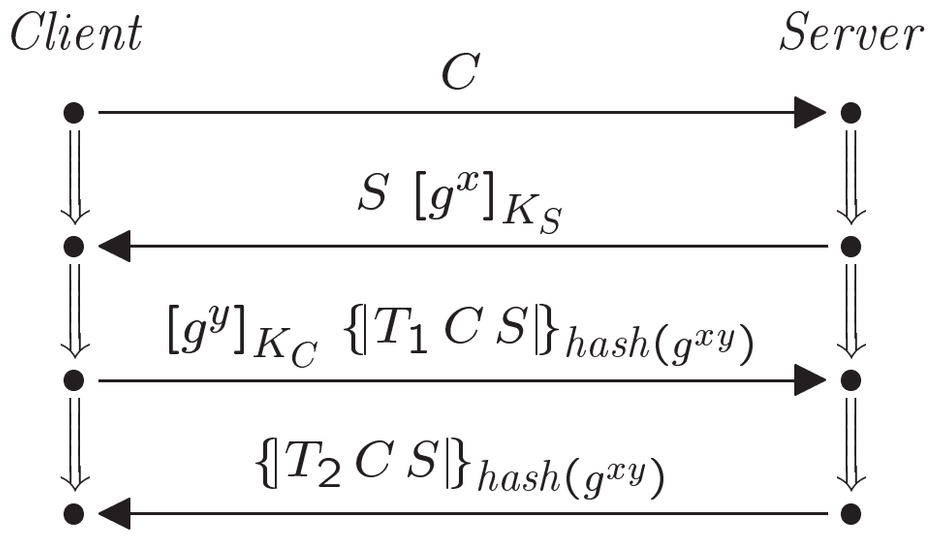
Incorporating Diffie-Hellman Into Strand Spaces

Jonathan Herzog
The MITRE Corporation

This work supported by the National Security Agency

The TLS-DH Protocol (Simplified)



- Three components:
  1. A handshake
  2. Authenticated Diffie-Hellman
     - $[M]_{K_X}$ is M signed with X's private key,
  3. Confirmation, using a hash of g^{xy} as the key

Incorporating Diffie-Hellman

- Standard Dolev-Yao model doesn't consider Diffie-Hellman
  - How to incorporate?
- One approach [PQ '01, BB '03, MS '03]:
  - Give adversary specific additional abilities
    - Multiplication, inverses, etc.
  - Prove secrets not deducible
- Good for finding flaws
- However, lack of flaws does not imply security
  - "Real" adversary may have additional powers
  - May be undiscovered attacks

Proving Security

- Focus on proofs of security rather than flaws
- Want proof method that captures all attacks
- Tempting to use "computational" model
  - Everything is an algorithm
  - Messages are bit-strings drawn from distributions
  - Proofs use reductions
    - "If an adversary can break the protocol, an adversary can break the underlying encryption"
  - Relies upon intractability assumptions
    - Actual form of Diffie-Hellman assumption

Present Work

- Computational proofs extremely meaningful
  - Grounded in complexity theory
- However, lacks benefits of Dolev-Yao model:
  - High level of abstraction
  - Simplicity
  - Automation
  - Re-use of general theorems
- This talk: best of both worlds
  - High-level security proofs for protocols like TLS
  - Existence of computational proofs guaranteed

General Approach

- Increase expressiveness of model
  - New operators
  - New adversary powers
- Assume Diffie-Hellman is hard
  - Translate Diffie-Hellman into formal terminology
  - (Introduce some strand space vocabulary)
- Analyze TLS
  - Assuming Diffie-Hellman to be hard
- Demonstrate translation accuracy
  - Show: if translation is false, Diffie-Hellman is easy

New Operators

- Randomized encryption
- Signatures, also randomized
- Hashing
  - Turns any message into a key
- Formal, free algebra abstraction of group operations
  - Atomic Diffie-Hellman elements: $d_a,\ d_b \in D$
    - Analogous to $g^x,\ g^y$
  - Formal Diffie-Hellman operation: $DH(d_a, d_b)$
    - Analogous to g^{xy}
    - Produces compound messages
  - Will reserve $g^x,\ g^y,\ g^{xy}$ for the computational setting

