SIGMA: SIGN-and-MAC

Crypto rationale and proposals

Presentation by Sara Bitan and Hugo Krawczyk

sarab@cs.technion.ac.il hugo@ee.technion.ac.il

IETF meeting, December 2001

Agenda

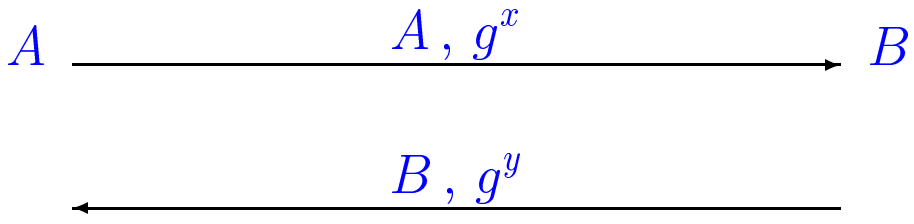
- Crypto background: key exchange
- The SIGMA approach
- Specific proposals
- Comparison
- MAC, encryption and ESP issues

Crypto Focus

- Focus on cryptographic design:
  - security: secrecy and authentication (and the subtleties of identity-key binding)
  - sound analysis(proponent's responsibility)
  - PFS: full, windowed
  - identity protection: who, active vs. passive
  - performance: computation, latency
  - DoS protection: adaptive, built-in
- A lot of other issues are essential for a working protocol but **orthogonal** to the above:
  - message formats
  - general mechanisms (e.g., retransmissions)
  - extent of negotiation
  - code preservation vs. "start from scratch"

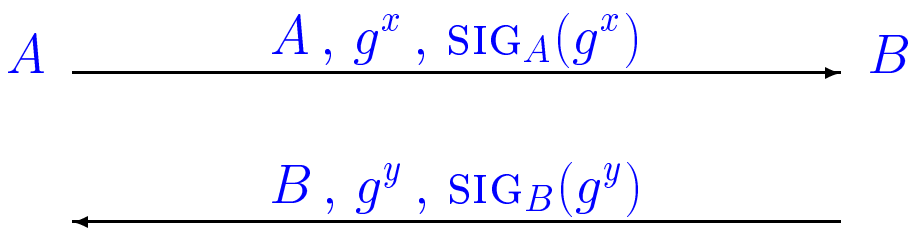
Building Authenticated Diffie-Hellman

The basic:



- assumes authenticated channels
- what if man-in-the-middle?

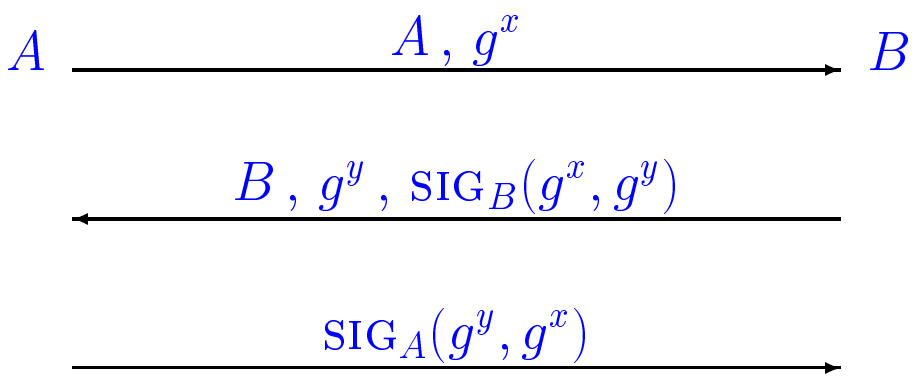
Attempt at Authenticated Diffie-Hellman



- what if attacker ever finds a triple $(x, g^x, \text{SIG}_A(g^x))$?
  - e.g., file of pre-computed (x, g^x) pairs
- ephemeral leakage should never allow long-term impersonation

Authenticated DH (with replay protection)

Note: nonces/cookies omitted (needed if g^x, g^y re-used)

