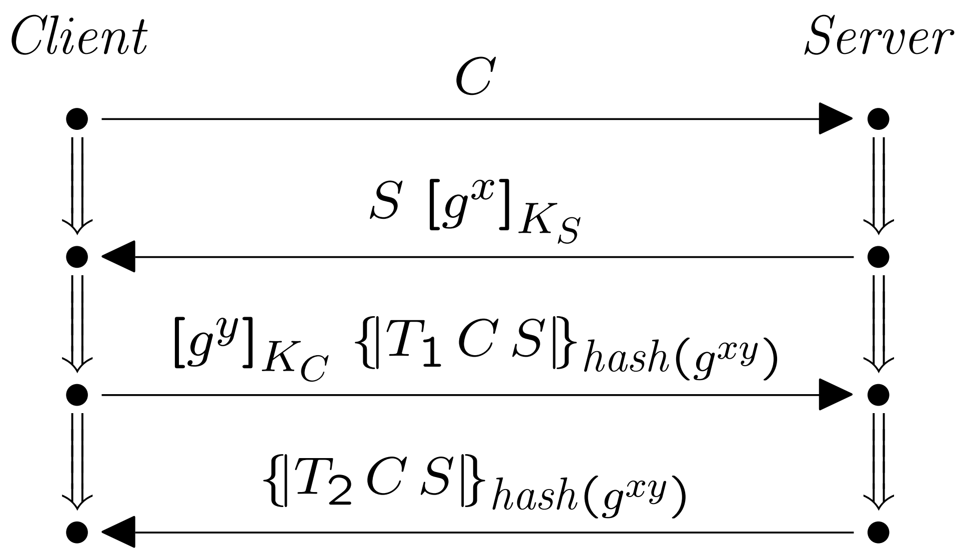
Incorporating Diffie-Hellman Into Strand Spaces

Jonathan Herzog
The MITRE **Corporation**

This work supported by the National Security Agenc

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 ke

Incorporating Diffie-Hellman

- Standard Dolev-Yao model doesn't consider Diff
  - 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 powe
  - 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 distrib
- Proofs use reductions
  - "If an adversary can break the protocol, adversary can break the underlying encry
- Relies upon intractability assumptions
  - Actual form of Diffie-Hellman assumptio

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

– Existence of computational proofs guarante

General Approach

Increase expressiveness of model

– New operators

– New adversary powers

Assume Diffie-Hellman is hard

– Translate Diffie-Hellman into formal termi

– (Introduce some strand space vocabulary)

Analyze TLS

– Assuming Diffie-Hellman to be hard

Demonstrate translation accuracy

– Show: if translation is false, Diffie-Hellman

New Operators

Randomized encryption

Signatures, also randomized

Hashing

- Turns any message into a key

Formal, free algebra abstraction of group opera

- Atomic Diffie-Hellman elements: d_a, $d_b \in$
  - Analogous to g^x, g^y
- Formal Diffie-Hellman operation: $DH(d_a,$
  - Analogous to g^{xy}
  - Produces compound messages
- Will reserve g^x, g^y, g^{xy} for the computational

Extending the Adversary

What additional powers to give to adversary?