Extending the Adversary

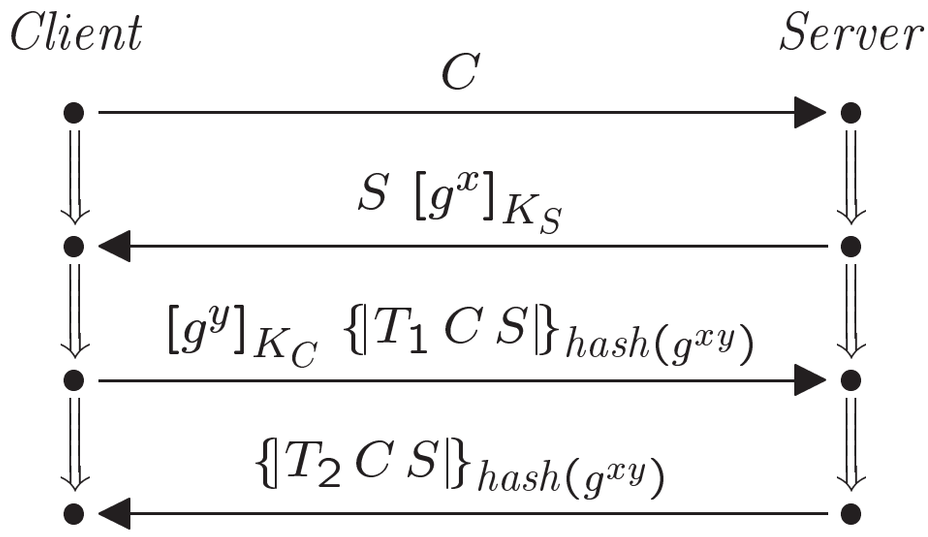
- What additional powers to give to adversary?
- Want to prove security against any efficient adversary
- Might as well give the adversary all reasonable powers
  - Adversary can perform every tractable function

$$f : A^* \rightarrow D$$

(A is any message)

- Other techniques free to consider smaller sets

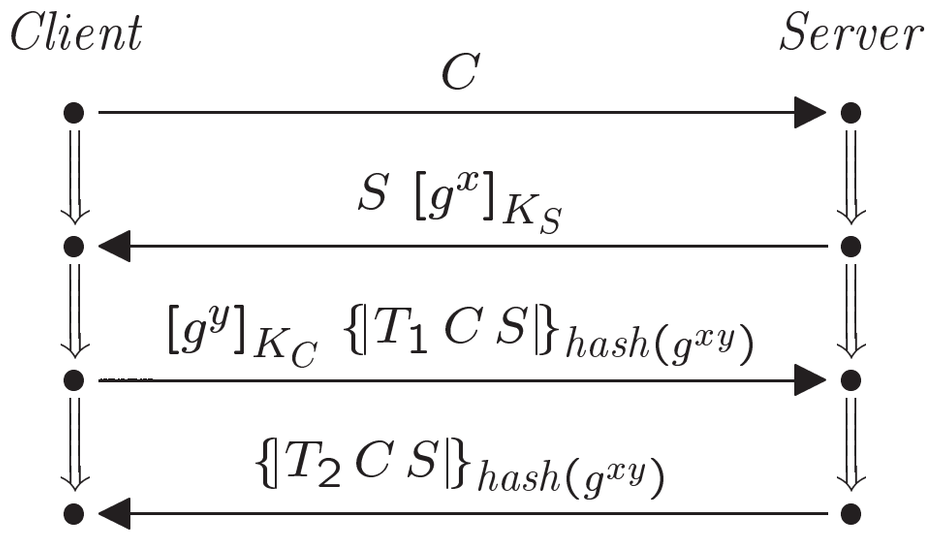
Strand Space Terminology



Regular Participant: One who follows the protocol

- As opposed to adversary

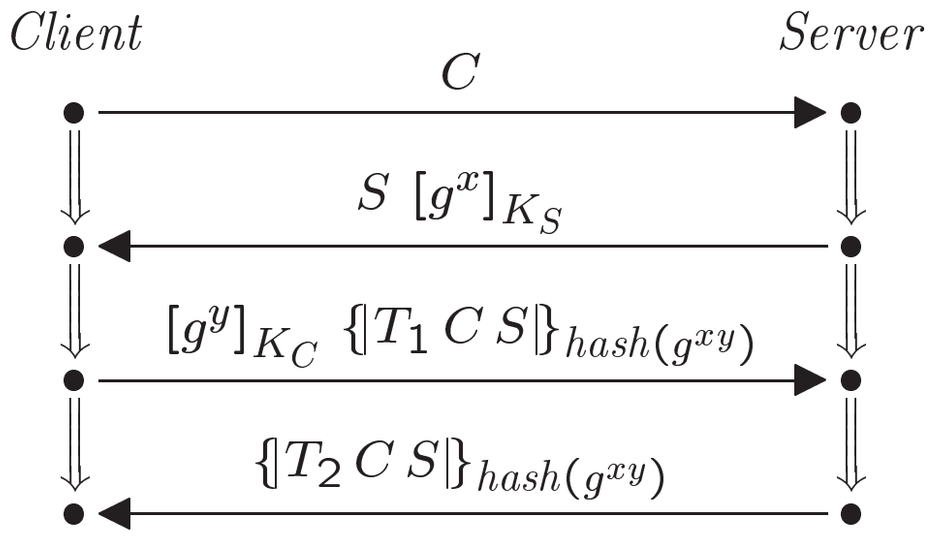
Strand Space Terminology



Strand: Sequence of messages sent, received

- Regular strand: trace of one particular execution
- Adversary strand: single operation
  - Link together to form more complex operations