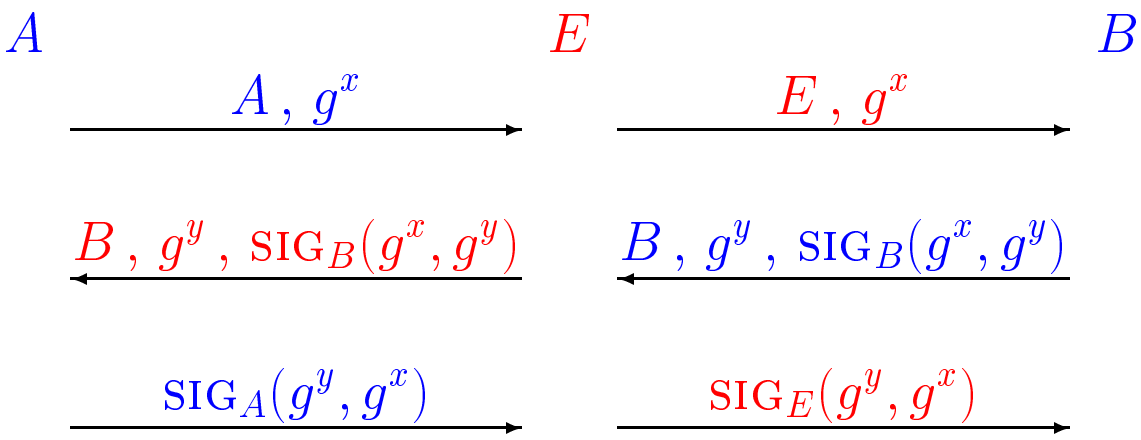


A: "Shared $K = g^{xy}$ with B" $(K \equiv B)$

B: "Shared $K = g^{xy}$ with A" $(K \equiv A)$

Looks fine, but...

DVW attack [DVW]



- any damage? wrong identity binding!

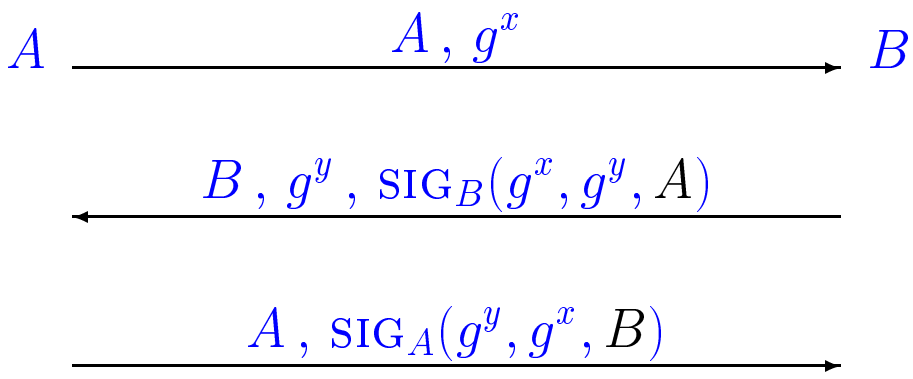
  A: "Shared $K = g^{xy}$ with B" ($K \equiv B$)

  B: "Shared $K = g^{xy}$ with E" ($K \equiv E$)

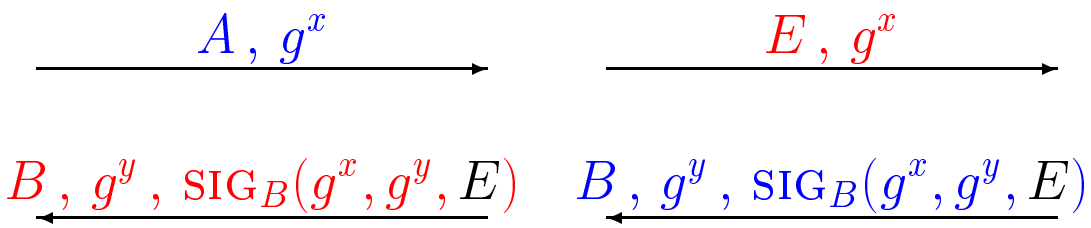
  E: doesn't know K but B will consider anything sent by A as coming from E

  $\{$*"deposit attached e-cash to my account"*$\}_K$

Authenticated DH (ISO)



Thwarts DVW attack:



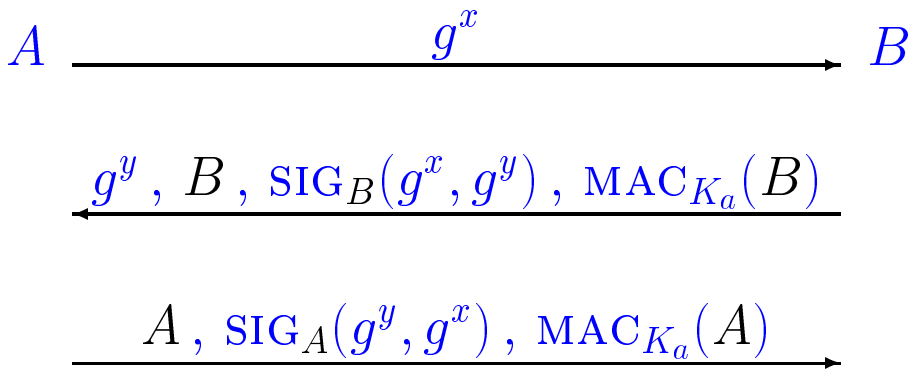
But is it secure? Yes: [CK - Eurocrypt'01]