Want to prove security against any efficient adv

Might as well give the adversary all reasonable

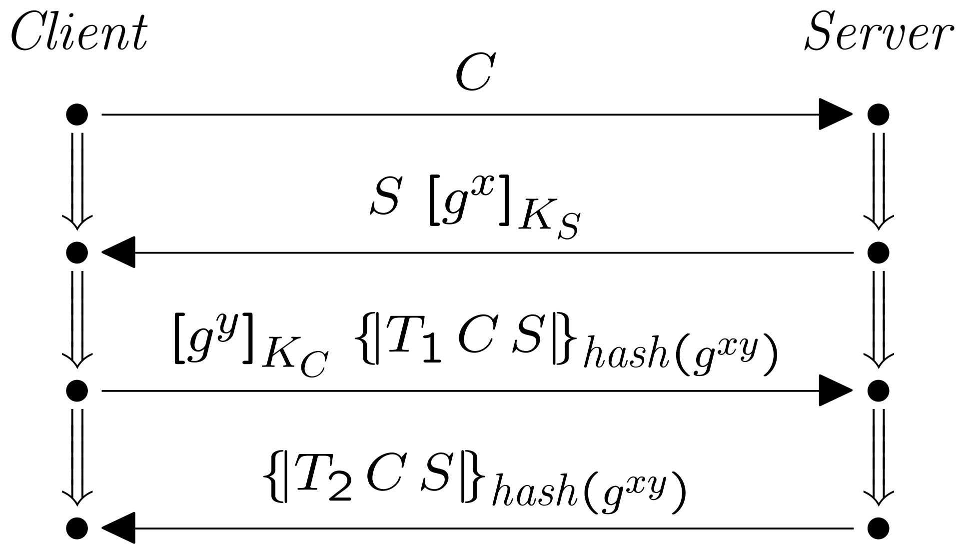
– Adversary can perform every tractable func

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

(A is any message)

Other techniques free to consider smaller sets

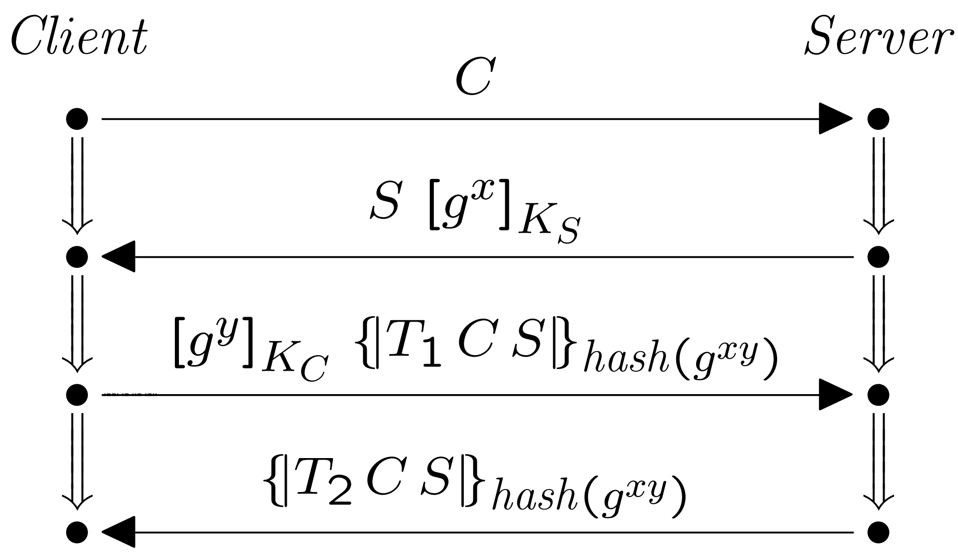
Strand Space Terminology



Regular Participant: One who follows the proto

As opposed to adversary

Strand Space Terminology



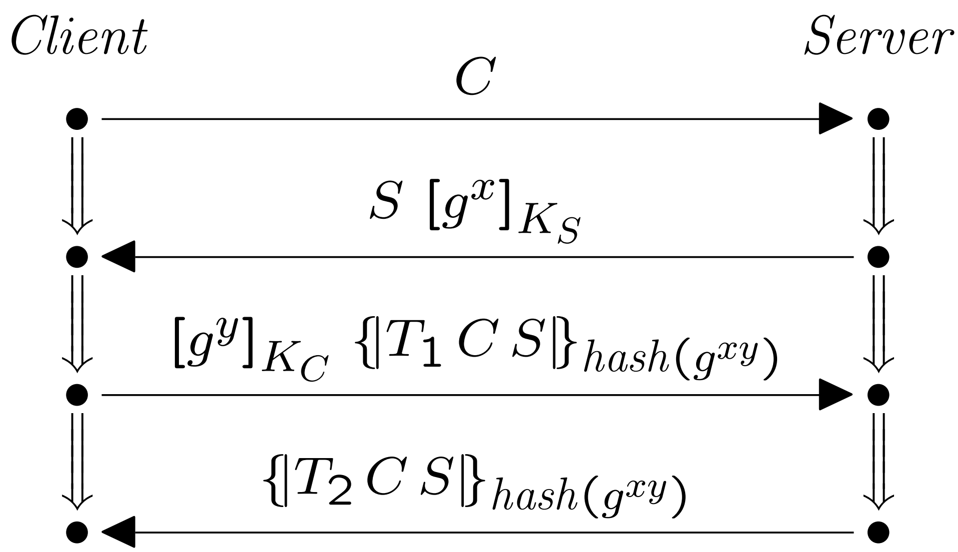
Strand: Sequence of messages sent, received

