Predicate-Based Key Exchange

James Birkett   Douglas Stebila

Information Security Institute
Queensland University of Technology

15^{th} Australasian Conference on Information Security and Privacy, 2010

Outline

Outline

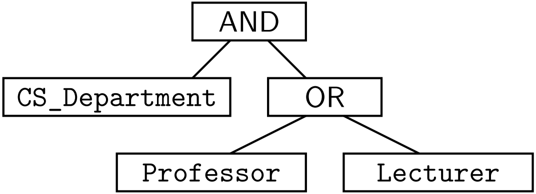
1. Background
   - Cryptographic Primitives
   - Key Exchange
2. Motivation
   - A Hypothetical Example
3. Our Contribution
   - Security Model
   - Generic Construction

Identity-based Cryptography

- Key generation centre (KGC) generates public parameters and master secret.
- KGC gives private keys to users based on their *identity*.
- Identities may be names, email addressess etc.
  E.g "`bob@example.com`", "`James Birkett`"
- Sender uses an identity to encrypt.

Attribute-based Cryptography

- KGC gives private keys to users based on their *attributes*.
- Attributes are boolean values.
  E.g “`CS_department=true`”, “`Professor=true`”, “`Student=false`”
- The list of attributes is fixed at setup.
- Sender uses an access structure to encrypt.

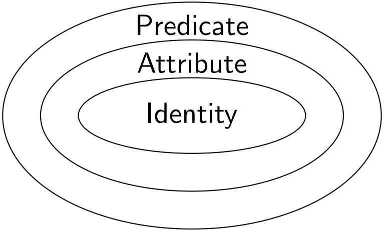


- Access structures limited to AND, OR and threshold operations.

Predicate-based Cryptography

- Generalises attributes to credentials.
- Credentials are name-value pairs.
  E.g "`Department=CS`", "`Department=Maths`"
- The list of credentials need not be fixed at setup.
- More complex access structures avaliable, e.g equality, subset or comparison operations as well as AND, OR and threshold.
- We call these access structures *predicates*, $\Phi(C)$.

Relationship

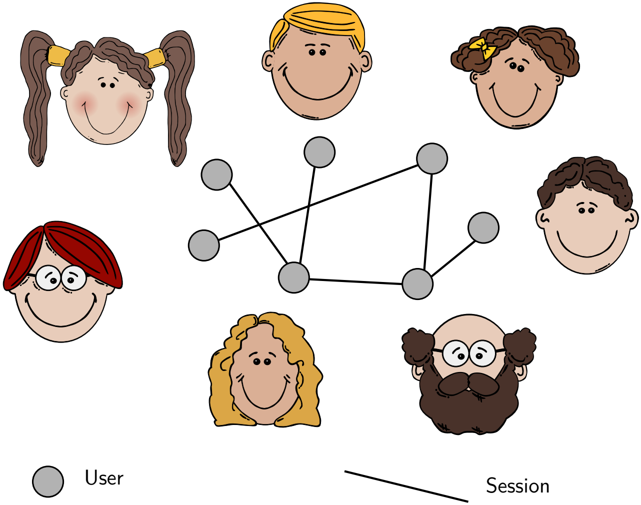


-Based Cryptography

- Attribute-based cryptography is a special case of Predicate-based cryptography.
- Our model and generic construction handles both.

Outline

Key-exchange



Outline

Therapy

With the Society of Secretive Psychologists.

Alice Needs:

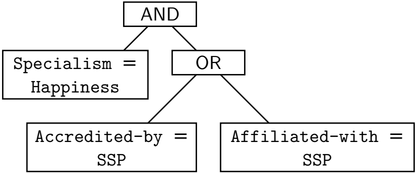
- A registered psychologist.
- A private channel.
- Anonymity.

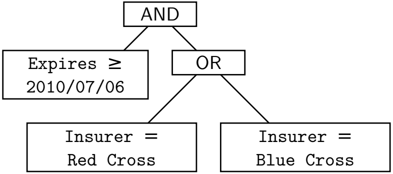
Bob Needs:

- A private channel.
- Proof of insurance.

