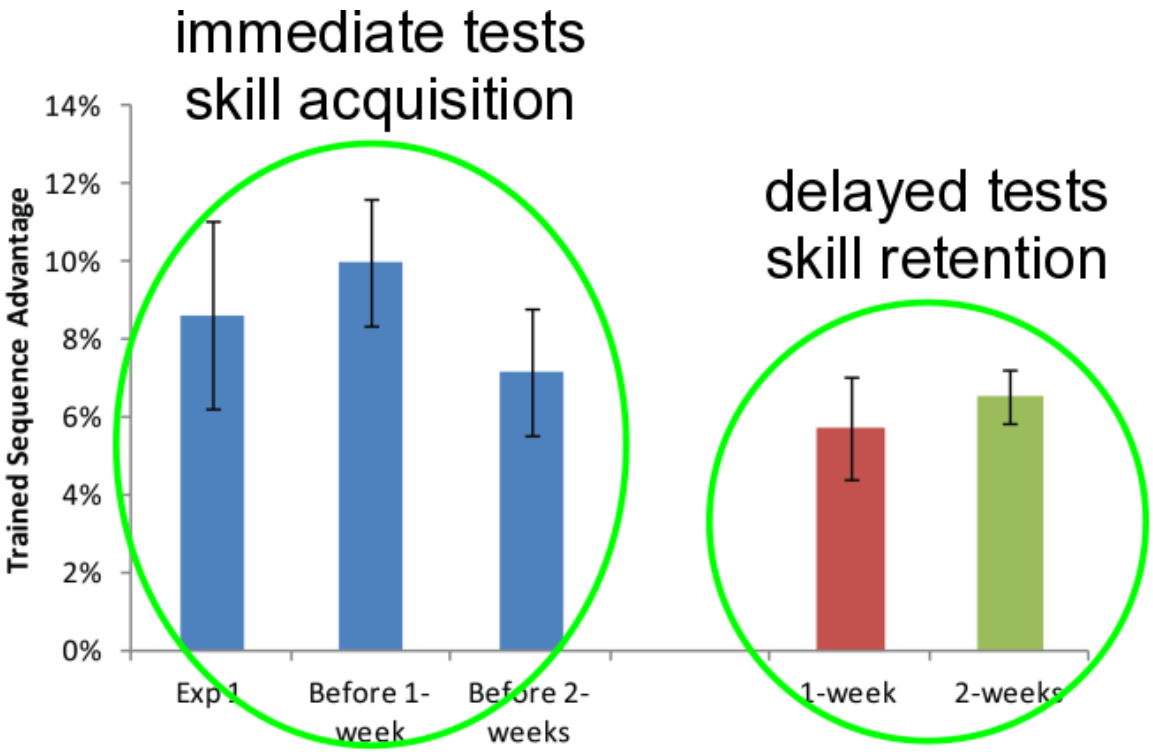
How does authentication works?

- Measure success rate of trained vs. untrained sequences.



https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final25.pdf

Forgetting



https://www.usenix.org/sites/default/files/conference/protected-files/bojinov_usenixsecurity12_slides.pdf

Back to practice...

Deniability

Plausible deniability

- We can claim we did not produce the message.
- In presumption of innocence establishment, someone has to prove the connection.
- Informally: PD protects us of being accused without proof.
- Good for lawyers, courts, ...

In fact, to the best of my knowledge no court in the history of law has ever used a cryptographic transcript as evidence that a conversation occurred.....However it makes the problem a bit more sexy - Matthew Green, http://blog.cryptographyengineering.com/2014/07/noodling-about-im-protocols.html

Deniability

Plausible deniability

- We can claim we did not produce the message.
- In presumption of innocence establishment, someone has to prove the connection.
- Informally: PD protects us of being accused without proof.
- Good for lawyers, courts, ...