Regular strand: trace of one particular exec

Adversary strand: single operation

– Link together to form more complex ope

Strand Space Terminology



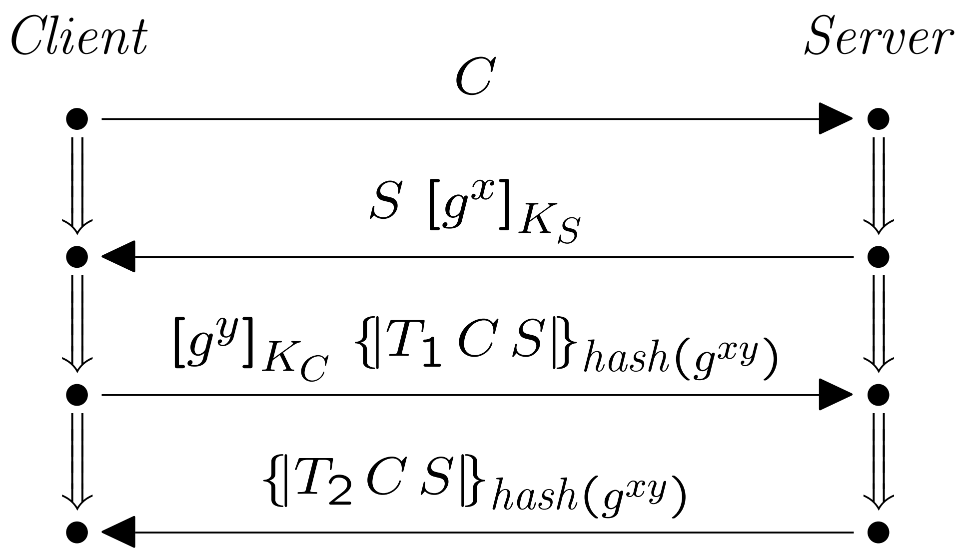
Bundle: Collection of communicating strands

Who says what to who

Global view of all conversations

Could be different from intended conversati

Strand Space Terminology



Origination: Strand utters value it never heard

“First” time value is used

No origination → secret

Note: value does not originate when used a

Formal Diffie-Hellman Condition

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

d_a and d_b originate only on regular strands

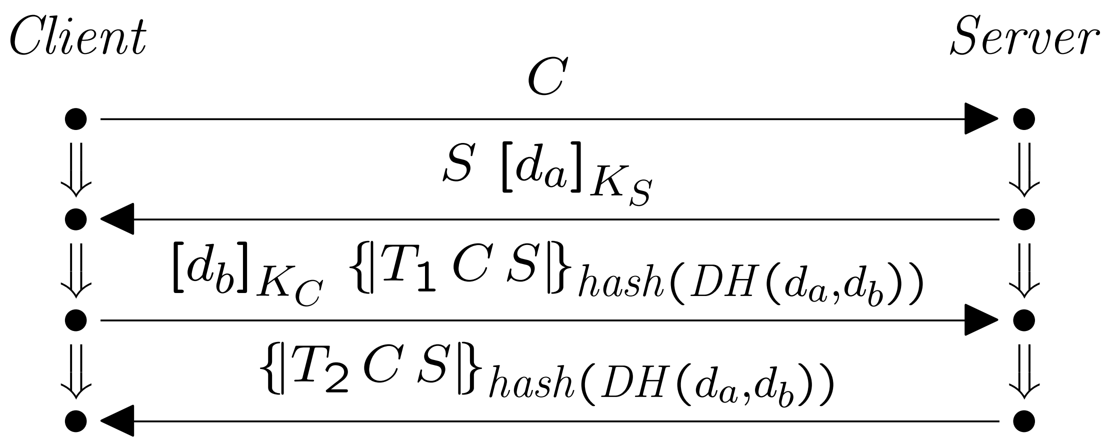
2. g^{xy} is not uttered by honest participants

