Attestation and Authentication Protocols Using the TPM

Ariel Segall

June 21, 2011

Motivation

- Almost all TPM interactions can be thought of as *cryptographic protocols*.
- Many real-world scenarios need attestation and machine authentication protocols.
- The TPM gives us the necessary building blocks to create protocols that accomplish complex, real-world goals.
  - Far more than just network access control!

"Us" includes you!

What is a Cryptographic Protocol?

For this talk:

A sequence of messages between participants

- All cryptography abstract
- Local assumptions about secret & fresh values
- Goal: each participant can prove their security goals met
  - Exact goals vary based on protocol
- Looking to prevent *structural flaws*

(For the interested: Strand Space method, Dolev-Yao adversary.)

What is a Cryptographic Protocol?

For this talk:

A sequence of messages between participants

- All cryptography abstract
- Local assumptions about secret & fresh values
- Goal: each participant can prove their security goals met
  - Exact goals vary based on protocol
- Looking to prevent *structural flaws*

(For the interested: Strand Space method, Dolev-Yao adversary.)

Notation:

- $\{M\}_K$ indicates that M is encrypted with key K
- $\{|M|\}_K$ indicates that K has signed M

Attestation vs. Authentication

For this discussion:

Attestation:

- Providing evidence about a target to an appraiser
- Used for predicting future target behavior

Authentication:

- Identification of a machine or user
- Usually, another partner in a protocol

A Motivating Scenario

A financial server contains sensitive financial and personal data.

We wish to limit access to:

- authorized users
- on authorized machines
- running authorized software

A Motivating Scenario

A financial server contains sensitive financial and personal data.

We wish to limit access to:

- authorized users Authentication
- on authorized machines Authentication
- running authorized software Attestation

Basic Authentication Mechanisms

Signature

- *A*'s private key signed this message, so the message must have come from *A*

Decryption

- Only *A*'s private key could have decrypted this data, so if the data is used, *A* must have received it

...assuming key secrecy, *A* associated with key

Attestation

- Goal: Predict behavior of a particular target
- Anything that provides evidence about a machine can qualify
  - User attestation could exist, but outside today's talk
- To be trustworthy:
  - Evidence provider must be authenticated
  - Target must be verifiable from evidence
- Example: TPM Quote attests to values in PCRs

Not all attestation is signature based!

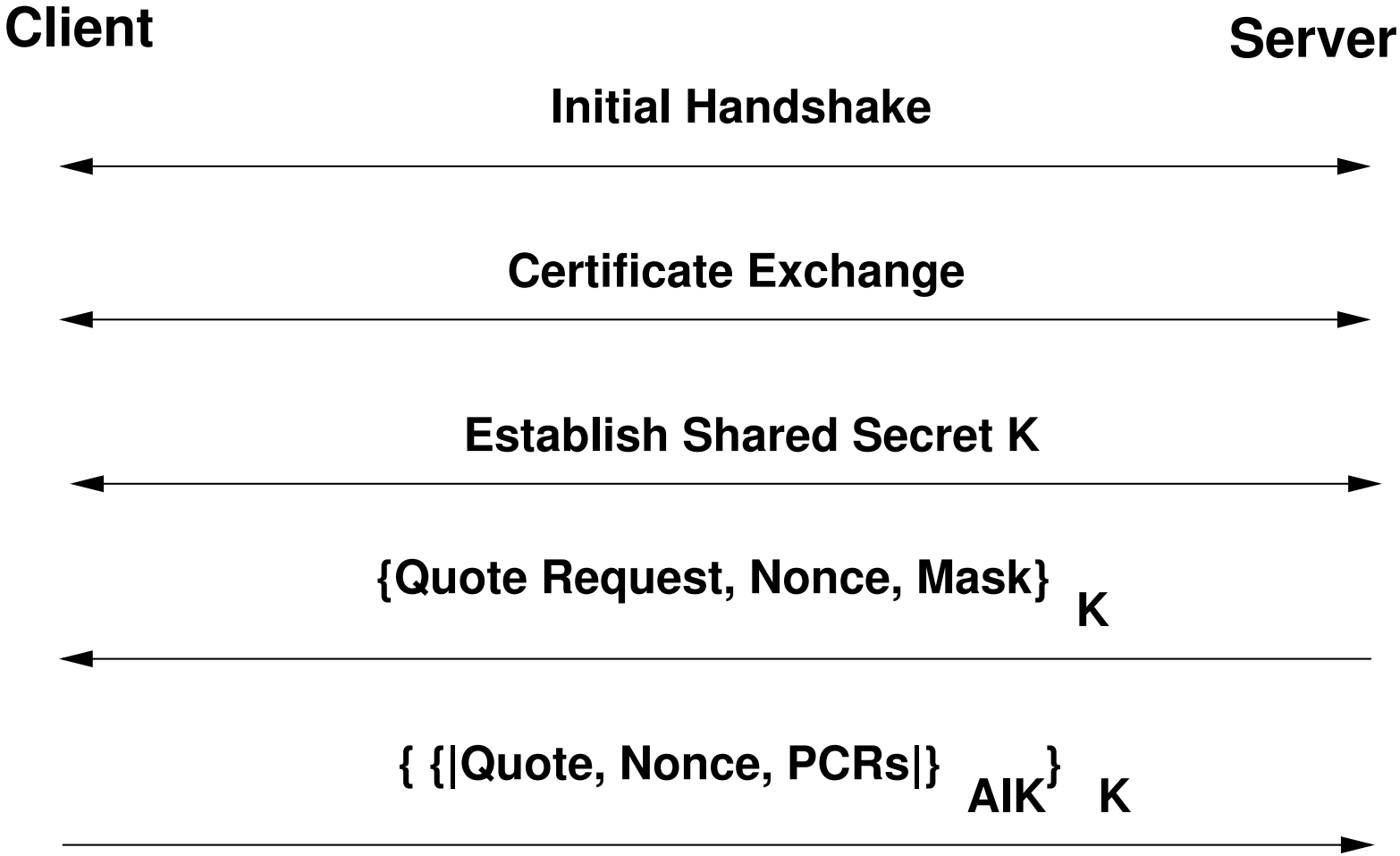
- CertifyKey allows decryption-based attestation as well
  - Certifies PCR values that must be present to use a key

