

Protocol for Protecting Against Impersonation

Protocol for Protecting Against Impersonation

Given:

A Monitor wishing to "prove" its identity

Protocol for Protecting Against Impersonation

Given:

A Monitor wishing to "prove" its identity

A Client wishing to "verify" the identity of the Monitor

Protocol for Protecting Against Impersonation

Given:

A Monitor wishing to "prove" its identity

A Client wishing to "verify" the identity of the Monitor

An attacker wishing to impersonate the Monitor to the Client

Protocol for Protecting Against Impersonation

Given:

A Monitor wishing to "prove" its identity

A Client wishing to "verify" the identity of the Monitor

An attacker wishing to impersonate the Monitor to the Client

Rules:

The attacker can generate keys just like the Monitor can

Protocol for Protecting Against Impersonation

Given:

A Monitor wishing to "prove" its identity

A Client wishing to "verify" the identity of the Monitor

An attacker wishing to impersonate the Monitor to the Client

Rules:

The attacker can generate keys just like the Monitor can

The attacker has a "prover" just like the one used by the Monitor

Protocol for Protecting Against Impersonation

Given:

A Monitor wishing to "prove" its identity

A Client wishing to "verify" the identity of the Monitor

An attacker wishing to impersonate the Monitor to the Client

Rules:

The attacker can generate keys just like the Monitor can

The attacker has a "prover" just like the one used by the Monitor

But only sees the outputs, not internal coin-flips, etc.

Protocol for Protecting Against Impersonation

Given:

A Monitor wishing to "prove" its identity

A Client wishing to "verify" the identity of the Monitor

An attacker wishing to impersonate the Monitor to the Client

Rules:

The attacker can generate keys just like the Monitor can

The attacker has a "prover" just like the one used by the Monitor

But only sees the outputs, not internal coin-flips, etc.

Attacker may query the "prover" some small number of times
(that is, tries to reveal key information from the "prover")

Protocol for Protecting Against Impersonation

Client

(verifier)

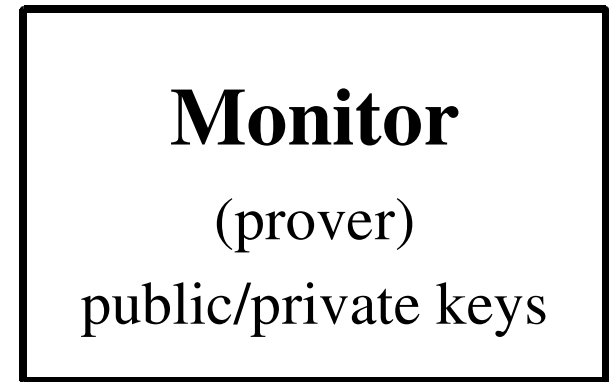
public/private keys

Monitor

(prover)

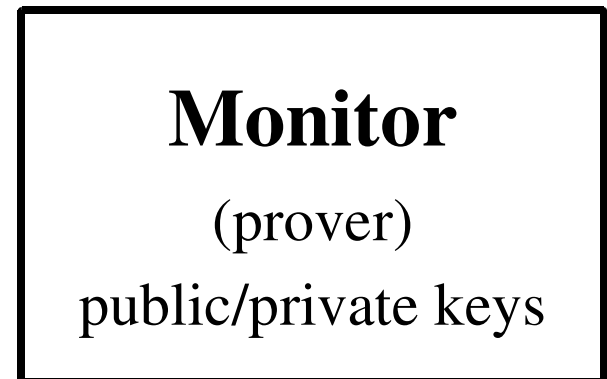
public/private keys

Protocol for Protecting Against Impersonation



Monitor to "prove" itself to Client

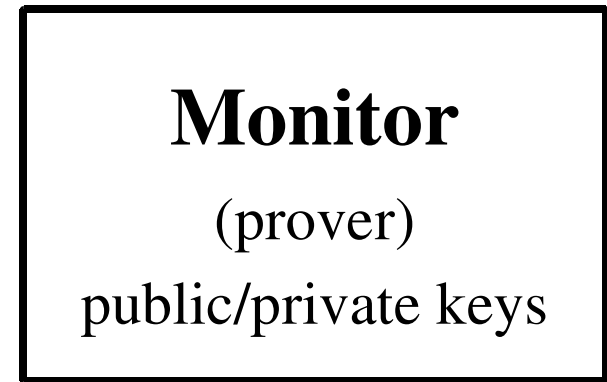
Protocol for Protecting Against Impersonation