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

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 str

No value $DH(d_1, d_2)$ originates on regular node

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 s

– authentication

Deriving the Diffie-Hellman Condition

How to justify such a condition?

– Does it diminish the computational sound
the model?

Derivation:

1. Give computational semantics to Strand Spa

2. Then show:

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

Derivation Sketch

Give bit-string value to every message in bundl

Every atomic term represents random variable

- Atomic terms given random value
- Compound terms built up from atomic ones

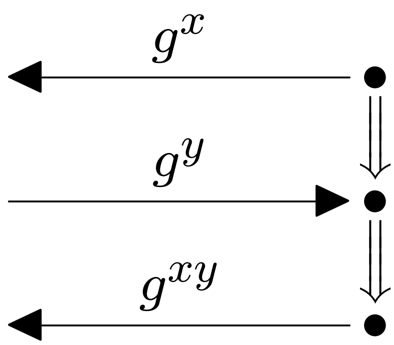
Adversary strands all tractable functions

Regular strands may not be

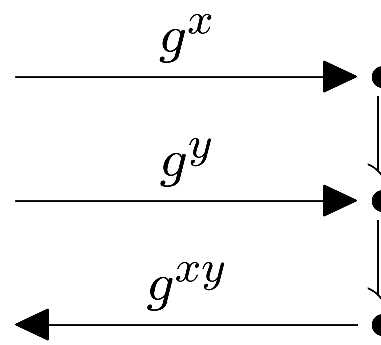
- Regular participants unconstrained
- Might represent intractable computations

Tractable and Intractable Regular Stran

Tractable regular strand



Intractable regular s



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
- Together, make regular strands tractable

Details in paper

Deriving Formal Diffie-Hellman conditi

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 message
- Each strand easy to compute
- Compose individual computations accordin 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 Spa

Areas for generalization:

– "Common protocol traits" based on TLS

  ◦ Group key protocols likely have other tra

– Approach possibly applicable to other forma

– Also probably applicable to other primitives

Backup slides

Randomized Encryption

Encryption explicitly takes randomness as argu

$$Enc : \mathsf{A} \times Key \times Rand \to \mathsf{A}$$

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

Signatures similar

Common Protocol Traits

Real protocol participants don't solve Diffie-H
problem

- Won't calculate g^{xy} unless they know x or y
- Presumably, regular participants choose g^x
  picking x.
- Def: regular strands are *conservative* if the
  use $DH(d_1, d_2)$ unless d_1 or d_2 originates on
  node

Also, honest participants don't commonly "say"

- Def: regular strands are *silent* if no $DH(d_1, d$
  inates on regular strands
- Still allows regular strands to use $DH(d_1, d$
  key
- All such keys are produced by hashing

Side-stepping Diffie-Hellman

If hashing is strong, a hash of $DH(d_1, d_2)$ ha
distribution as random value

Hence, no need to calculate pre-image to hashe

– Pick random values instead

– If this changes anything, then hashing is not

∘ Proof uses conservatism of regular stran

No longer need to solve Diffie-Hellman to calcu
regular strands

All strands efficiently computable

Computational Soundness of Dolev-Yao

Work in progress

Backes, Pfitzmann, Waidner

– Universally Composable Cryptographic Libra

Lincoln, Mitchell, Mitchell, Scedrov

– Incorporating poly-time indistinguishability i
cess 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 pr all correct protocols
Untyped	Typed
Limited adversary powers w.r.t. Diffie-Hellman	Unlimited adversary ers w.r.t. Diffie-Hel
Decision procedure	Pretty sure resul be incorporated into tools