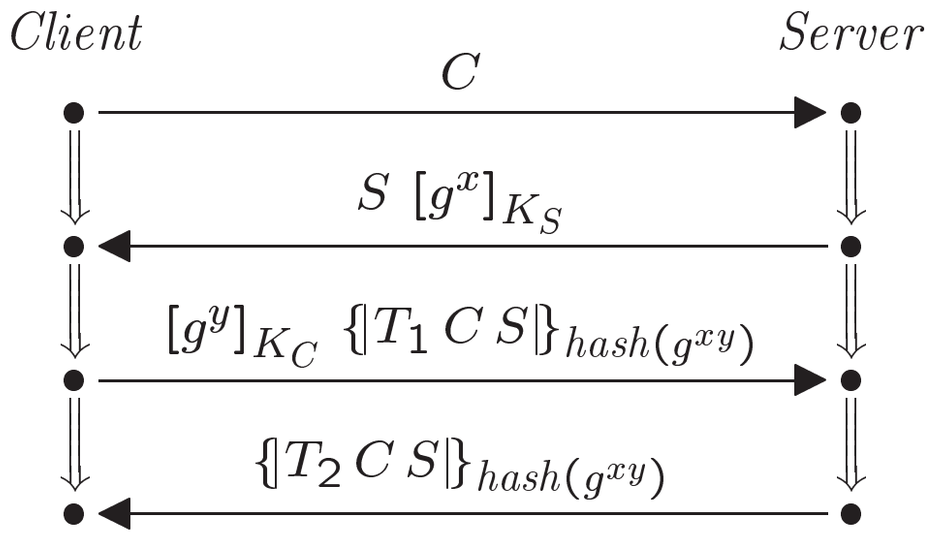
Strand Space Terminology



Bundle: Collection of communicating strands

- Who says what to who
- Global view of all conversations
- Could be different from intended conversation

Strand Space Terminology



Origination: Strand utters value it never heard

- "First" time value is used
- No origination $\rightarrow$ secret
- Note: value does not originate when used as key

Formal Diffie-Hellman Condition

If 1. g^x and g^y are created only by honest participants,

d_a and d_b originate only on regular strands

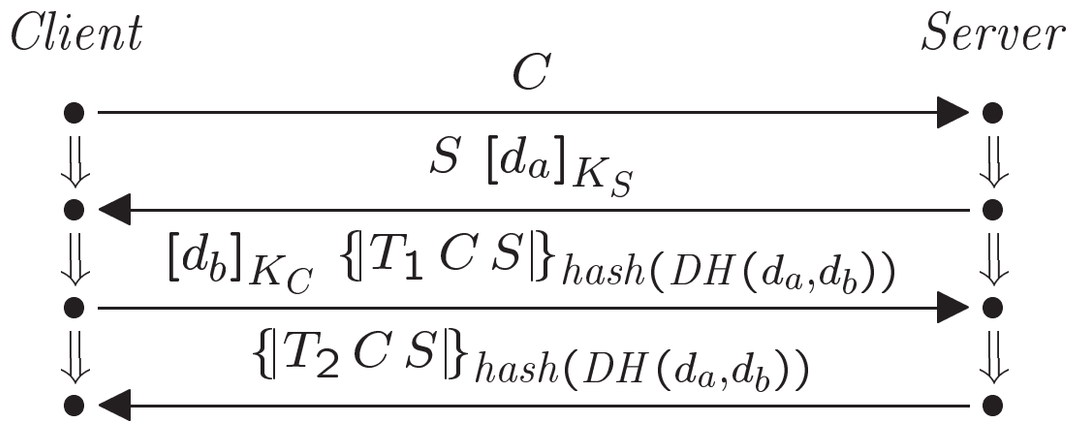
2. g^{xy} is not uttered by honest participants

$DH(d_a, d_b)$ does not originate on regular strands

Then g^{xy} is not emitted by adversary either

$DH(d_a, d_b)$ does not originate at all

Proof Sketch of Security



- Assumption: d_a, d_b originate only on regular strands
- No value $DH(d_1, d_2)$ originates on regular nodes
- Therefore $DH(d_a, d_b)$ does not originate:
  - secrecy
- Thus $hash\,(DH(d_a, d_b))$ does not originate
- Encrypted with secret key $\rightarrow$ emitted by regular strands:
  - authentication