In fact, to the best of my knowledge no court in the history of law has ever used a cryptographic transcript as evidence that a conversation occurred.....However it makes the problem a bit more sexy - Matthew Green, http://blog.cryptographyengineering.com/2014/07/noodling-about-im-protocols.html

Full deniability

- Plausible deniability does not help against rubber-hose attack.
- Full deniability is hard to achieve.

Decrypt-to-anything approach

- In time of coercion, open ciphertext to arbitrary plaintext.
- Very strong model, not very practical though.
- Based on lattice problems, keys are bigger than messages.
- Intuition: Need to have 10 keys so we can open it in 10 different ways.

Plan-ahead approach

- Prepare *cover* messages in advance.
- A and B negotiate on innocent messages somehow.
- Limited number of cover messages.
- In time of coercion, reveal innocent cover msg.
- Resembles *Chaffing and winnowing*

Full deniability

- Plausible deniability does not help against rubber-hose attack.
- Full deniability is hard to achieve.

Decrypt-to-anything approach

- In time of coercion, open ciphertext to arbitrary plaintext.
- Very strong model, not very practical though.
- Based on lattice problems, keys are bigger than messages.
- Intuition: Need to have 10 keys so we can open it in 10 different ways.

Plan-ahead approach

- Prepare *cover* messages in advance.
- A and B negotiate on innocent messages somehow.
- Limited number of cover messages.
- In time of coercion, reveal innocent cover msg.
- Resembles *Chaffing and winnowing*

Full deniability

- Plausible deniability does not help against rubber-hose attack.
- Full deniability is hard to achieve.

Decrypt-to-anything approach

- In time of coercion, open ciphertext to arbitrary plaintext.
- Very strong model, not very practical though.
- Based on lattice problems, keys are bigger than messages.
- Intuition: Need to have 10 keys so we can open it in 10 different ways.

Plan-ahead approach

- Prepare *cover* messages in advance.
- A and B negotiate on innocent messages somehow.
- Limited number of cover messages.
- In time of coercion, reveal innocent cover msg.
- Resembles *Chaffing and winnowing*

Plausible deniability

- Plausible deniability is usable in practice.
- Many protocols that guarantee PD.
  - Deniable authentication.
  - Deniable key exchange.
  - Deniable public key encryption.
- OTR, TextSecure chat protocols claim PD.
- HMQV - implicitly authenticated key-exchange protocol claim PD.
- SIGMA - authenticated key-exchange protocol.
  - SIGn-and-MAC.
  - Used in IKE.
  - OTR based on SIGMA.
  - Deniability - hadred with signatures.

Plausible deniability

- Plausible deniability is usable in practice.
- Many protocols that guarantee PD.
  - Deniable authentication.
  - Deniable key exchange.
  - Deniable public key encryption.
- OTR, TextSecure chat protocols claim PD.
- HMQV - implicitly authenticated key-exchange protocol claim PD.
- SIGMA - authenticated key-exchange protocol.
  - SIGn-and-MAC.
  - Used in IKE.
  - OTR based on SIGMA.
  - Deniability - hadred with signatures.

Good-to-know in the deniability field

First formal deniable auth treatment: Started by Dwork, Naor, Sahai. *Concurrent Zero-Knowledge.*

Hugo Krawczyk

- Canetti-Krawczyk formal model. Prove security properties of KE protocols. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels (2001).
- SKEME: a versatile secure key exchange mechanism for Internet (1996).
- SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols (2003).
- HMQV: A High-Performance Secure Diffie-Hellman Protocol (2005).
- Cryptographic Extraction and Key Derivation: The HKDF Scheme (2011).

Good-to-know in the deniability field

First formal deniable auth treatment: Started by Dwork, Naor, Sahai. *Concurrent Zero-Knowledge.*

Hugo Krawczyk

- Canetti-Krawczyk formal model. Prove security properties of KE protocols. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels (2001).
- SKEME: a versatile secure key exchange mechanism for Internet (1996).
- SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols (2003).
- HMQV: A High-Performance Secure Diffie-Hellman Protocol (2005).
- Cryptographic Extraction and Key Derivation: The HKDF Scheme (2011).