Therapy

How Predicate-Based Key Exchange Could Help





Predicate-based Key Exchange

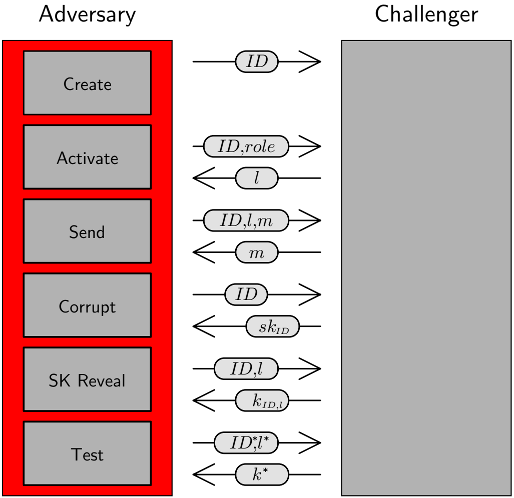
- If you do not need anonymity (credential-privacy) then you do not need predicate-based key exchange!
- Instead you may simply present a list of credentials signed by the trusted third party.

Outline

Identity-based Key-Exchange Security

- Challenger maintains a list of users $ID_1, \ldots, ID_n$.
- Each user has a secret key sk_{ID}.
- Each user U_{ID} maintains a list of sessions.
- Each session contains:
  - The ID of the peer ID'.
  - A list of messages exchanged, $m_1, \ldots, m_r$.
  - A state variable.
  - (Possibly) a key $k_{ID,\ell}$.

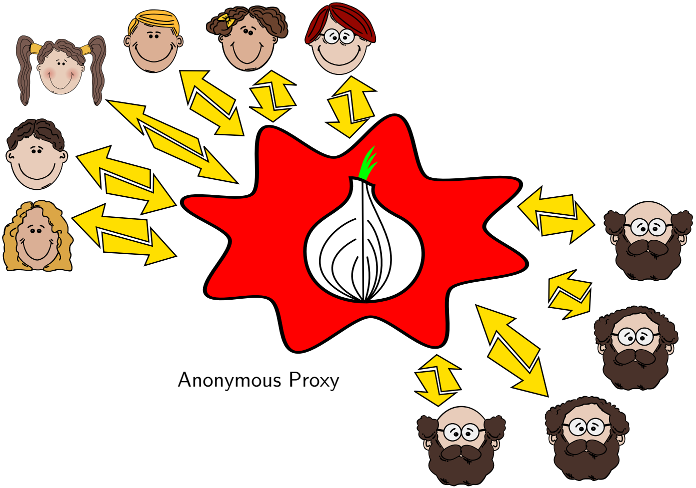
Identity-based Key-Exchange Security (cont)



Separating credentials from addresses

- Unique identities incompatible with credential-privacy.
- Cannot direct messages using credentials.
- Instead use user numbers independent from credentials for addressing.

Addressing the Addressing Problem
Attempt 1

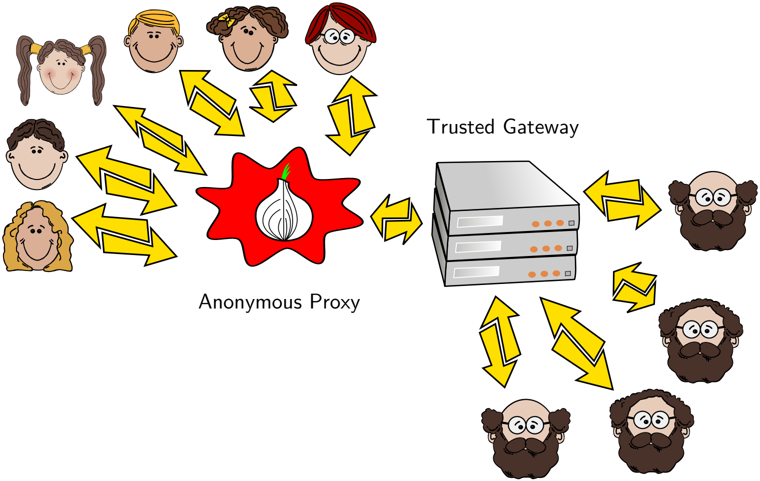


Addressing the Addressing Problem

Attempt 1

- Anonymous proxy servers / routing services may hide initiator's address.
- Initiator still needs to direct messages to the recipient.

Addressing the Addressing Problem
Attempt 2

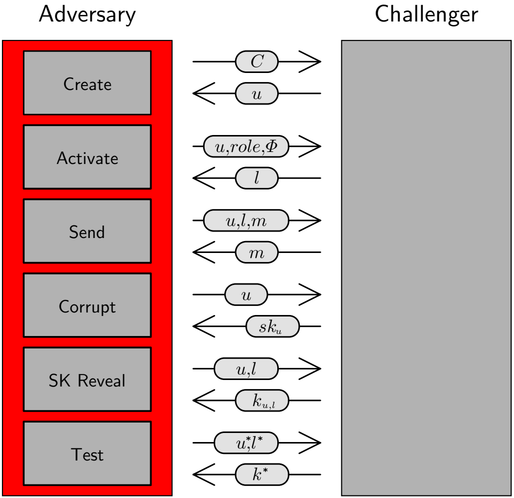


Addressing the Addressing Problem

Attempt 2

- Society of Secretive Psychologists operates their own trusted gateway.
- Gateway knows credentials of each psychologist.
- Gateway can choose psychologist satsifying a given predicate Φ.

