Secure Verification of Delegated Presentation Delivery

Jon Callas, Tamzen Cannoy and Nicko van Someren
Presented by Brian LaMacchia

It’s a Tradition!

- Alice, Bob, Charles want to do a funny rump session



There’s a Problem

- They can’t make it to Crypto, so they can’t deliver it themselves.

There’s a Problem

- They can’t make it to Crypto, so they can’t deliver it themselves.

- Only option: outsourcing to an untrustworthy third party

We want “verified delegation”

- Verify that the session was delivered in a humorous **and** tamper-evident fashion
  - Ensure that **all** jokes were delivered as required
  - Ensure that the audience laughed in **all** the right places

Distributed problem solving

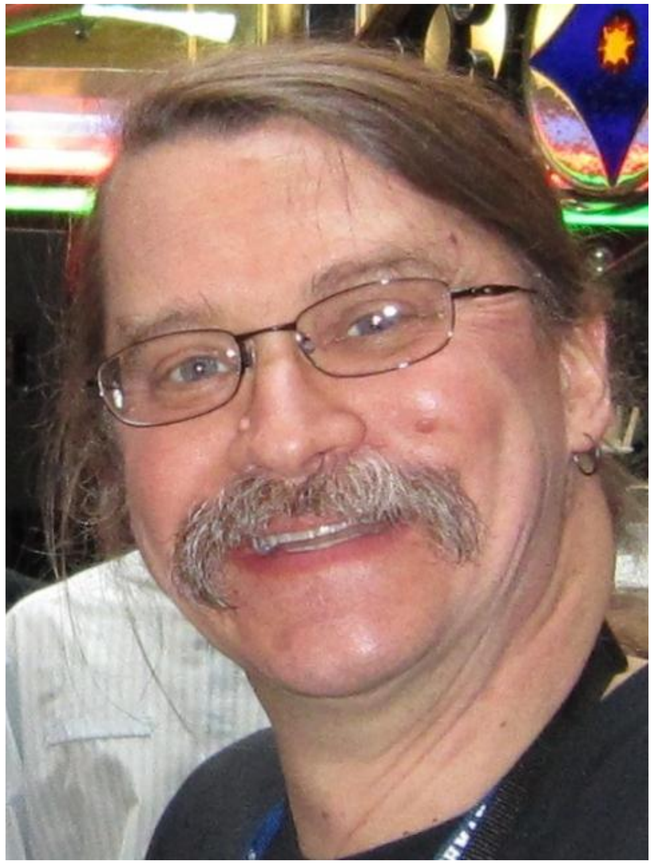
- Everyone has a suggestion for a solution strategy!

Solution 1

- Charles suggests a hardware security module

Solution 2

- Bob suggests a web of trust



Solution 3

- “Go ask,” Alice says, “someone who was there”

Hybrid Solution?

- As a compromise, they try asking a web of trusted security modules.

Hybrid Failure

- This fails, since secure devices have no sense of humor

Apply cryptography to the problem!

- Bi-deniable homomorphic encryption

Apply cryptography to the problem!

- Bi-deniable homomorphic encryption
  - Either party can prove the other didn’t get the joke

Apply cryptography to the problem!

- Bi-deniable homomorphic encryption
  - Either party can prove the other didn't get the joke
- Set up a secure computation to distribute the work to determine if it's funny from several participants

Cryptographic Problems

- Not secure against collusion or everyone denying everything

Cryptographic Problems

- Not secure against collusion or everyone denying everything
- We don’t have a finite formula for funny

Time lock puzzle as solution

Time lock puzzle as solution

- The humorous content of this talk will be delivered at a later date...