A Straightforward Guess

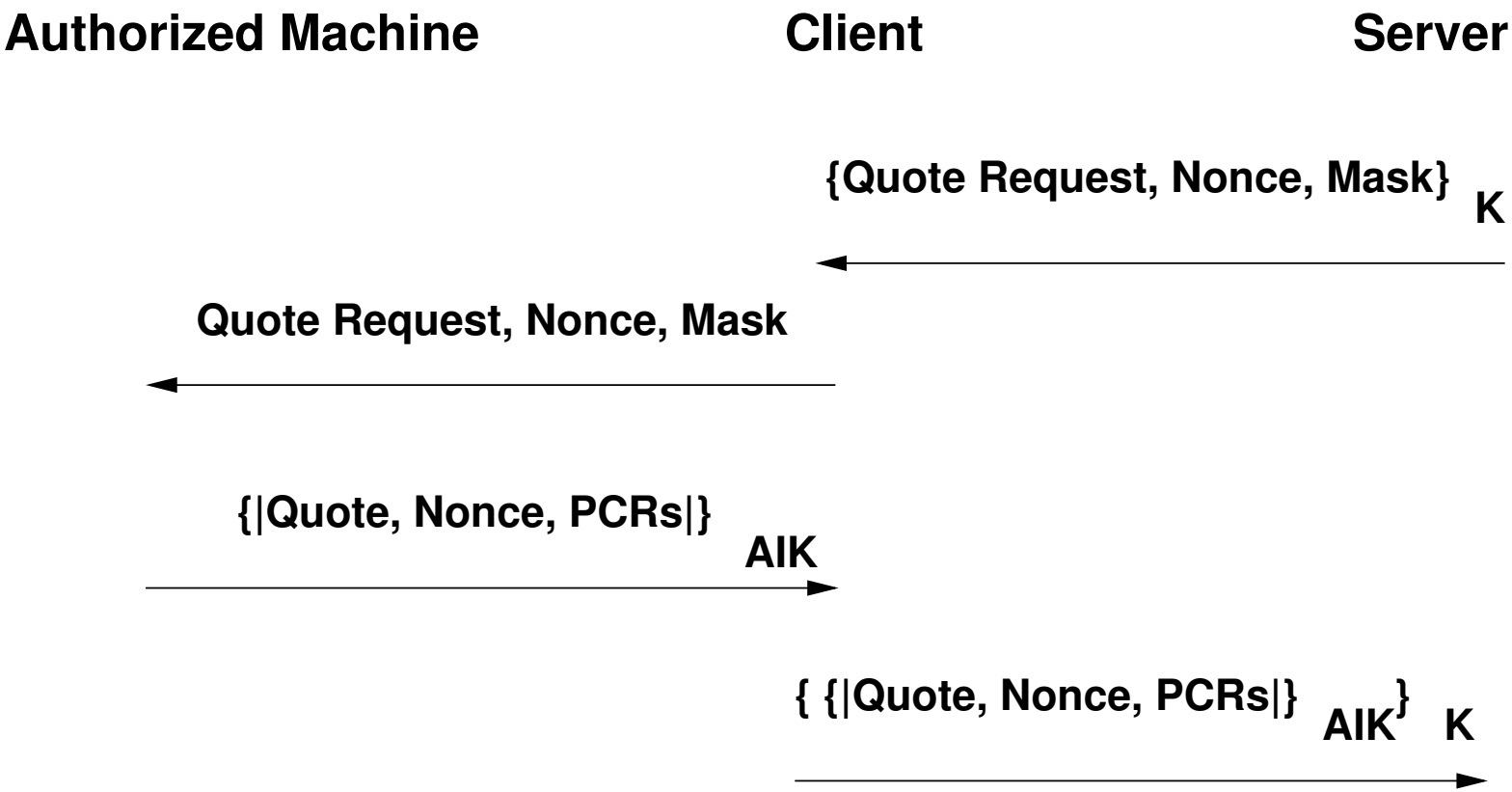
Let's say the server already uses TLS to authenticate the client and server, and would like to integrate the TPM for attestation and machine authentication.