Motivation

Plausible deniability in practice

Still motivating

Digital signatures

- Non-repudiation property.
- Good for legal contracts, bad for instant messaging.
- Scenario: Alice is bribed/forced to testify against Bob.
- (Or she is just mad angry Ex).
- Alice can prove Bob's authorship to the third party (e.g., Judge).

Bad scenarios for Bob

Bad scenarios for Bob:

- Pretty Good Privacy - signed messages. Easy to prove authorship.
  - Web of trust.
  - Key used for a while.
- IM message signed with RSA/DSA/...
  - Bob's certificate issued by trusted CA.
  - Another obvious trusted linking of Bob to used key-pair.

HMAC

HMAC

- ~~Non-repudiation~~, Alice and Bob can both create & verify.
- Authenticity of messages is guaranteed.
- Bob cannot prove authorship to a thirdparty.

Deniability

- Definition based on an existence of a simulator $\mathcal{S}$.
- There exists an effective (polynomial) algorithm (simulator) $\mathcal{S}$ that produces transcripts indistinguishable from the real ones.
- Simulator does not know private keys of parties.
- If $\mathcal{S}$ exists, any conversation can be *a posteriori* claimed to be produced by the simulator.

Deniability

Strong Deniability

- Claim that anybody could have modified the message.

Weak Deniability

- Only A or B could have created a valid message.
- A and B possess the full set of necessary key material derived from this shared secret for any message at the time it is sent.
- In practice: MAC-ed messages with common key from authenticated key exchange.

Deniability

Strong Deniability

- Claim that anybody could have modified the message.

Weak Deniability

- Only A or B could have created a valid message.
- A and B possess the full set of necessary key material derived from this shared secret for any message at the time it is sent.
- In practice: MAC-ed messages with common key from authenticated key exchange.

Deniability

```
+-------+ "I pwn3d XYZ", sig(..)    +-------+
| Alice | <-----------------------> |  Bob  |
+-------+                           +-------+
    ^
    | "Bob pwn3d XYZ, here's a proof"   +-------+
    +-------------------------------> | Judge |
                                        +-------+
```

- Simple offline attack.
- Bob: Alice made it up in order to compromise me.

Deniability

```
+-------+ "I pwn3d XYZ", sig(..)    +-------+
| Alice | <-----------------------> |  Bob  |
+-------+              ^            +-------+
   ^                   |
   |                   +---- Sniff -------------+
   |                                            |
   | "See? Bob really said he pwn3d XYZ." +-------+
   +------------------------------------> | Judge |
                                          +-------+
```

- Idea: Alice gives all Enc and Mac keys to Judge.
- Hard to implement (TOR used, anonymizers). Is the channel real?
- Even if OK, Judge cannot prove it further (e.g., Jury). Not that big deal, police could trust plaintext log...

Deniability

```
+-------+          +-------+
| Alice |<-+    +->|  Bob  |
+-------+  |    |  +-------+
           |    |
  non-OTR  |    |  (OTR) Bob-Judge[Alice]
           v    v
         +-------+
         | Judge |
         +-------+
```

- Alice gives private key for Judge to act behalf of Alice.
- Alice does not really trust to Judge (impersonation,...)

Deniability

```
+-------+          +-------+
| Alice |<-+    +->|  Bob  |
+-------+  |    |  +-------+
           |    |
           v    v
     +----------------+       +-------+
     | Alice's Lawyer | <---> | Judge |
     +----------------+       +-------+
```