Monitor to "prove" itself to Client

Monitor gets Client's public key

Protocol for Protecting Against Impersonation

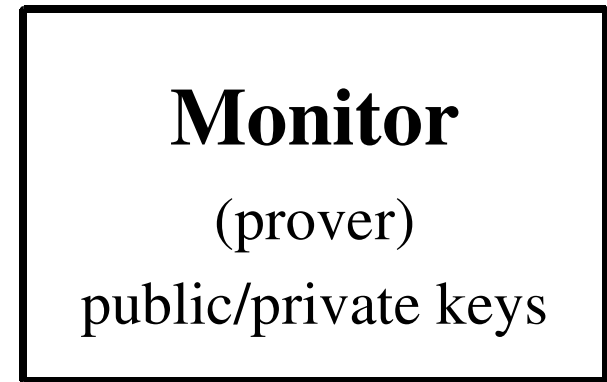


Monitor to "prove" itself to Client

Monitor gets Client's public key...

and using its keys proves "I know Client's secret key
or I know Monitor's private key"

Protocol for Protecting Against Impersonation



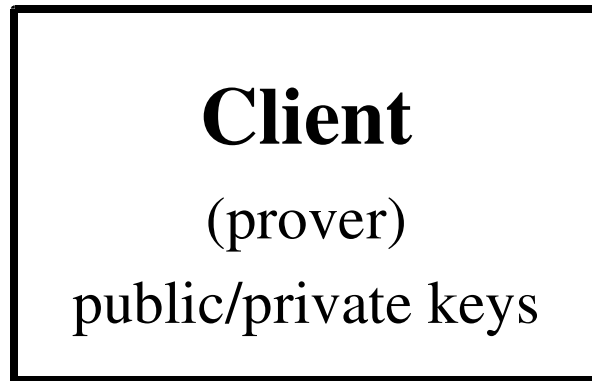
Monitor to "prove" itself to Client

Monitor gets Client's public key...

and using its keys proves "I know Client's secret key
or I know Monitor's private key"

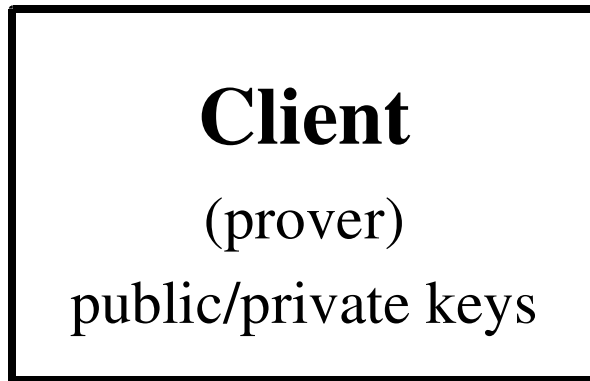
If no info is released saying which, only Client can
be sure he is talking to Monitor since attacker only

Protocol for Protecting Against Impersonation



Suppose Client wants to be the man in the middle:
Client tries to make Horowitz think he is the Monitor

Protocol for Protecting Against Impersonation

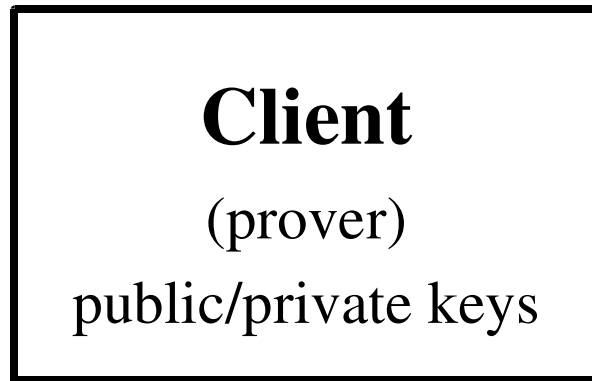


Suppose Client wants to be the man in the middle:

Client tries to make Horowitz think he is the Monitor

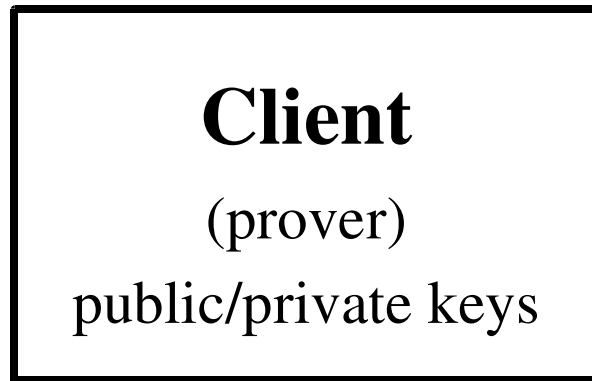
But that requires no communication with Monitor

Protocol for Protecting Against Impersonation



Suppose Client wants to be the man in the middle:
Client tries to make Horowitz think he is the Monitor
But that requires no communication with Monitor
If Client's attack succeeds, then Client knows Monitor's private key.