A Straightforward Guess

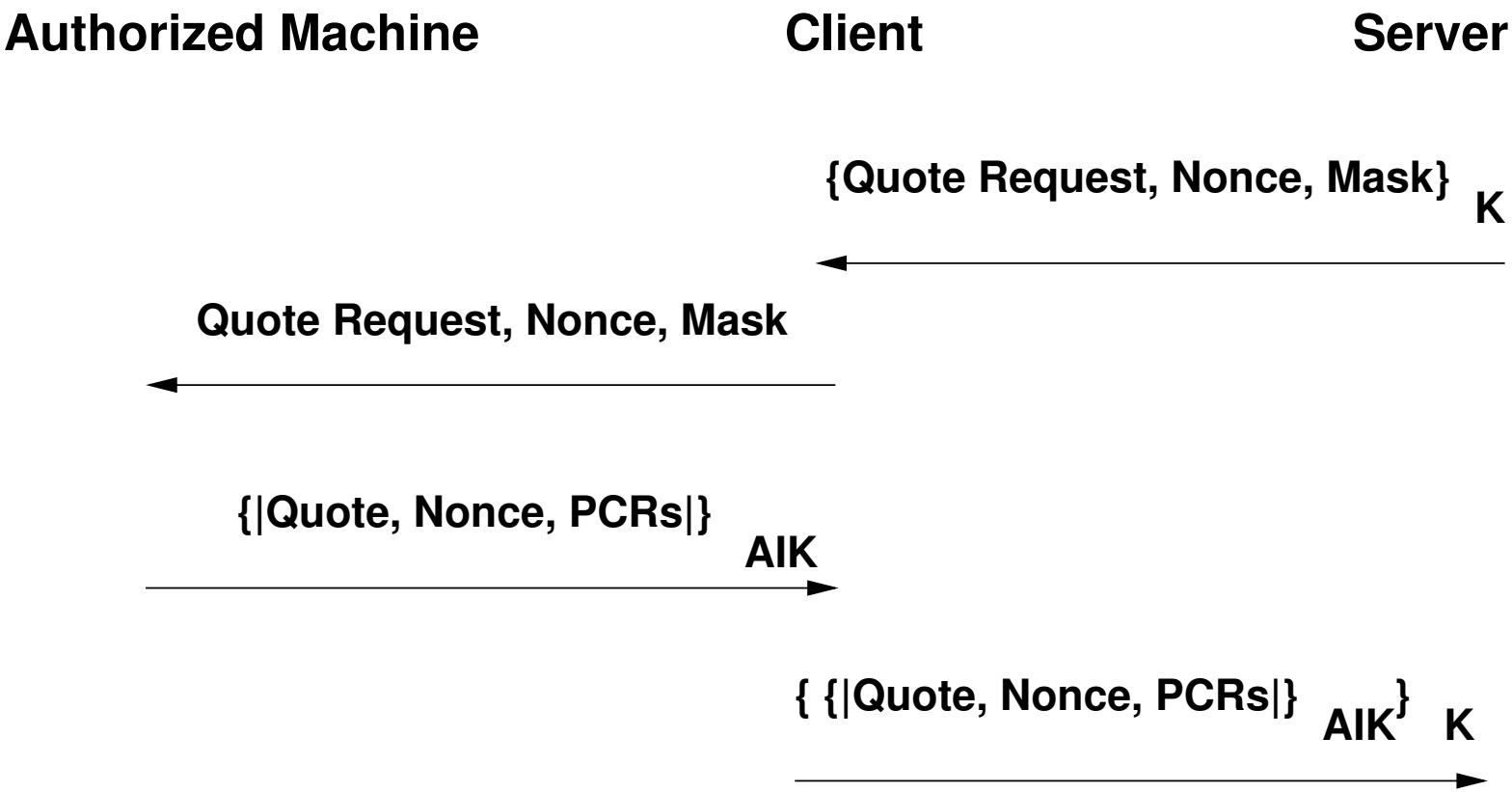
Let's say the server already uses TLS to authenticate the client and server, and would like to integrate the TPM for attestation and machine authentication.



The Attack



The Attack



By themselves, TPM quotes provide attestation, not meaningful machine authentication.

Basic Machine Authentication using a TPM

- Use a binding key to encrypt a session key
- Sign messages with an AIK, signing key
- Integrate protocol data into TPM Quote
  - Nonce (if you're careful)
  - Extended PCR (usually resettable)

Basic Machine Authentication using a TPM

- Use a binding key to encrypt a session key
- Sign messages with an AIK, signing key
- Integrate protocol data into TPM Quote
  - Nonce (if you're careful)
  - Extended PCR (usually resettable)

All of these assume that local software will only use TPM appropriately...

Basic Machine Authentication using a TPM

- Use a binding key to encrypt a session key
- Sign messages with an AIK, signing key
- Integrate protocol data into TPM Quote
  - Nonce (if you're careful)
  - Extended PCR (usually resettable)

All of these assume that local software will only use TPM appropriately... and therefore, we'll usually want attestation too.