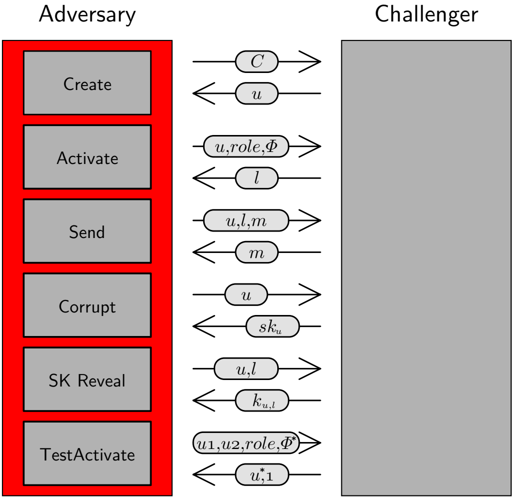
Session-Key Security



Session-Key Security (cont)

- Adversary may not corrupt any user such that $\Phi(C) = 1$.
  - Forward Security: adversary may corrupt user after the Test query.
- Adversary may not SKReveal u^*, ℓ^*.
- Adversary may not SKReveal u, ℓ if $s_{u,\ell}$ is a peer of s_{u^*,ℓ^*}.

Credential Privacy



Credential Privacy (cont)

- Φ^* must satisfy $\Phi^*(C_{u_0}) = \Phi^*(C_{u_1})$
- Adversary may not Activate u^*.
- Adversary may not Corrupt U_{u_0} or U_{u_1}.
- Adversary may not SKReveal $u^*, 1$.
- Adversary may not SKReveal u, ℓ if $s_{u,\ell}$ is a peer of $s_{u^*,1}$.

Credential Privacy and Unlinkability

Credential Privacy	Unlinkability
No user can determine anything about your credentials other than $\Phi(C)$, i.e. whether you satisfy their predicate.	You cannot tell if two sessions are with the same person or not.

- Credential privacy implies Unlinkability.

Outline

Protocol Flow

$\Pi_{\mathcal{S},\mathbb{G}}$ – Protocol flow

Initiator		Responder
secret key sk_I responder predicate Φ_I		secret key sk_R initiator predicate Φ_R
$x \xleftarrow{R} \mathbb{Z}_q$ $X \leftarrow g^x$		
	$\xrightarrow{X, \Phi_I}$	$y \xleftarrow{R} \mathbb{Z}_q$ $Y \leftarrow g^y$ $\sigma_R \leftarrow \mathrm{Sign}(sk_R, (\mathbf{resp}, X, \Phi_I, Y, \Phi_R), \Phi_I)$
If $\neg\mathrm{Verify}((\mathbf{resp}, X, \Phi_I, Y, \Phi_R), \Phi_I, \sigma_R)$: $status \leftarrow$ **Failed** Abort $\sigma_I \leftarrow \mathrm{Sign}(sk_I, (\mathbf{init}, X, \Phi_I, Y, \Phi_R, \sigma_R), \Phi_R)$ $Z \leftarrow Y^x$ $k \leftarrow H(X, \Phi_I, Y, \Phi_R, Z)$ $status \leftarrow$ **Established**	$\xleftarrow{Y, \Phi_R, \sigma_R}$	
	$\xrightarrow{\sigma_I}$	If $\neg\mathrm{Verify}((\mathbf{init}, X, \Phi_I, Y, \Phi_R,), \Phi_R, \sigma_I)$: $status \leftarrow$ **Failed** Abort $Z \leftarrow X^y$ $k \leftarrow H(X, \Phi_I, Y, \Phi_R, Z)$ $status \leftarrow$ **Established**

Summary

- Existing key-exchange models identify credentials with addresses.
- Predicate-based models must find an alternative to this.
- Predicate-based key exchange is only useful if you require credential-privacy.

- Future work
  - Adapt the model to include state-reveal or ephemeral-key-reveal queries.
  - Develop constructions which are secure against these queries.