Identity Protection: from ISO to SIGMA

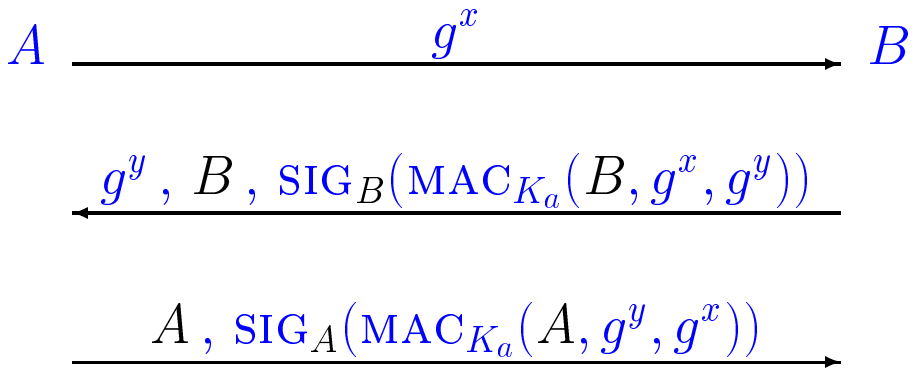
- ISO protocol: requires peer's id under signature
  - can only protect id's against *passive* attacks
  - active protection possible for one peer at the expense of extra signature and identity disclosure (or added round trips). E.g. JFK.
- Solution: do not bind peer's identity to sig
  - STS protocol (but attacks are possible)
  - two other variants (mac-ed signature and signed-key) are insecure
- Provable secure and efficient:
  SIGN-and-MAC (SIGMA)

  [Kra'95]: proposed to Photuris, adopted in IKE

SIGMA: the basic protocol



Equivalent security (just MAC space saving):



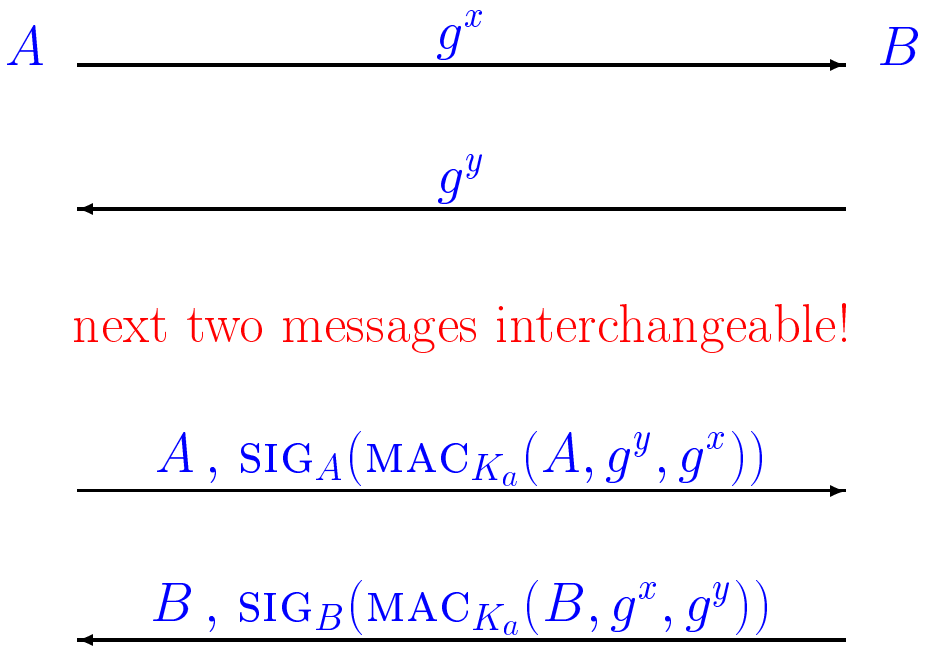
K_a derived from g^{xy}; can encrypt with K_e