Protocol for Protecting Against Impersonation



Suppose Client wants to be the man in the middle:

Client tries to make Horowitz think he is the Monitor

But that requires no communication with Monitor

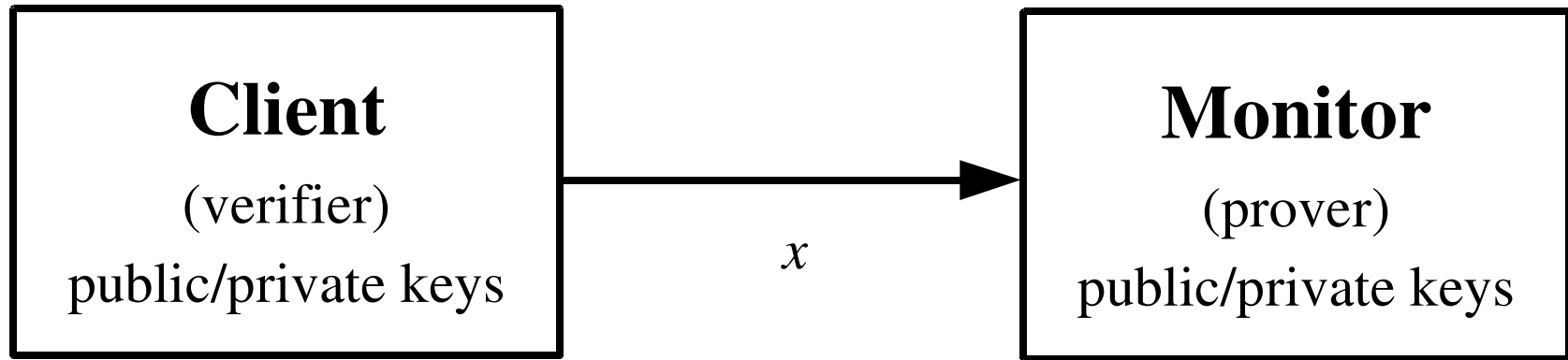
If Client's attack succeeds, then Client knows Monitor's private key.

Hence Client's attack cannot succeed.

Protocol for Protecting Against Impersonation

More Specifically...

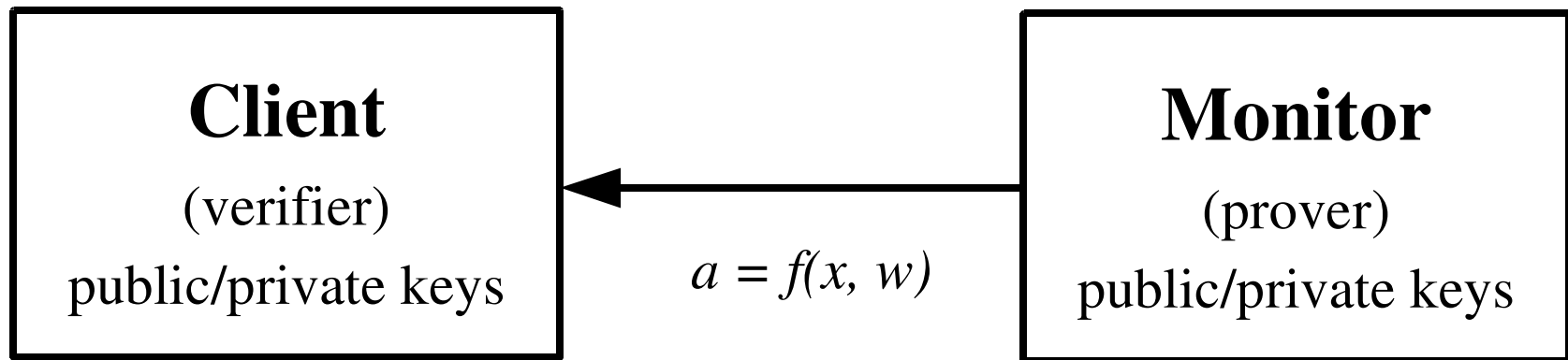
Protocol for Protecting Against Impersonation



Protocol:

Client remembers and sends random number x to Monitor

Protocol for Protecting Against Impersonation

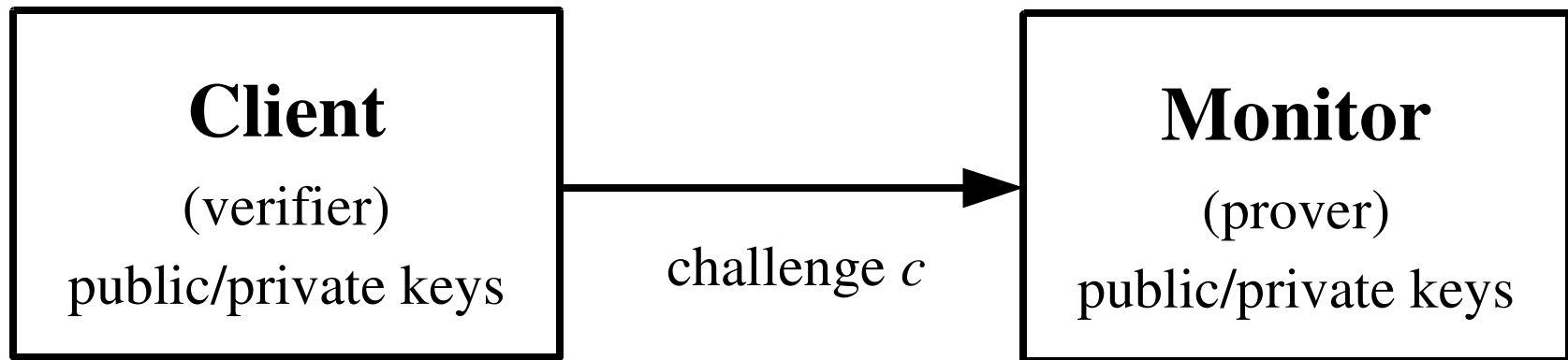


Protocol:

Client remembers and sends random number x to Monitor

Monitor computes and sends message a from x and w

Protocol for Protecting Against Impersonation



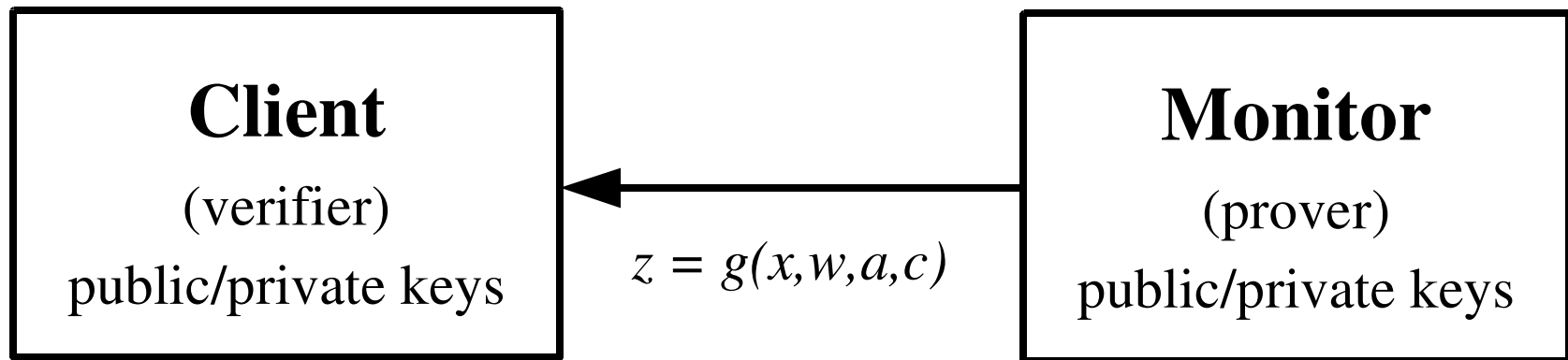
Protocol:

Client remembers and sends random number x to Monitor

Monitor computes and sends message a from x and w

Client sends a "challenge" number c to Monitor

Protocol for Protecting Against Impersonation



Protocol:

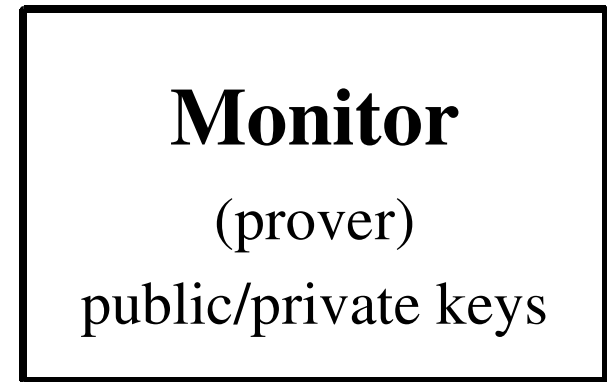
Client remembers and sends random number x to Monitor

Monitor computes and sends message a from x and w

Client sends a "challenge" number c to Monitor

Monitor computes and sends response z to Client

Protocol for Protecting Against Impersonation



Protocol:

Client remembers and sends random number x to Monitor

Monitor computes and sends message a from x and w

Client sends a "challenge" number c to Monitor

Monitor computes and sends response z to Client

Client verifies validity of the exchange