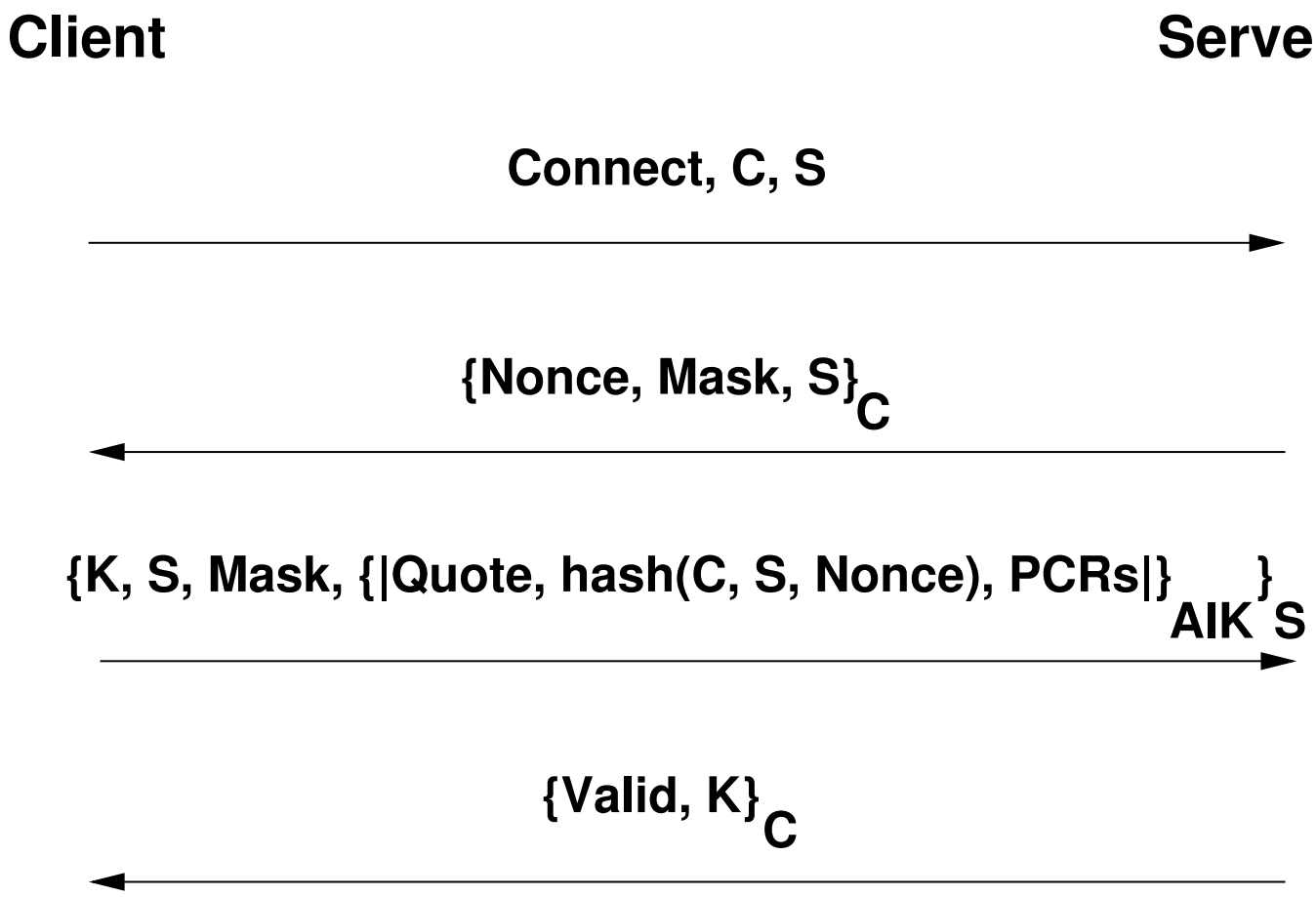
Detour: Network-Accessible TPMs

Systems can allow remote access to some TPM commands.

Network-accessible commands cannot safely be used for machine authentication.

- These commands potentially allow remote parties to masquerade as the TPM's machine
- If a network-accessible TPM violates a protocol's security assumptions, you have a potential problem.
  - Even if your TPMs don't provide access, communications partners may be less constrained.
- Attestation can reduce this risk
  - Verify software stack, remote access configuration
  - Note: Hard today!

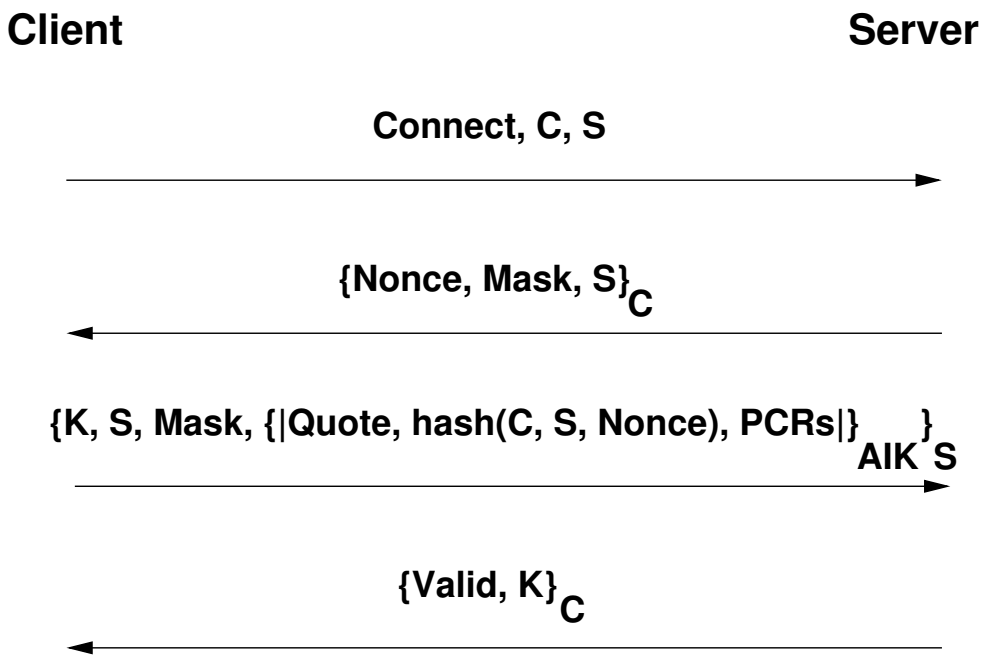
EVA: An Alternate Approach



This substitutes for TLS in our scenario.