Note: MAC $\equiv$ `prf` in IKE

SIGMA: Basic Design Facts

- **The essential step:**
  MAC your own IDentity!
- If ID not inside MAC security is totally compromised (even if ID included in signature!)
- Signature and MAC have complementary and essential security functionalities against M-i-t-M
  - signature protects secrecy of key against exponent replacement by MitM
  - MAC protects identity-key binding against DVW-type attacks by MitM
- ID protection via encryption (resistant to active attacks); but core authentication security decoupled from ID protection!
- Flexibility: a lot of possible design trade-offs (see next)

SIGMA: secure and flexible



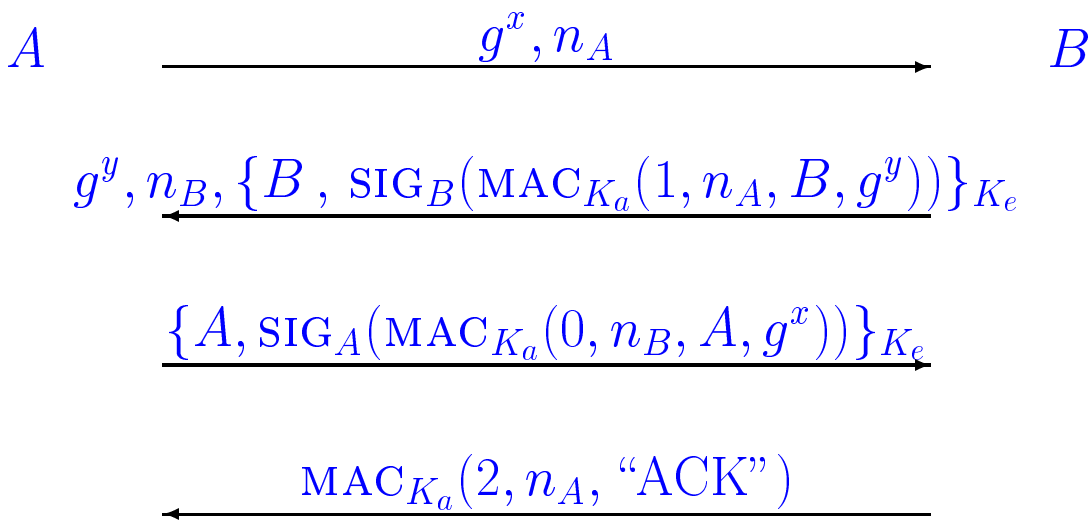
- interchangeability $\Rightarrow$ design tradeoffs!
  - id protection (active, passive)
  - round trips, computation latency
  - DoS protection (adaptive or built-in)

Properties of all SIGMA proposals

- Provable secure
- Full PFS (but allow reuse of DH exponents)
- One identity secure against active attackers, one against passive (best possible)
- Best performance for PFS (1 sig, 1 ver, 1 DH)
- Two round trips for core protocol:
  - SIGMA-4 includes DoS protection (in 2 RT)
  - SIGMA-I and SIGMA-R require optional round trip for adaptive DoS protection
- Note: following descriptions place MAC inside signature; MAC outside is equally good IF it explicitly covers identity!

Specific SIGMA proposals: SIGMA-I

(SIGMA instantiation in draft-sigma; added ack)



- 2 RTs in normal operation
- extra RT if DoS protection activated
- I's id protected against active attacks, R's id against passive

SIGMA-I: IKE-like notation

```
HDR, SA, KE, Ni    -->
                                HDR, SA, KE, Nr,
                   <--          IDir*, [CERT*,] SIG_R*
HDR, IDii*,
 [CERT*,] SIG_I*   -->
                   <--          HDR, "ACK", HASH-ACK
```

Notation:

*: encryption against active attacks

SIG_I = signature of I on HASH_I

SIG_R = signature of R on HASH_R

HASH_I = prf(SKEYID, 0 | Nr | IDii_b | MSG_I)

HASH_R = prf(SKEYID, 1 | Ni | IDir_b | MSG_R)

