KEY ESTABLISHMENT PROTOCOLS AND THE DIFFIE-HELLMAN PROBLEM

Simon Blake-Wilson

Certicom Corp.
and
CACR, University of Waterloo

November 2, 1999

Overview

Purpose of this talk:

- Discuss some of the subtleties involved in designing key establishment protocols.
- Survey some formal approaches to protocol design.
- Stimulate interest in the problems.

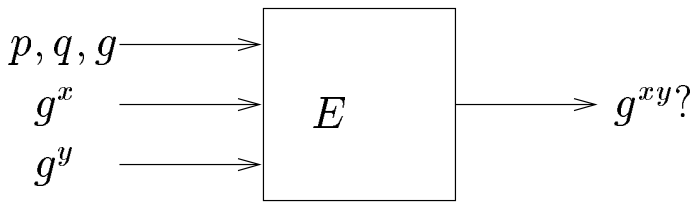
Outline

- Introduction
- Some protocols . . .
- Goals (Informal)
- Design methodologies
- Two formal approaches
- Summary and open problems

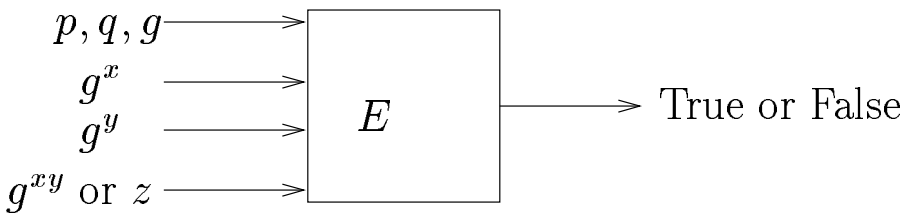
1. Introduction

- Key establishment protocols are used over an open network by two parties to establish a shared secret key, called the session key.
- The session key is subsequently used to secure a conversation between the parties.
- Ideally, the established key should have precisely the same attributes as a key established face-to-face.
- Public-key protocols require parties to share a priori only some authentic information.

Today's primary building blocks are the Diffie-Hellman problem:



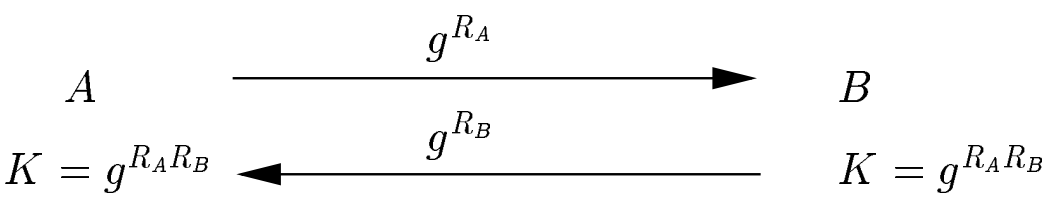
And the decision Diffie-Hellman problem:



We'll also use a secure signature scheme and a secure MAC scheme.

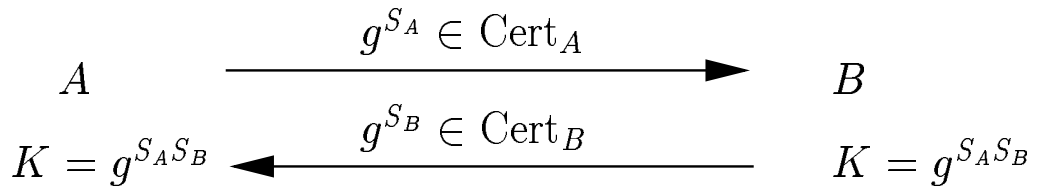
2. Some Protocols ...

Ephemeral Diffie-Hellman protocol:



Problems: man-in-the-middle attack, small subgroup attack.