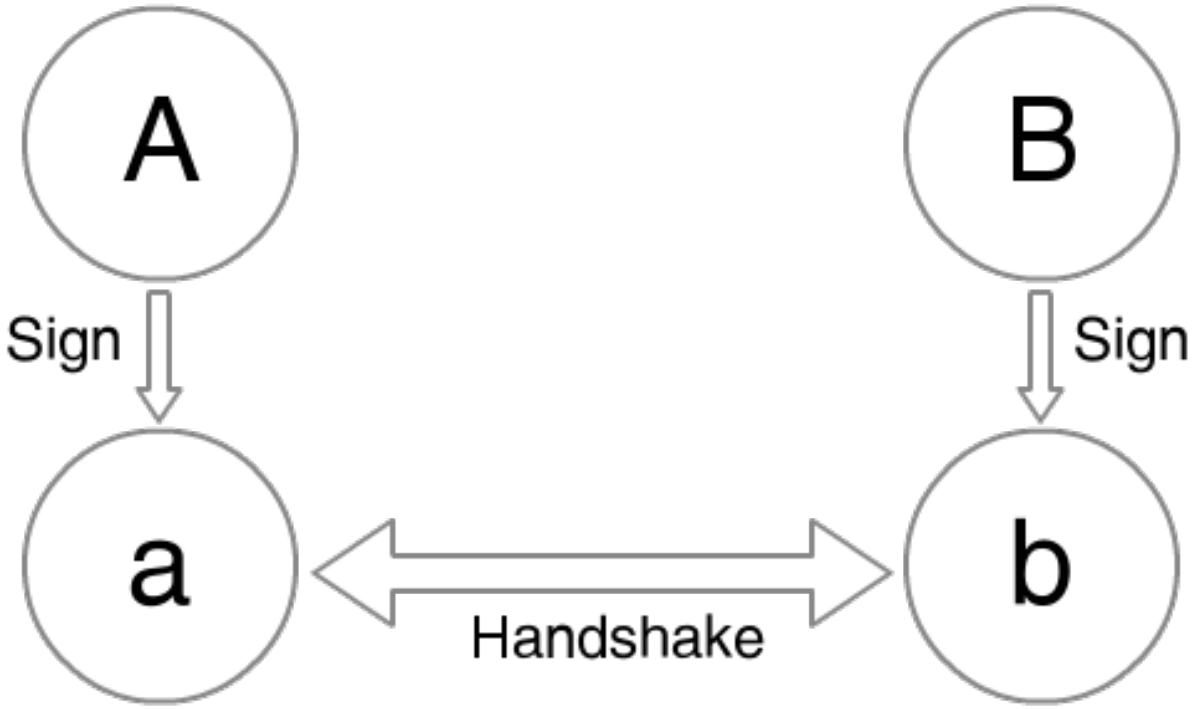
- Not any better. Does Alice really trust to the lawyer?
- Does Judge trust to the Alice's lawyer?
- $\rightarrow$ No trusted third party!

OTR

Off-the-Record protocol

- Confidentiality
- Integrity
- Forward Secrecy
- Mutual authentication using Socialist Millionaire Protocol
- Deniability

OTR AKE - simplified key-exchange



https://whispersystems.org/blog/simplifying-otr-deniability/

OTR AKE - in detail

BOB ALICE

$\mathrm{AES}_r(g^x), \mathrm{HASH}(g^x)$ (1)

→ "D-H Commit Message"

g^y (2)

← "D-H Key Message"

$M_B = \mathrm{MAC}_{K_{m_1}}(g^x, g^y, pub_B, keyid_B)$
$X_B = \{pub_B, sig_B(M_B)\}$

$r, \mathrm{AES}_c(X_B), \mathrm{MAC}_{K_{m_2}}(\mathrm{AES}_c(X_B))$

→ "Reveal Signature Message"

$M_A = \mathrm{MAC}_{K_{m'_1}}(g^y, g^x, pub_A, keyid_A)$ (3)
$X_A = \{pub_{A'}, keyid_{A'}, sig_A(M_A)\}$

$\mathrm{AES}_{c'}(X_A), \mathrm{MAC}_{K_{m'_2}}(\mathrm{AES}_{c'}(X_A))$

← "Signature Message"