Deriving the Diffie-Hellman Condition

- How to justify such a condition?
  - Does it diminish the computational soundness of the model?
- Derivation:

  In particular,

  1. Give computational semantics to Strand Spaces

  2. Then show:

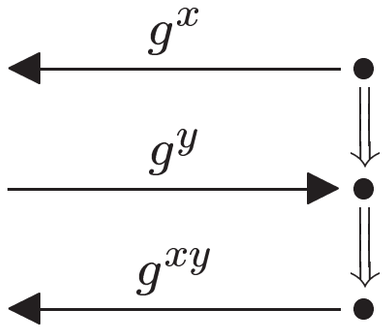
     "If a bundle violates the formal Diffie-Hellman condition, it maps to an efficient algorithm that solves Diffie-Hellman"

Derivation Sketch

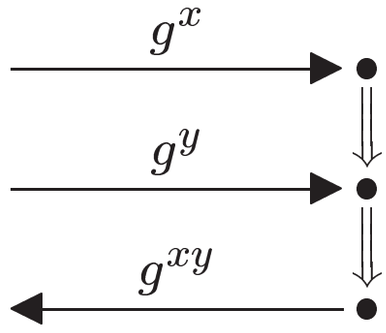
- Give bit-string value to every message in bundle
- Every atomic term represents random variable
  - Atomic terms given random value
  - Compound terms built up from atomic ones
- Adversary strands all tractable functions
- Regular strands may not be
  - Regular participants unconstrained
  - Might represent intractable computations

Tractable and Intractable Regular Strands

Tractable regular strand



Intractable regular strand



Tractable Regular Strands

- Want to avoid intractable strands
- Details highly strand-specific
  - Also specialized for TLS
- General idea:
  - Invoke traits of protocols like TLS
    - Real participants know secret exponents
    - Don't utter secret values, but hash into keys
  - Together, make regular strands tractable
- Details in paper

Deriving Formal Diffie-Hellman condition

- Suppose some bundle violates security property
  - d_a, d_b originate on regular nodes
  - $DH(d_a, d_b)$ originates only on adversary node
  - Regular strands tractable
- Turn it into algorithm
  - g^x, g^y given as inputs, assign to d_a, d_b
  - Choose values for all other atomic messages
  - Each strand easy to compute
  - Compose individual computations according to bundle structure
  - Node for $DH(d_a, d_b)$ now has value for g^{xy}.
  - Case analysis: g^{xy} appears unencrypted

Conclusion

- Diffie-Hellman incorporated into Strand Spaces
  - Does not diminish computational soundness
- Probably can be used by automated Strand Space tools
- Areas for generalization:
  - "Common protocol traits" based on TLS
    - Group key protocols likely have other traits
  - Approach possibly applicable to other formalisms
  - Also probably applicable to other primitives

Backup slides

Randomized Encryption

- Encryption explicitly takes randomness as argument:

$$Enc : A \times Key \times Rand \rightarrow A$$

$$Enc(M, K, r) = \{|M|\}_K^r$$

- Signatures similar

Common Protocol Traits

- Real protocol participants don't solve Diffie-Hellman problem
  - Won't calculate g^{xy} unless they know x or y
  - Presumably, regular participants choose g^x by first picking x.
  - Def: regular strands are *conservative* if they never use $DH(d_1, d_2)$ unless d_1 or d_2 originates on regular node
- Also, honest participants don't commonly "say" g^{xy}:
  - Def: regular strands are *silent* if no $DH(d_1, d_2)$ originates on regular strands
  - Still allows regular strands to use $DH(d_1, d_2)$ as a key
  - All such keys are produced by hashing

Side-stepping Diffie-Hellman

- If hashing is strong, a hash of $DH(d_1, d_2)$ has same distribution as random value
- Hence, no need to calculate pre-image to hashes
  - Pick random values instead
  - If this changes anything, then hashing is not strong
    - Proof uses conservatism of regular strands
- No longer need to solve Diffie-Hellman to calculate regular strands
- All strands efficiently computable

Computational Soundness of Dolev-Yao

- Work in progress
- Backes, Pfitzmann, Waidner
  - Universally Composable Cryptographic Library (2003)
- Lincoln, Mitchell, Mitchell, Scedrov
  - Incorporating poly-time indistinguishability into process calculi
- More direct approaches
  - Abadi and Rogaway
  - Bogdan
  - Myself
- Probably will be settled in next five years

Comparison with Millen, Shmatikov

Them	Me
Finds flaws	Produces proofs
May not find all flaws	May not produce proof for all correct protocols
Untyped	Typed
Limited adversary powers w.r.t. Diffie-Hellman	Unlimited adversary powers w.r.t. Diffie-Hellman
Decision procedure	Pretty sure result can be incorporated into other tools