

A Dependently Typed Attestation Protocol Language

Anna Fritz and Perry Alexander

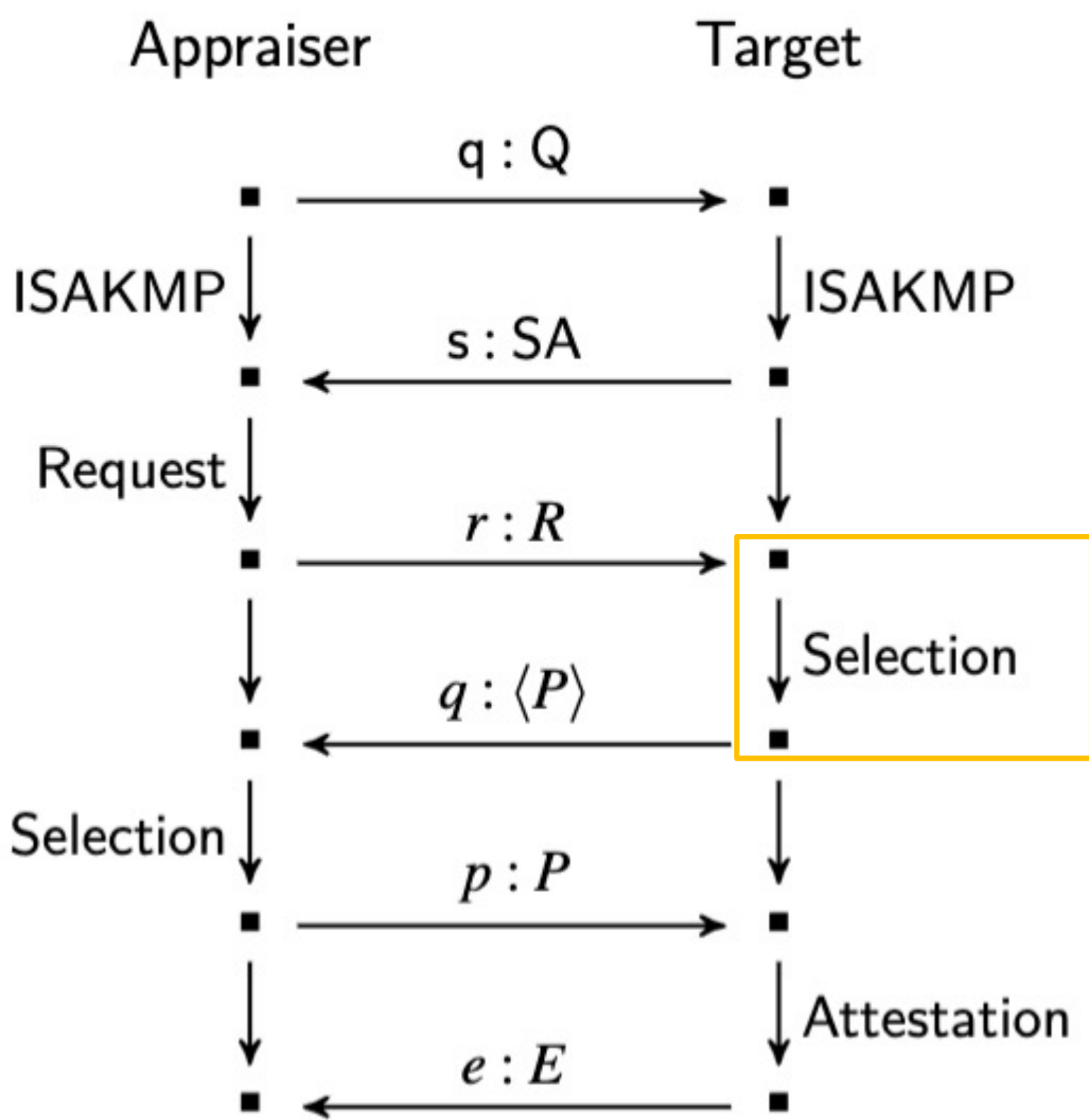
University of Kansas

Motivation

Remote attestation is the act of making trust decisions about a communicating party. During this process, an appraiser asks a target to execute an attestation protocol that generates and returns evidence. The appraiser then evaluates the evidence to make claims about the target. We use the language Copland to formally specify attestation protocols and introduce Copland centered negotiation as prerequisite to attestation. The goal of negotiation is to generate a protocol that meets the target’s needs for constrained disclosure and the appraiser’s desire for comprehensive information.

The Process of Negotiation

- ISAKMP
Both the target and appraiser participate in ISAKMP to develop a security association (SA) which defines a common vocabulary and establishes a secure channel for latter communications.
- Request
The request, $r:R$, is a set of Copland phrases that are sent by the appraiser. It suggests the desired terms for attestation.
- Target’s Selection
The target receives the request and generates a proposal, $q:\langle P \rangle$, which is a list of Copland terms. To produce this list, the target applies its *selection policy* to generate all terms that satisfy the request. Next, the target applies its *privacy policy* to filter out any terms that may expose sensitive information. Using both policies, the target is able to gather terms for the proposal.
- Appraiser’s Selection
The appraiser receives the request and applies some ordering function to the proposal to select the best term for attestation.
- Attestation
The target receives the chosen term for attestation. The target evaluates the Copland phrase, $p:P$ to evidence, $e:E$.



Using Dependent Types for Policy

We propose two different ways of writing a dependently typed privacy and selection policy. These policies allow the target to find a list of protocols that fit the appraiser’s request while not disclosing sensitive information.

Indexed Types

We can index the term language with the type of evidence produced. This gives us a static check as to if the policy is satisfied.

```
Inductive evidence : place -> Type :=  
  
Fixpoint privPolicy p (e:evidence p) : Prop :=  
  
Inductive term p (e:evidence p) -> Type :=  
| TMeas : forall e, term e  
| THash : forall e, term e -> privPolicy (EHash p) -> term (EHash p)  
...
```

Subset Types

The subset type implements set comprehension to build a set such that all the elements of the set satisfy the privacy policy.

```
Inductive evidence : Type :=  
  
Fixpoint privPolicy (e:evidence) : Prop :=  
  
Inductive term : evidence -> Type :=  
| TMeas : forall e, term e  
| THash : forall e, term e -> term (EHash)  
...  
  
Definition selectDep e ( _ :term e) := { t : term e | privPolicy e }
```



8TH ANNUAL
HOT TOPICS in the SCIENCE OF SECURITY
APRIL 13-15, 2021 | VIRTUAL