Secure Tunnel Establishment

Protocol

for the Internet

Amir Herzberg
Juan Garay
Hugo Krawczyk

IBM - T.J. Watson Research Center

Secure Tunnel Establishment

- key negotiation protocol between two IP nodes
- goals:
  - share a key
  - authenticate parties
  - authenticated connection/session establishment
  - parameters negotiation (algorithms, key size, etc.)
- shared key:
  - used for authentication, encryption, etc. (via a network-layer protocol, e.g. swIPe)
  - actually, via derived session keys
  - master vs. session key (even session sub-keys)
    efficiency
    key exposure
    simultaneous sessions
  - cacheing of master shared key

Approaches to Key Sharing

- manual installation:
  - some form of physical/human trust (e.g., phone, paper, agents)
- key distribution center:
  - mutually trusted third party (server)
  - protocol for distribution (Kerberos, KryptoKnight)
  - efficient solution, but: trust, on-line server
- public key:
  - requires trusted integrity of public keys (not secrecy!)
  - distribution: from sneaker-net to full certification
  - additional mechanism to derive a shared key
  - "universal" (weak trust) but relatively expensive (computation, key size, certification, patents)

Our Approach to Tunnel Establishment

- support different master key sharing methods (esp. PK, KDC)
- hierarchical solution
  - share a master key (refresh periodically)
  - cache (optional)
  - derive session keys

$$\text{Master-Key} \xrightarrow{\text{(cache)}} \text{Session-Key(s)}$$

- performance/efficiency:
  - minimal interaction
  - minimal computation overhead (esp. PK)
  - single/compact packet format
  - support Internet heterogeneous environment
- security:
  - hierarchy and key refreshment
  - PK: effect of private key exposure
  - sound cryptographic design

Specific Proposal

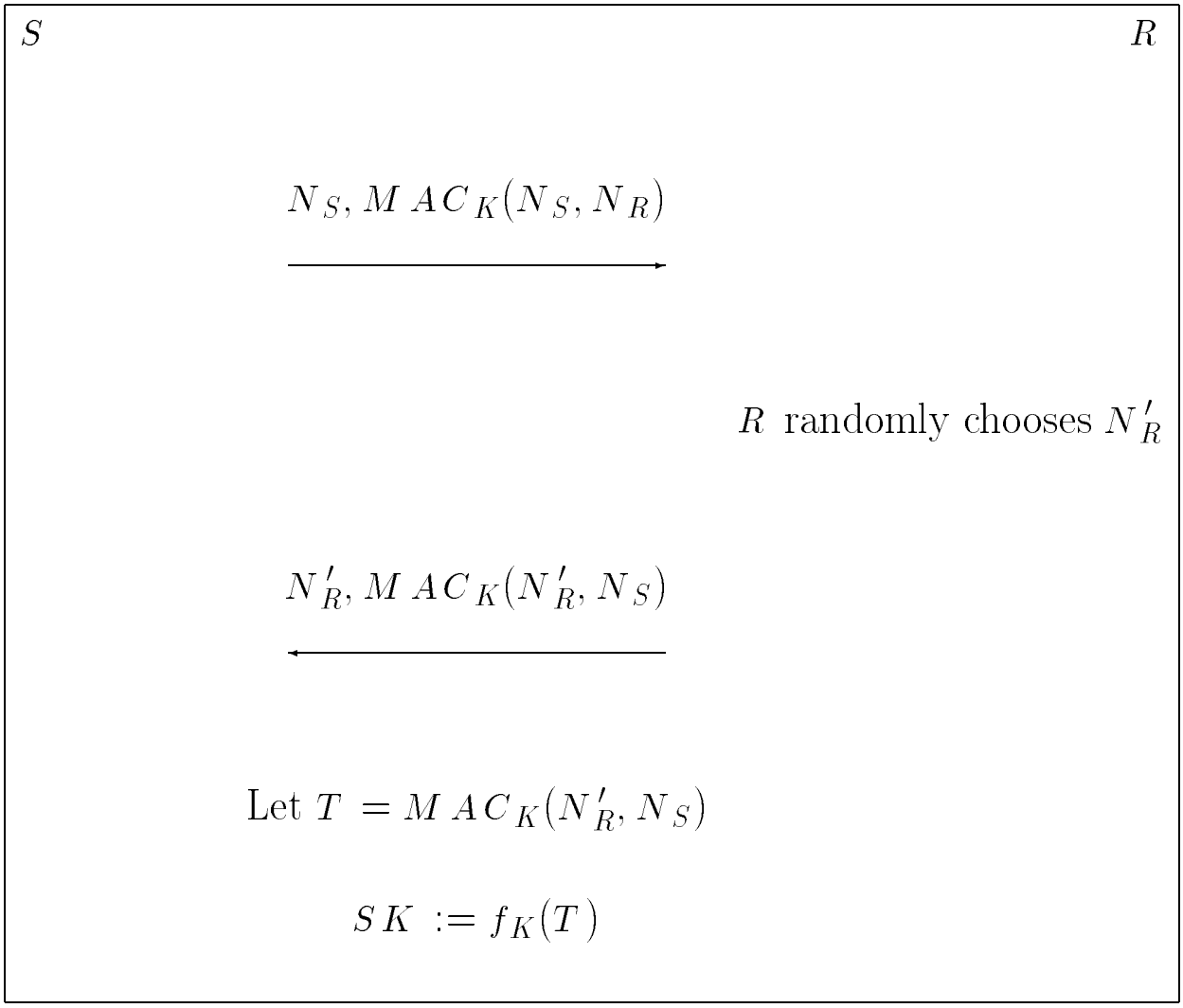
We present protocols for:

- exchanging a master key using public key and deriving session key
- authenticating and deriving session keys from (cached) master key without use of public key

Properties:

  - protocols combined into one single format
  - minimal interaction: two messages only
  - public key to a minimum (cache!)
  - minimize number of (long) exponentiations
  - extendible to the key distribution center scenario (to be included in our proposal)
  - supports simultaneous sessions

Session Establishment Protocol



Assumes:	shared master key K
	shared nonce N_R (from previous communication)
MAC_K:	Msg Auth Code using key K (e.g. key-ed MD5)
f_K:	pseudorandom function

Observations

1. Only basic information shown: more information to be authenticated (e.g., IP address).

2. The session key SK is not explicitly transmitted.
   Thus, no need to explicitly encrypt/authenticate it.

3. Can similarly derive several session keys.
   Different keys may be needed, e.g., for confidentiality and for data integrity verification.

4. Use of shared nonces vs. time:
   - nonces used against replay
   - nonce approach is usually more secure since secure time synchronization is hard to achieve
   - if shared nonce N_R is lost can resynchronize using extra flow or time (in the $S \rightarrow R$ flow)
   - effect of replay by S is limited: can pass authentication but not learn/replay session key

Master Key Exchange Protocol

(Based on public key)

Two parties S and R.

- posses the authentic public key of each other
- share a nonce

E_X RSA encryption with X's public key.
Information is padded with a random string prior to encryption.

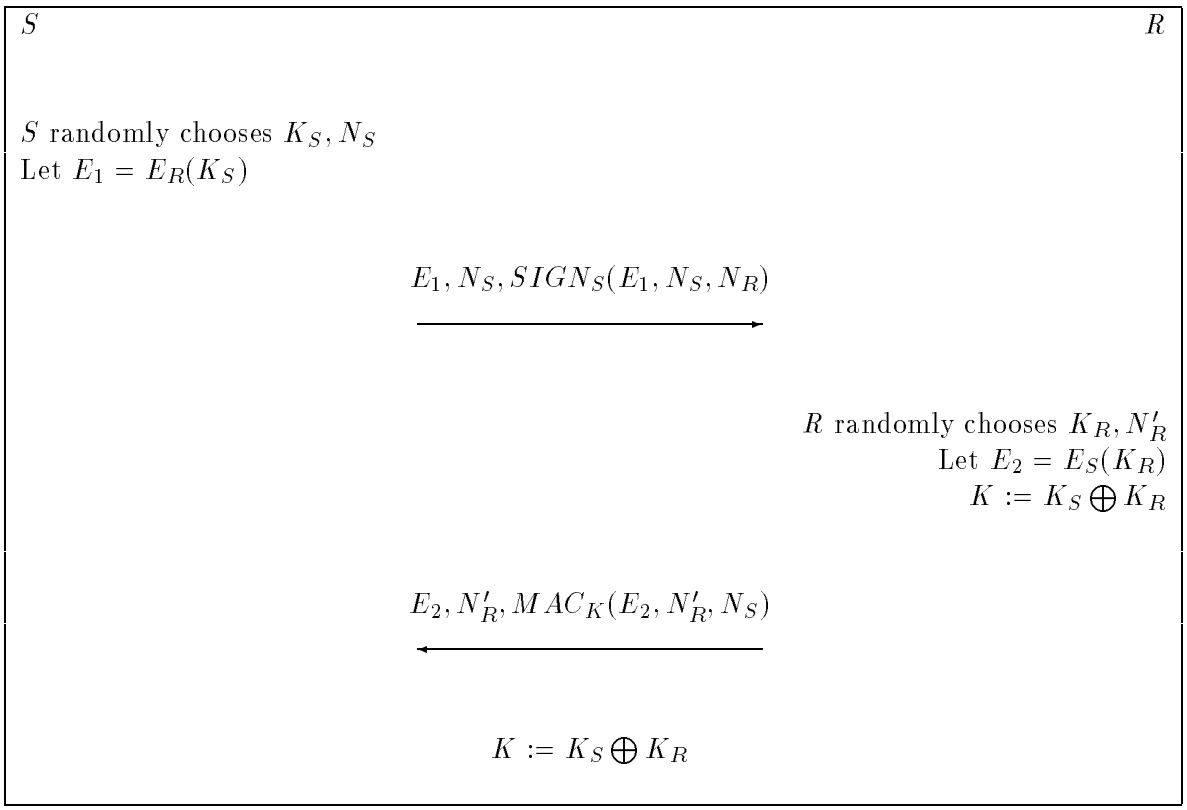
$SIGN_X$ X's RSA signature.
First hash the signed information, and then apply RSA (i.e., exponentiation with X's private key).

N_X A nonce (i.e., a random number) chosen by X.

K_K Random string chosen by party X.

K The shared master key, outcome of the protocol.

Master Key Exchange Protocol



Observations

1. Robust against compromise of one party's public key
   (Diffie-Hellman does better but computationally expensive)

2. Efficiency: 2 flows, and 3 long exponentiations.

3. R first verifies the signature and, only if this succeeds it, decrypts K_R.
   This reduces computational overhead, e.g., against clogging attack.

4. Nonce issue: as before (extra flow/time, limited effect of replay)

5. Derivation of session key: $SK := f_K(T')$,
   for $T' = MAC_K(E_2, N'_R, N_S)$.

6. Same packet format for both protocols
   (i.e., master-key exchange and session establishment)

Proposal to WG

- WG draft
- Scope:
  - as above
  - KDC support
  - details
- suggestions, decisions