The new session key can be used to run a user-authentication protocol, once machine has passed attestation.

EVA Features



- Encrypted nonce authenticates C to S
- Encrypted session key authenticates S to C
- Fresh nonce guarantees fresh quote for S
- Session key secret, shared by S and C

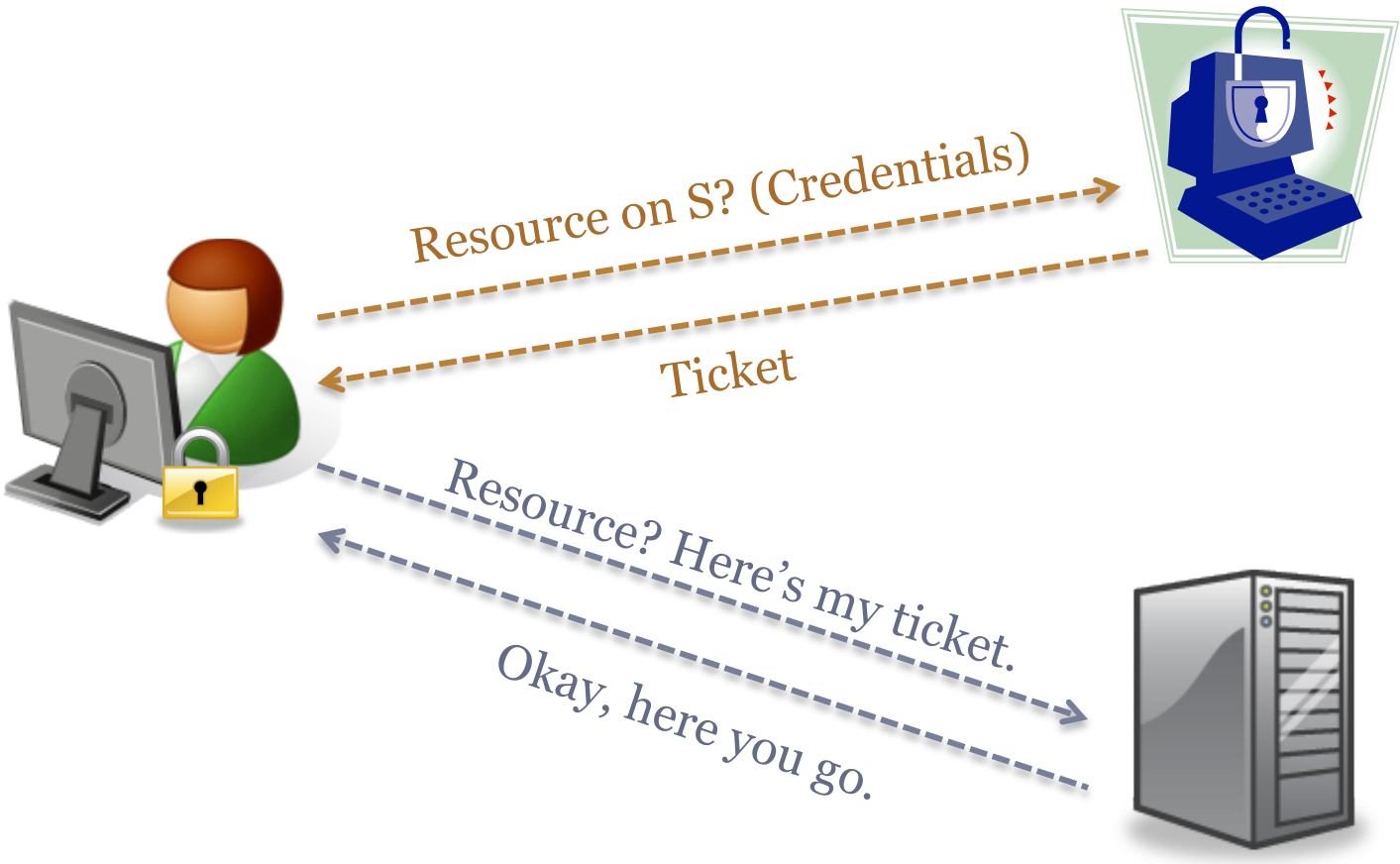
Safer Ways of Using Standards

Could we still use TLS?
Standard protocols can often be extended to use attestation if the protocol issues a certificate, ticket, or other token.