Static Diffie-Hellman protocol:

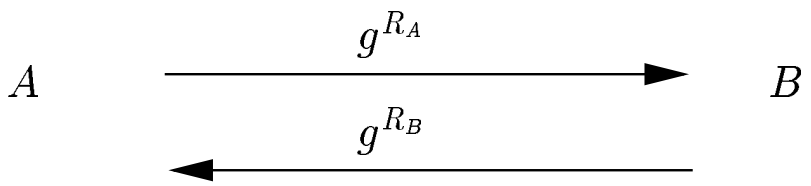


Problems: known-key attack.

Notation

A, B	(honest) entities
E	adversary
$\longrightarrow$	communications over an open (unauthenticated) network
$\longrightarrow$	communications over an authenticated network
p, q, g	Diffie-Hellman parameters (fixed system-wide)
R_A, g^{R_A}	A's short-lived (ephemeral) key pair
S_B, g^{S_B}	B's long-lived (static) key pair

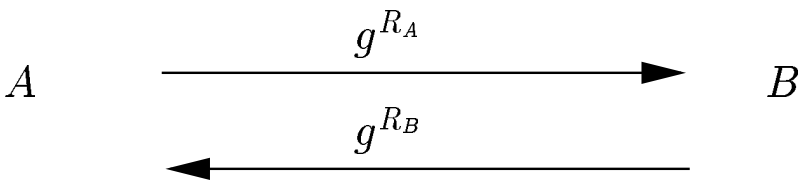
MTI protocol ... combining static and ephemeral keying material:



$$K = g^{R_A S_B + S_A R_B}$$

Problems: small subgroup attack, forward secrecy.

'Improved' MTI protocol:



$$K = g^{R_A S_B + S_A R_B + R_A R_B}$$

Problems: known-key attack.

A variety of "attacks" have been devised:

- Man-in-the-middle attacks
- Reflection attacks
- Interleaving attacks
- Known-key attacks
- Unknown key-share attacks
- ...

Such attacks are typically subtle and require little computational overhead.

(Formal) analysis is needed to avoid the use of flawed protocols.

3. Goals (Informal)

Security goals . . . supply as needed by application:

- *Implicit key authentication* (of B to A): A is assured that no entity other than B can possibly compute the session key.
- *Explicit key authentication* (of B to A): A is assured in addition that B actually has computed the session key.
- *Unknown key-share resilience*: A is assured that B does not mistakenly think a key shared with A is shared with a third party E.

Security goals . . . supply as needed based on threat analysis:

- *Known-key security*: compromise of session key(s) does not compromise other session keys.
- *Forward secrecy*: compromise of A's and/or B's static private keys does not compromise previous session keys.
- *Key-compromise impersonation*: compromise of A's static private key does not enable E to impersonate other entities to A.
- *Weak agreement* vs *transport* vs *strong agreement*
- . . .

Efficiency goals:

- Minimal number of passes
- Low computation overhead
- Low communication overhead

Other desirable attributes:

- Anonymity
- Non-interactiveness
- Non-reliance on encryption
- Non-reliance on hash functions
- Non-reliance on time-stamping

4. Design Methodologies

How can we design and analyze protocols?

- ad hoc (heuristic security)
  - helps avoid known attacks
  - attack-response design not successful
- formal methods
  - successful at finding flaws
  - assurance limited to formal system
- complexity-theoretic ("provable security")
  - reduction to hard problem
  - formal definition of goals
  - security asymptotic

Complexity-theoretic methodology appears to offer most assurance.

Disclaimer:

> If it's provably secure,
> it's probably not.
>
> — Lars Knudsen

Complexity theoretic design involves:

1. Specification of model
2. Definition of goals within this model
3. Statement of assumptions
4. Description of protocol
5. Proof that the protocol meets its goals within the model

Informal definition of goals, proposed protocols:

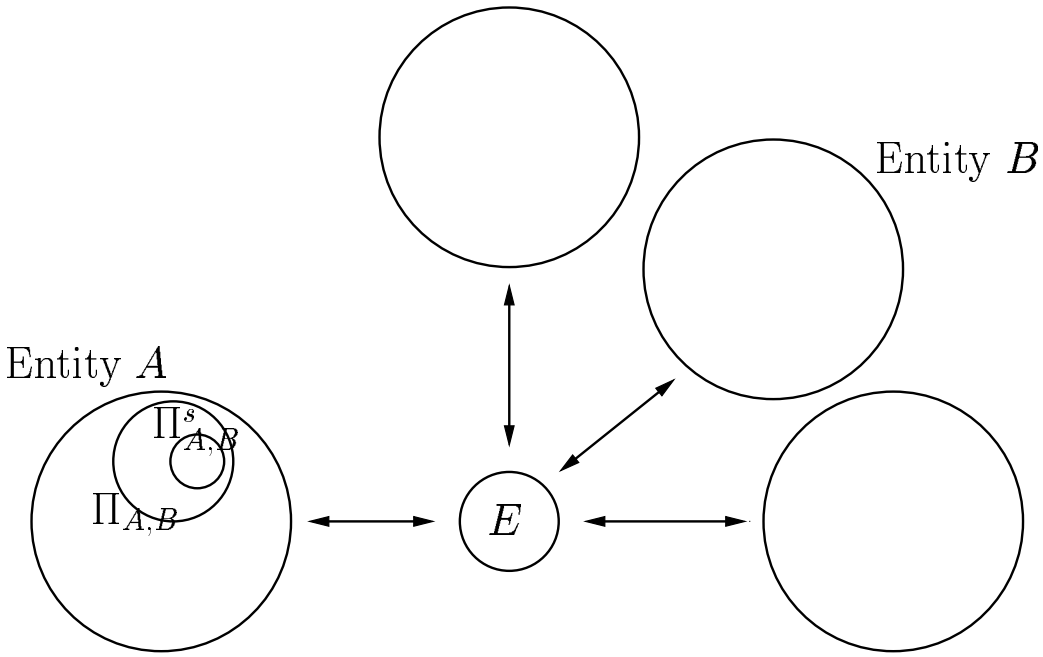
- Bird, ... ,Yung ('91)
- Diffie, van Oorschot, Wiener ('92)

Formal specification of model, definition of goals, proposed protocols, proof of security:

- Bellare, Rogaway ('93)
- Bellare, Rogaway ('95)
- Shoup, Rubin ('96)
- Blake-Wilson, Menezes ('97)
- Blake-Wilson, Johnson, Menezes ('97)
- Bellare, Canetti, Krawczyk ('98)
- Mayer, Yung ('99)
- Shoup ('99)

5. Bellare-Rogaway Approach

Model of open network:



E, the adversary, controls all communications. E is equipped with oracles $\Pi^s_{A,B}$.

E can make queries:

`Send`

`Corrupt`

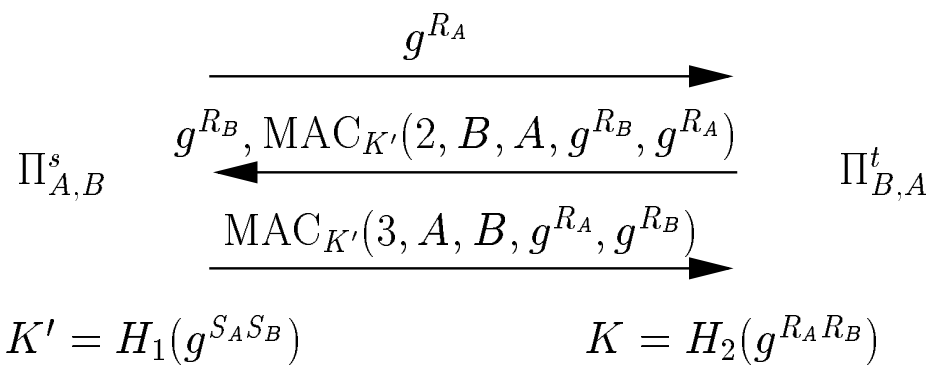
`Reveal`

A key establishment protocol is secure if:

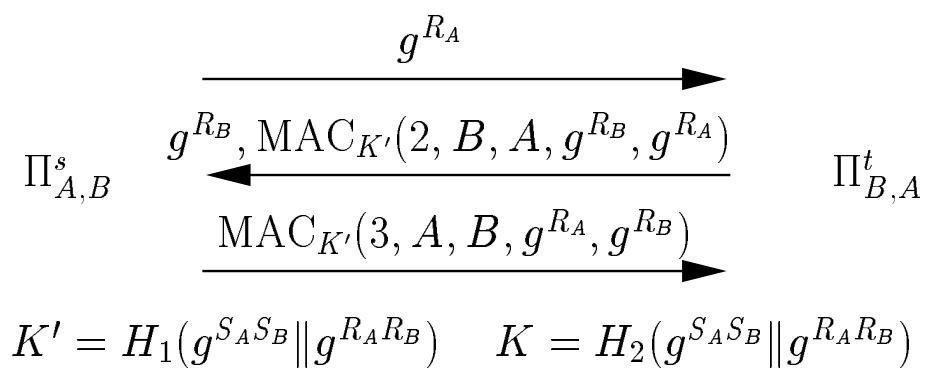
1. The protocol does what it's supposed to in the absence of an adversary.
2. $\Pi^s_{A,B}$ only accepts if E has transmitted flows faithfully between $\Pi^s_{A,B}$ and some $\Pi^t_{B,A}$.
3. E cannot distinguish a session key held by a $\Pi^s_{A,B}$ oracle from a random key, except by trivial means.

Designed to imply mutual explicit key authentication, unknown key-share resilience, known-key security, (and forward secrecy).

Protocols that can be shown to meet this definition - protocol 1:



And protocol 2:



Theorem: [BJM97] Protocol 1 is a secure key establishment protocol provided:

1. H_1 and H_2 are independent random oracles.
2. The Diffie-Hellman problem is hard.
3. MAC is secure.

6. Bellare-Canetti-Krawczyk Approach

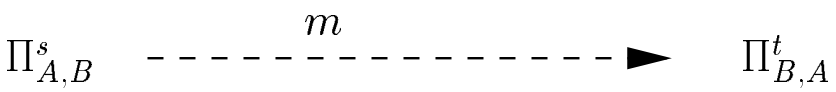
Model can be thought of in terms of the Bellare-Rogaway model.

Open network in which E is allowed to send oracles any messages. Authenticated network in which E is restricted to sending bona fide messages.

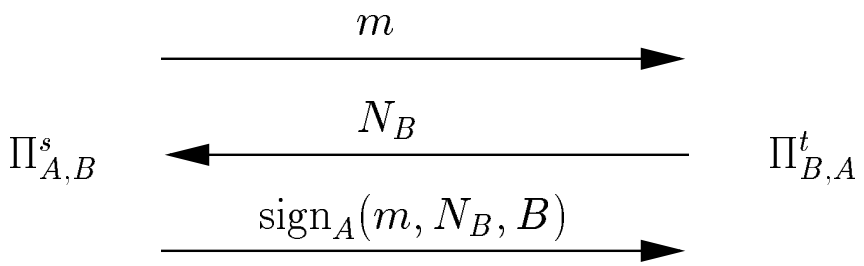
Oracles and E log important information in transcript.

A key establishment protocol (in open or authenticated networks) is secure if it emulates an idealized key establishment protocol.

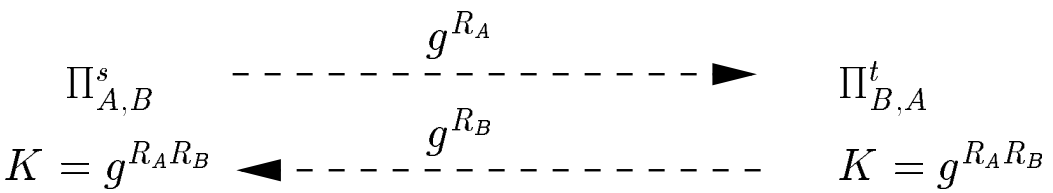
Useful tool: the message transmission (MT) protocol in an authenticated network:



The following protocol, known as an MT-authenticator, emulates the MT protocol in an open network:

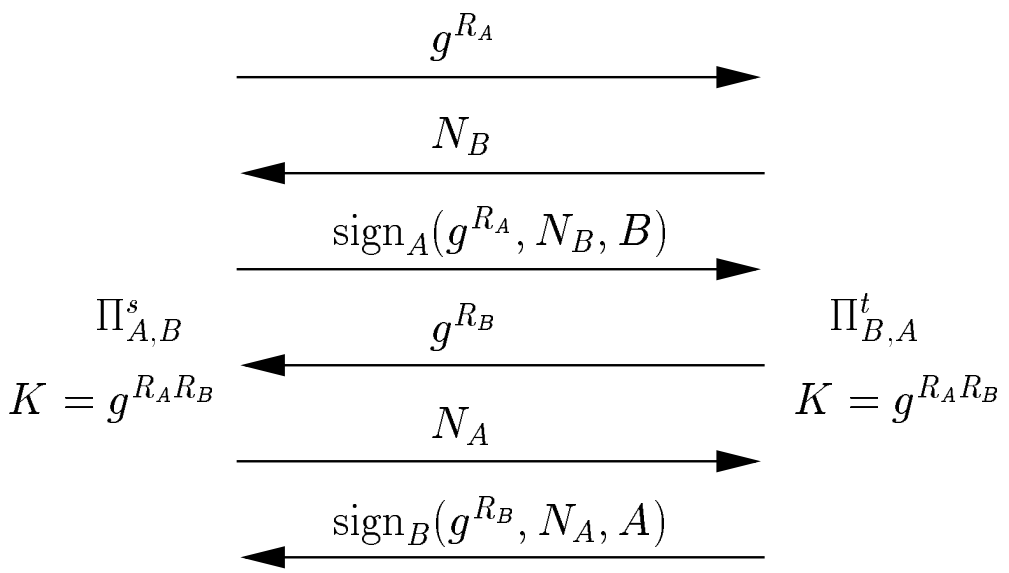


Build a key establishment protocol from the ephemeral Diffie-Hellman protocol in an authenticated network:

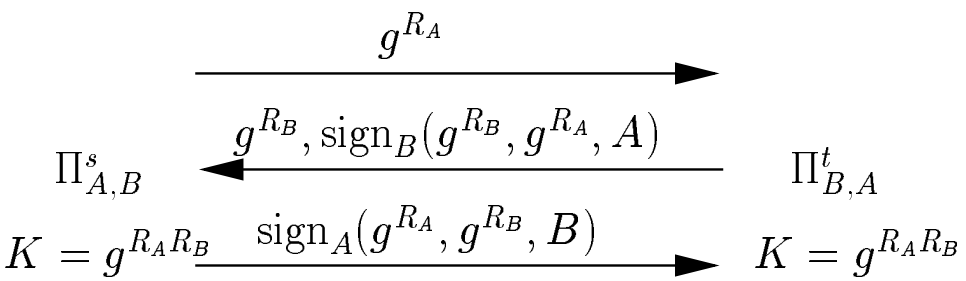


Apply the MT-authenticator to the flows of ephemeral Diffie-Hellman:

$C_{\lambda_{\mathsf{sig}}}(DH)$:



Combining flows, we obtain protocol 3:



Theorem: [BCK98,S99] Protocol 3 is a secure key establishment protocol provided:

1. The decision Diffie-Hellman problem is hard.
2. Signature scheme is secure.

7. Conclusions

- Building key establishment protocols using the Diffie-Hellman problem is non-trivial.
- Formal design methodology appealing way to avoid subtle flaws.
- Further study needed.

8. Directions for Future Work

- Compare and enhance models and security definitions.
- Exact security analysis.
- Extend models to encompass subsequent use of session key.
- Extend models to goals like unilateral key authentication.

9. Further reading!

Adapting the Bellare-Rogaway approach to the public-key setting:
`www.cacr.math.uwaterloo.ca/ sblakewi/`

The Bellare-Canetti-Krawczyk approach:
`www-cse.ucsd.edu/users/mihir/`

Critique of models and enhancements:
`www.shoup.net/papers/`