1. Hash commitment
2. Diffie-Hellman Key Exchange
3. Encrypted exchange of long-term keys & signatures

http://blog.cryptographyengineering.com/2014/07/noodling-about-im-protocols.html

OTR message - in detail

ALICE BOB

$$M_x = \{\text{keyid}_A, \text{keyid}_B, g^{x'}, t, \text{AES-CTR}(msg)\}$$

$$K_j^* = \{K_{m_{j-1}}, K_{m'_{j-1}}, K_{m_j}, K_{m'_j}\}|\emptyset$$

$$\{M_{x_i}, \text{MAC}_{K_{x_i}}(M_{x_i}), K_j^*\} \longrightarrow$$

"Data Exchange Message"

$$\longleftarrow \{M_{y_i}, \text{MAC}_{K_{y_i}}(M_{y_i}), K_{j+1}^*\}$$

"Data Exchange Message"

$$\{M_{x_{i+1}}, \text{MAC}_{K_{x_{i+1}}}(M_{x_{i+1}}), K_{j+2}^*\} \longrightarrow$$

"Data Exchange Message"

1. New Diffie-Hellman share
2. Malleable ciphertext
3. MAC on ciphertext
4. Revealed MAC key (for last message)

http://blog.cryptographyengineering.com/2014/07/noodling-about-im-protocols.html

OTR

Weak deniability

- Offers weak deniability - HMAC keys generated from shared secret.
- Bob responds to any challenge, signs key, not mentioning Alice in signature.

Towards strong deniability

- Strong deniability = anyone could forge transcripts. Not that simple.
- In between = with network traffic of conversation can forge.

OTR

Weak deniability

- Offers weak deniability - HMAC keys generated from shared secret.
- Bob responds to any challenge, signs key, not mentioning Alice in signature.

Towards strong deniability

- Strong deniability = anyone could forge transcripts. Not that simple.
- In between = with network traffic of conversation can forge.

OTR

- New DH key-exchange with every message.
- Old HMAC keys published in media channel in plaintext.
- Uses malleable encryption scheme (AES-CTR).
- Malleable $\rightarrow$ attacker can create a fake ciphertexts from real ones.
- Aim: attacker can create a valid HMAC and forge transcripts.

Criticism

- Does not really work in practice. Unrealistic.
- Very weak guarantees, simulator approach does not work.

OTR

- New DH key-exchange with every message.
- Old HMAC keys published in media channel in plaintext.
- Uses malleable encryption scheme (AES-CTR).
- Malleable $\rightarrow$ attacker can create a fake ciphertexts from real ones.
- Aim: attacker can create a valid HMAC and forge transcripts.

Criticism

- Does not really work in practice. Unrealistic.
- Very weak guarantees, simulator approach does not work.

Nice attack on deniability

- No trusted third party problem.
- Use of secure function evaluation - Garbled circuits.
- $f(x, y, z)$, x Alice's private key, y msg to send, z received msg.
- Output is an OTR message to send to Bob.

```
+-------+            +-------+          +-------+
| Alice |<-+      +->| Judge |          |  Bob  |
+-------+  |      |  +-------+          +-------+
           |      |                        |
 priv. key |      |  message               | Bob's priv.
           v      v                        v
     +-----------------+               +---------+
     | Garbled Circuit | <-- OTR --->  | IM app  |
     +-----------------+               +---------+
```

Nice attack on deniability

- No trusted third party problem.
- Use of secure function evaluation - Garbled circuits.
- $f(x, y, z)$, x Alice's private key, y msg to send, z received msg.
- Output is an OTR message to send to Bob.

```
+-------+            +-------+          +-------+
| Alice |<-+      +->| Judge |          |  Bob  |
+-------+  |      |  +-------+          +-------+
           |      |                         |
 priv. key |      |  message                | Bob's priv.
           v      v                         v
     +-----------------+                +--------+
     | Garbled Circuit | <-- OTR --->   | IM app |
     +-----------------+                +--------+
```

OTR resources

- OTR v3 4.0 https://otr.cypherpunks.ca/Protocol-v3-4.0.0.html
- Mathew Green - Noodling about IM protocols