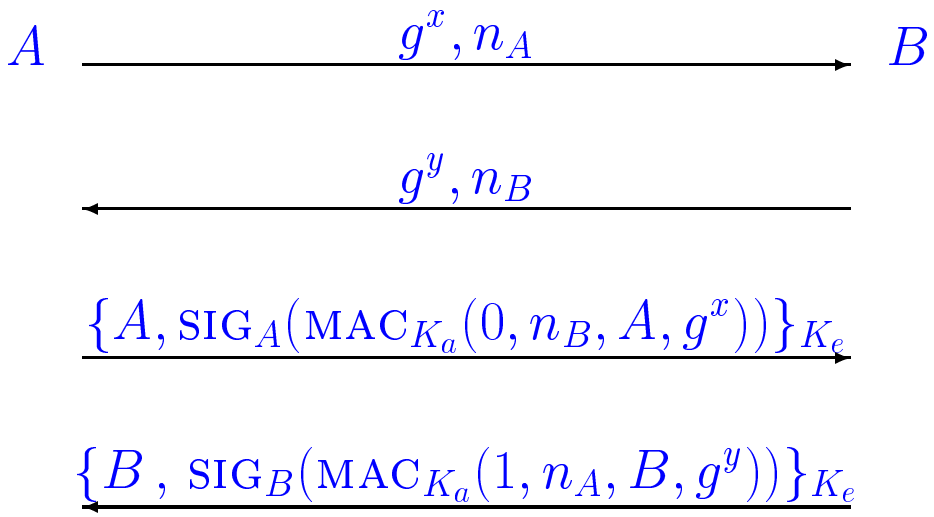
MSG_I = all information sent by I (except SIG)

MSG_R = all information sent by R (except SIG)

HASH-ACK = prf(SKEYID, 2 | Nr | HDR | "ACK")

Specific SIGMA proposals: SIGMA-R

(IKEv2-like but explicit MAC and provable security)



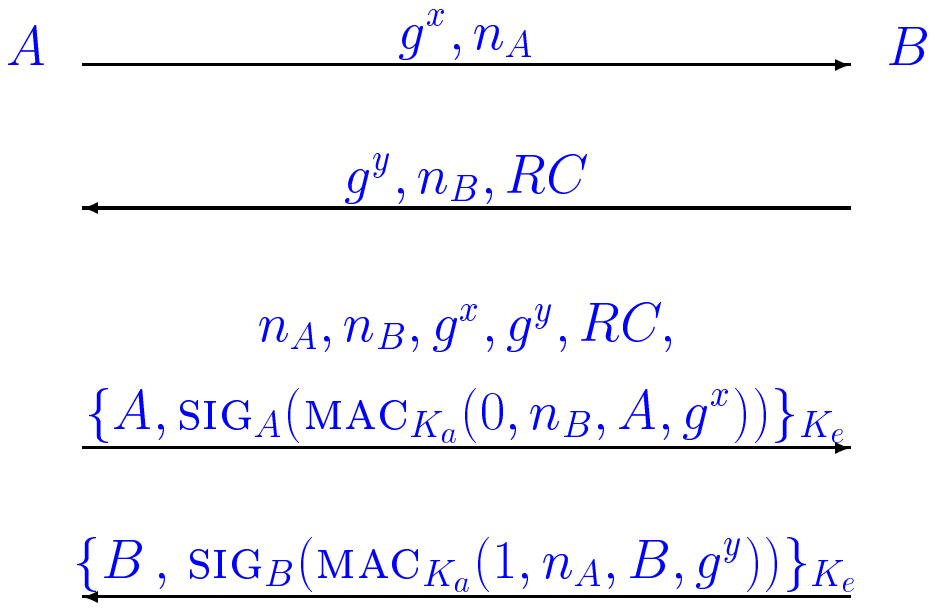
- 2 RTs in normal operation
- extra RT if DoS protection activated
- R's id protected against active attacks, I's id against passive

SIGMA-R: IKE-like notation

```
HDR, SA, KE, Ni  -->
                 <-- HDR, SA, KE, Nr
HDR, IDii*,
 [CERT*,] SIG_I* -->
                 <-- HDR, IDir*, [CERT*,] SIG_R*
```

Specific SIGMA proposals: SIGMA-4

(A sig-based version of P-SIGMA in draft-sigma: a "resolution" of SIGMA, IKEv2 and JFK)



- 2 RTs, including DoS protection via a cookie RC computed on n_B, n_A, g^y
- R's id protected against active attacks, I's id against passive
- long msg3, long input to cookie

SIGMA-4: JFK/IKE notation

JFK-like notation:

```
I->R: Ni, g^i
R->I: Ni, Nr, g^r, GRPINFOr, RC
I->R: Ni, Nr, g^i, g^r, RC
      E{Ke}(IDi, sa,  SIG{i}(MAC{Ka}(0,info-I)))
R->I: E{Ke}(IDr, sa', SIG{r}(MAC{Ka}(1,info-R)))

RC=Cookie-function(Ni,Nr,g^r)
info-I = Nr, Ni, IDi, g^i, g^r, sa
info-R = Ni, Nr, IDr, g^r, g^i, sa'
```