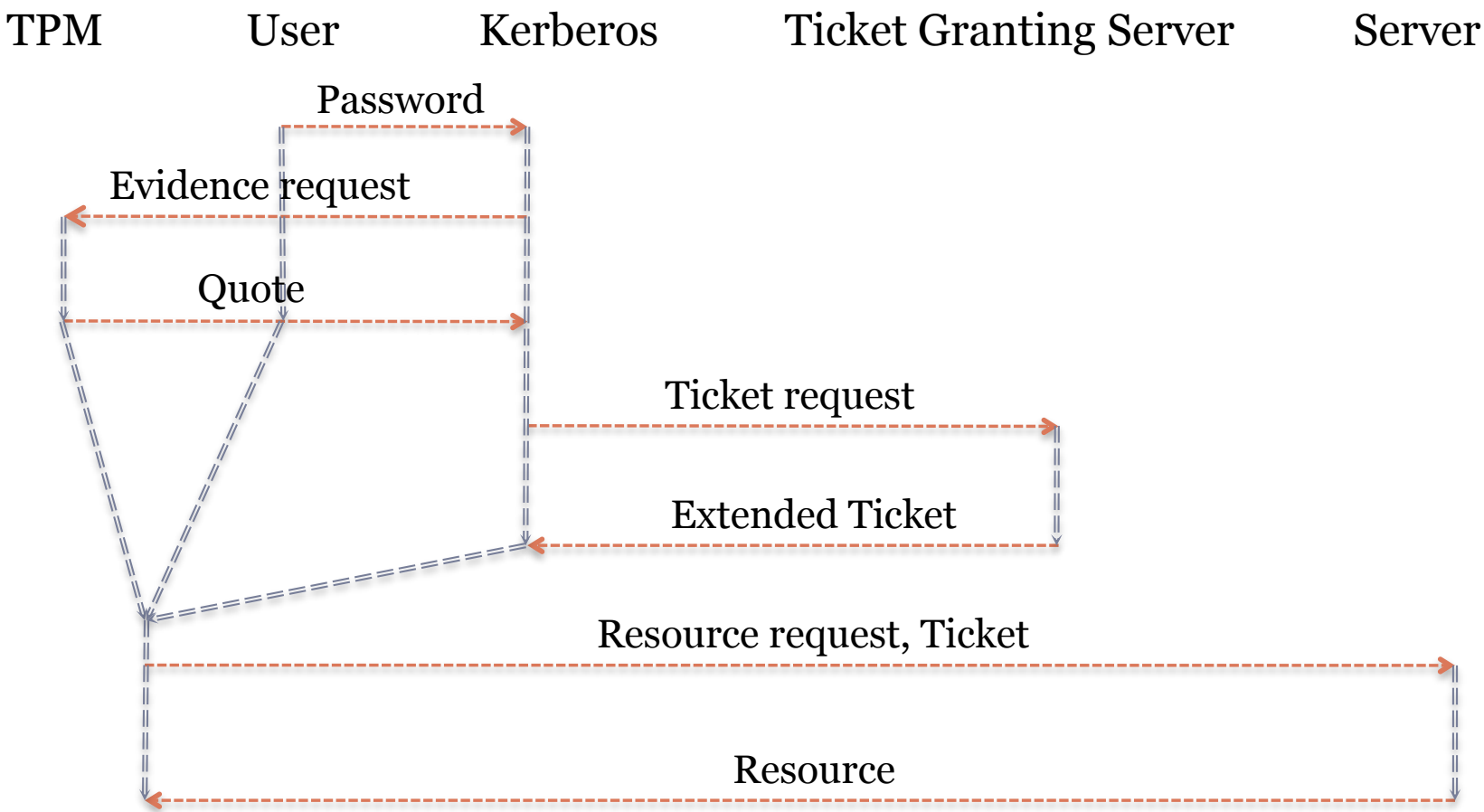
- Change only the certificate issuance portion
  - Can provide incremental upgrade path
- Kerberos: Add machine auth + attestation to original ticket granting
  - User, machine auth, attestation tied together using EVA-like protocol
- TLS: Add attestation to certificate issuance, use short-lived certs

Tradeoff: Frequency of reissuance vs. accuracy.