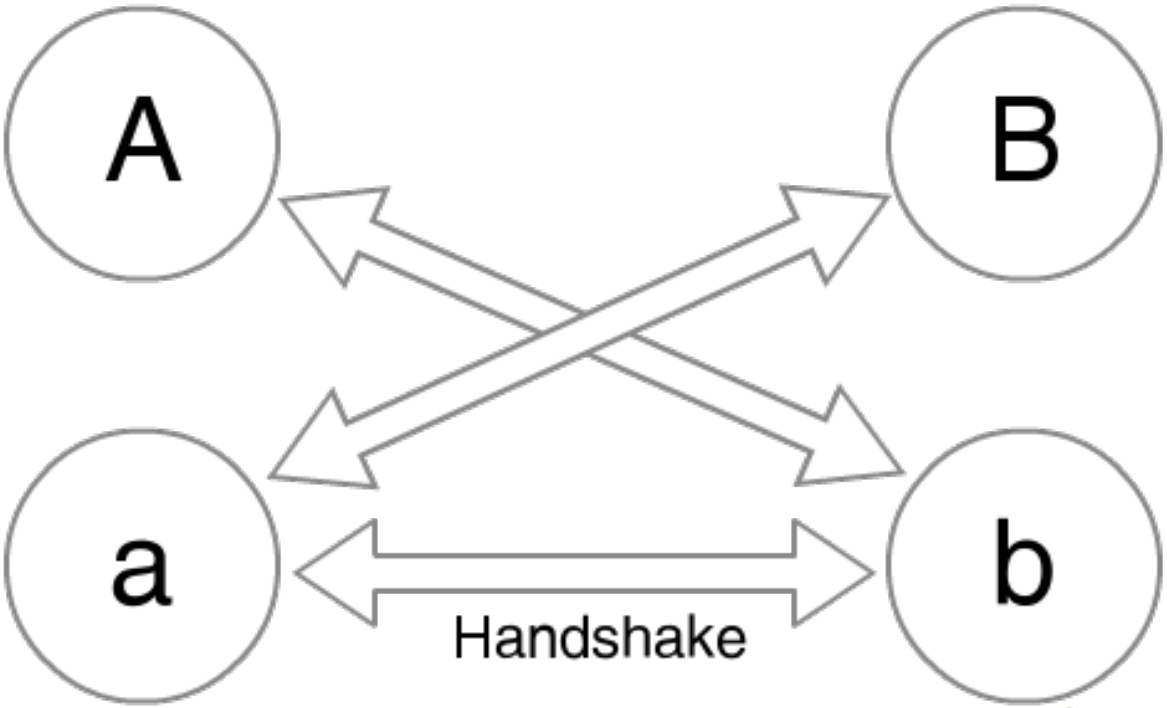
  http://blog.cryptographyengineering.com/2014/07/noodling-about-im-protocols.html
- Bruhns, Greg - Secure function evaluation vs. deniability

  http://phrack.org/issues/68/14.html
- Riamondo, Gennaro, Krawczyk - Secure Off-the-Record Messaging

  https://www.dmi.unict.it/diraimondo/web/wp-content/uploads/papers/otr.pdf

SecureText

- If deniability is required, avoid signatures.
- Another approach - TextSecure v2 protocol.
- Does not use signatures at all.



https://whispersystems.org/blog/simplifying-otr-deniability/

TextSecure v2 KDF

- Using triple Diffie-Hellman key exchange.
- Users A, B have long-term DH keys, $(a, g^a), (b, g^b) \in \mathbb{Z}_p \times$ Curve25519 resp.

KDF

- A generates: $(x_{a,0}, g^{x_{a,0}}) \in_R \mathbb{Z}_p \times$ Curve25519.
- $K = f(g^{x_{b,z} \cdot a} || g^{x_{a,0} \cdot b} || g^{x_{b,z} \cdot x_{a,0}})$
- The only step when long-term keys are employed.
- Increases forgability - anyone can fetch pubkey and generate K.