IKE-like notation:

```
HDR, KEi, SAi, Ni       -->
                        <-- HDR, KEr, SAr, Nr, RC
HDR, RC, KEr, SAr, Nr,
KEi, Ni, IDii*, SIG_I* -->
                        <--    HDR, IDir*, SIG_R*

With RC=Cookie-function(Ni, Nr, KEr, SAr)
Traffic SA and [CERT*,] payloads omitted
```

Comparison

Measures:
Security/analysis
DoS: adaptive, built in, cookie gen/ver cost
Id prot: I/R active/passive, transferable proof
Performance (computation)
Round trips

	SIGMA-I	SIGMA-R (IKEv2')	SIGMA-4	JFK
Sec	proof	proof	proof	proof [1]
IDi	active	passive	passive	active[2]
IDr	passive	active	active	none
DoS	adaptive	adaptive	built-in	built-in
Perf	min-PFS shrt-cky	min-PFS shrt-cky	min-PFS long-cky	+1 sig/ver long-cky
RTs	2(3)	2(3)	2	2

[1] but high cost: decreased pfs, weak privacy
(R's id revealed + proof of comm), performance-
[2] lost if (r,SIG(g^r)) ever exposed

Dual use of MAC

- Two functionalities for MAC
  1. core authentication security of the protocol (identity-key binding) – see Slide 9
  2. identity protection against active attackers (requires integrity mechanism on top of encryption)
- Cleaner and robust: separate the two MAC's:
  - basic principle: keep core authentication independent of id protection
  - example: what if ID not included under encryption, or under a MAC-ed message?
  - use ESP for id protection (save re-specifying ENC modes and algorithms)
  - separation also allows for non-MAC-based ESP specs [Jut01]
- What is the cost of separation? A one-block SHA-1 computation!

A summary of MAC options

1. `ENC{Ke}(...,ID,SIG,...), MAC{Ka}(ciphertext)`
   secure ONLY if ID is under ciphertext
2. `ENC{Ke}(...,SIG,...), MAC{Ka}(ID,ciphertext)`
   explicit inclusion of essential ID under MAC;
   does not depend on ID position in the protocol
   (e.g. if sent in the clear in the first message)
3. Two MACs (with clearly differentiated goals):
   - one for essential protocol security: `MAC{Ka}(ID)`
     (or included under SIG as in current IKE:
     `SIG(MAC{Ka}(ID, other-signed-info))`
   - another for ciphertext protection only:
     `MAC{Ka}(ciphertext)` (as in item 1)
     (allows use of any confidentiality+integrity
     protecting ESP transform)

   Cost of additional MAC: a one-block SHA-1
   computation

"Exercise": rationale for IKEv2

Stage 1: exchange DH and SA negotiation

```
HDR, SA, KE, Ni    -->
                   <--    HDR, SA, KE, Nr
```

Stage 2: authenticate DH exchange and SAs

```
HDR, IDi, SIG(msg1,msg2)    -->
                        <-- HDR, IDr, SIG(msg1,msg2)
```

Identity protection omitted (since core exchange authentication does not depend on it)

Stage 3: derive keys (Ka) from g^{ir} and use them to protect ipsec transform negotiation

```
HDR*, SA, TSi, TSr    -->
                      <--    HDR*, SA, TSi, TSr
```

Rationale flaw and solution

Above rationale is flawed:

- 3-stage protocol is insecure (why? hint: DVW)
- security is "miraculously" saved by the piggy-backing of stage 2 on stage 3 (hint: SIGMA)

Lesson: Define exact inputs to SIG and MAC (explicitly ensure essential elements)

Recommendation: Make explicit that ID goes under the MAC (regardless of ID's position in the protocol); sign everything you send and the other party's nonce.

Pre-Shared Secret Key

Based on any of the SIGMA variants:

- just do not use the signature (but MAC only)
- How to identify the shared key without revealing I's id:
  (i) point to the shared key via a key-id (static or dynamic) passed in first message, or
  (ii) derive K_e directly from g^{xy}
  - option (i) gives active protection to I and R, option (ii) gives passive to I
- superior defense against DH cryptanalysis
- shares all protocol mechanisms with sig mode! (adds minimal complexity)
- intended for use with strong keys (machine generated and stored); many applications...