Kerberos from 50,000 Feet



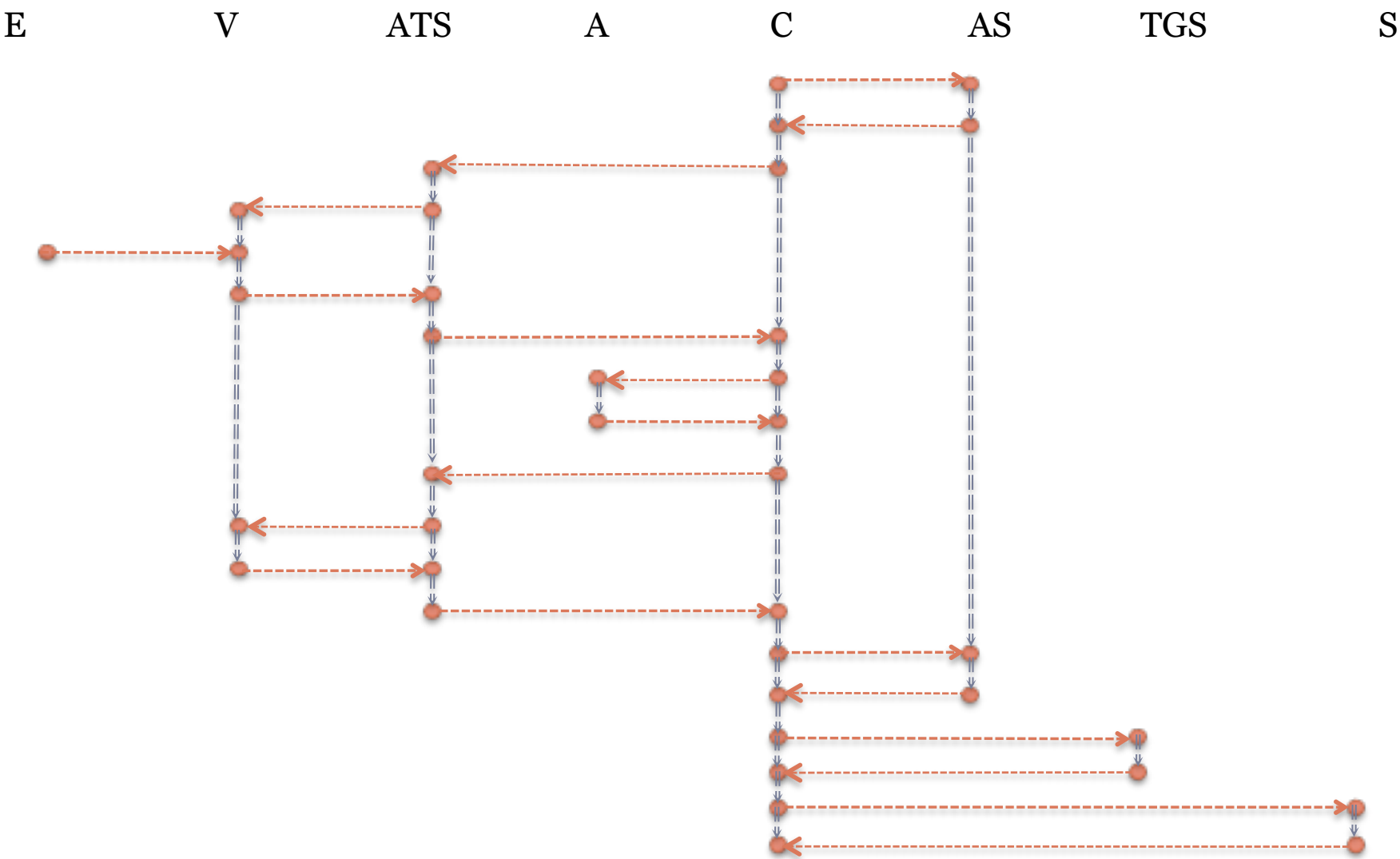
Kerberos With Attestation (KAT)



User ticket now implies verification of client machine.

Services that accept Kerberos tickets require no modification!

Why We Won't Discuss KAT Details



Mutual Attestation

Protocols where multiple parties verify each other

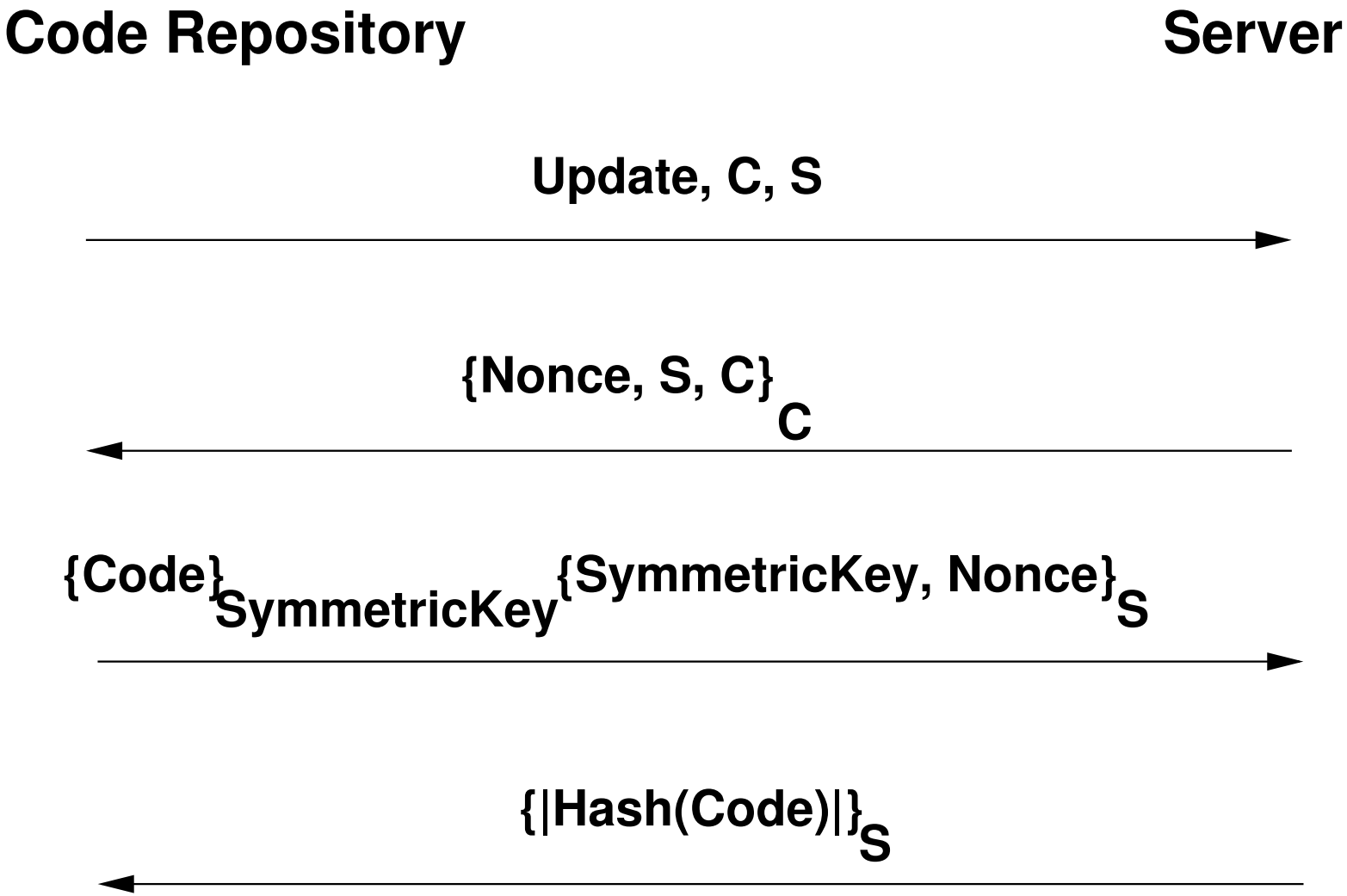
- Very common, highly varied!
- Critical for scenarios handling high-security data
  - Authorized users connecting to sites providing trusted data
  - Upload of sensitive software updates to remote devices
  - Establishing trusted communication among a variety of parties
- May involve trusted third parties
- May need to take privacy into account

Secure Code Upgrade Scenario

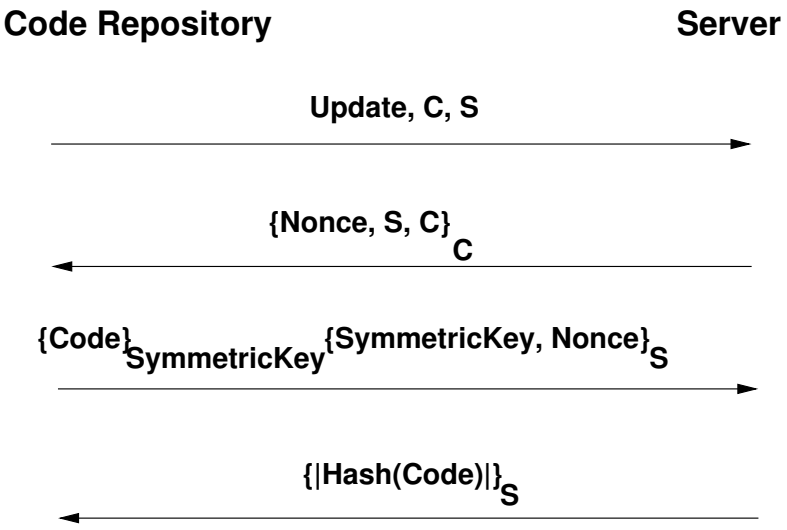
A company with a distributed network of servers wishes to upgrade their proprietary software.

- Servers should only accept software from legitimate central repositories
- Servers should never accept old software
- Proprietary code should only be released to legitimate servers
- All code should be encrypted in transit

Authenticated Code Update Protocol



Authenticated Code Update Protocol



- Here, TPM used for authentication only
  - "C" decryption (binding) key
  - "S" actually two: one binding, one signing
- Fresh symmetric key used for large-scale encryption

Attesting Code Update Protocol

What if we wanted to also attest to the state of one or both machines?

Authenticating protocols using TPM keys can become attesting protocols easily

- CertifyKey can be used to remotely verify PCR bindings on any key
- Sealed data can have PCR constraints
  - To decrypt, TPM must contain correct PCR values
- However, PCR-constraining methods require predictable values
- If there is a range of acceptable values, use quote-based protocol
  - Remember, tying quotes to machine auth requires work

A Few Other Interesting Problems

- Combined hardware user + TPM machine auth
  - Smartcards? Biometric readers?
  - Common challenge: Limited interface
- Trustworthy kiosks: Can I safely use this airport machine to check my e-mail?
- Trustworthy clouds: Can I load my data or trust my results? What am I actually connecting to?
- E-cash and electronic purchases: Do I have control over my virtual wallet?
- ...

Common theme: Need a good architecture to trust, need protocols to verify

Looks Easy, Right?

Protocols can go wrong in many ways.

- Man in the middle
- Message replay
- Message swapping between multiple protocol instances
- Message modification
- Message swapping between multiple protocols!
- ...

Any structural flaw will allow failure regardless of the perfection of the implementation.

A protocol designed for a particular purpose may have "flaws" when used out of context.

Know Your Goals

When evaluating a protocol, consider:

- What information do you need?
- What information needs to be current?
- What information needs to be correlated?
- What needs to be kept secret?
- What does each participant need to know when the protocol completes?

Normally, we discourage grow-your-own protocols. However, there is a shortage of protocols that fully utilize the TPM!

Doing This At Home

Cryptographic Protocol Shapes Analyzer (CPSA)

- Tool from MITRE for analyzing structural properties of protocols
- Discovers all possible executions of a protocol (*shapes*) using the strand space method
- What we used to design and analyze these examples
- Implementation being formally proved correct
- Note: Does not support every possible cryptographic feature
- Publicly available: http://hackage.haskell.org/package/cpsa

Conclusion

- The TPM, particularly its key infrastructure, provides essential support for cryptographic protocols
- Real-world scenarios require a variety of attestation and authentication protocols
- It's easy to make mistakes that cause security failure; however:
- We can design reliable protocols that meet the needs of many important scenarios with today's TPM.

Questions?

- asegall@mitre.org
- CPSA: http://hackage.haskell.org/package/cpsa

For more information on learning protocol design and analysis, contact me.