The odd couple: MQV and HMQV

Jean-Philippe Aumasson

n|w University of Applied Sciences Northwestern Switzerland
School of Engineering

Summary

MQV = EC-DH-based key agreement protocol,

- proposed by Menezes, Qu and Vanstone (1995),
- improved with Law and Solinas (1998),
- widely standardized (ANSI, ISO/IEC, IEEE), and recommended (NIST, NSA suite B).

HMQV = variant of MQV,

- proposed by Krawczyk (2005),
- attacked by Menezes,
- validity of attacks unclear.

References

A. Menezes, M. Qu, S. Vanstone. Some new key agreement protocols providing implicit authentication. SAC'95.

L. Law, A. Menezes, M. Qu, J. Solinas, S. Vanstone. An efficient protocol for authenticated key agreement. Design, Codes, and Cryptography, 2003.

H. Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. CRYPTO' 05. Full version on ePrint (2005/176).

A. Menezes. Another look at HMQV. ePrint (2005/205). Journal of Mathematical Cryptology, 2007.

A. Menezes, B. Ustaoglu. On the importance of public-key validation in the MQV and HMQV key agreement protocols. INDOCRYPT'06.

Road map

PART I

- Key agreement protocols
- Elliptic curves
- The MQV protocols

PART II

- The “insecurity” of MQV
- HMQV
- The “insecurity” of HMQV

CONCLUSION

PART I

Key agreement protocols

Taxonomy

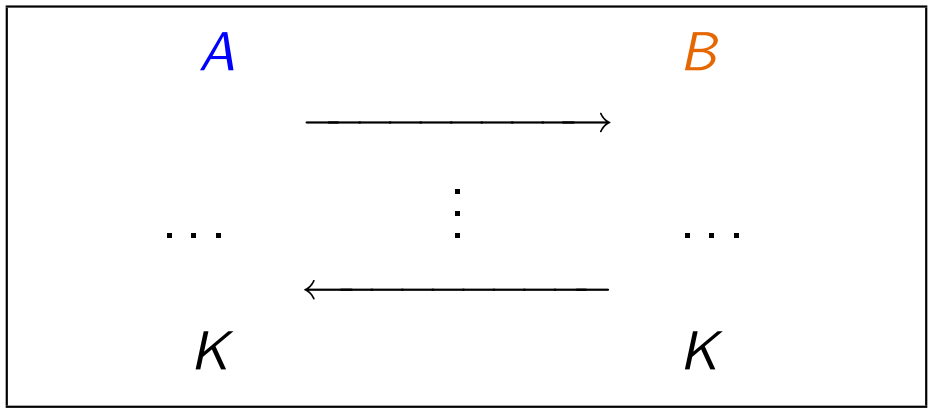
From the HAC. . .

Key establishment is a protocol whereby a shared secret becomes available to two or more parties, for subsequent cryptographic use.

A **key transport** protocol is a key establishment technique where one party creates or otherwise obtains a secret value, and securely transfers it to the other(s).

A **key agreement** protocol is a key establishement in which a shared secret is derived by two (or more) parties as a function of information contributed by, or associated with, each of these, (ideally) such that no party can predetermine the resulting value.

Key agreement



Entities can *communicate*, and run *computations*.

Either based on symmetric or *asymmetric* techniques.

Efficiency considered in terms

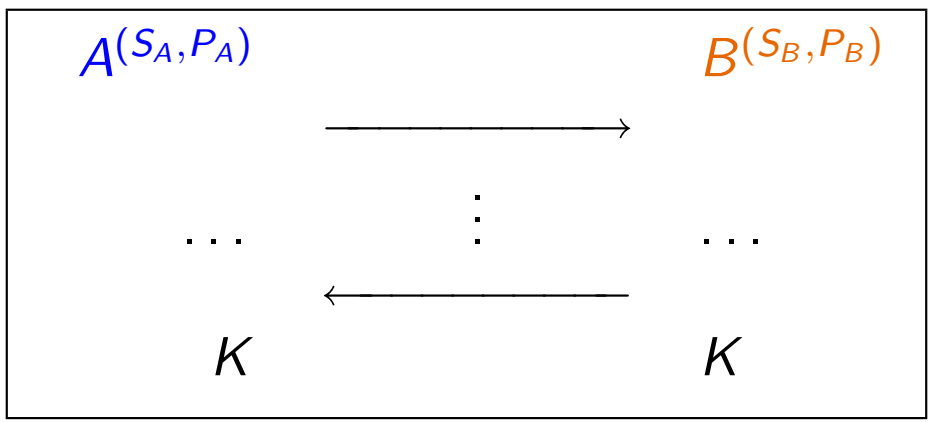
- number of messages sent (*passes*),
- amount of data per message,
- complexity of computations,
- possibility of precomputation.

Authenticated key agreement (AK)

Both entities are assured that no one else can learn the value of the established key.

$\Rightarrow$ need *authenticated information* (P_A, P_B)
(e.g. via certificates and a CA)

S_A and S_B are long-term private keys



A knows P_B, B knows P_A.

Share value K called the **(ephemeral) session key**.

AK with key confirmation (AKC)

An AK protocol provides key confirmation of B to A if A is assured that B *actually possesses* the key.

- $\approx$ proof of knowledge,
- in practice, show $H(K)$ for a one-way hash function H.

AK with key confirmation of *both* A and B:
authenticated key agreement with key confirmation.

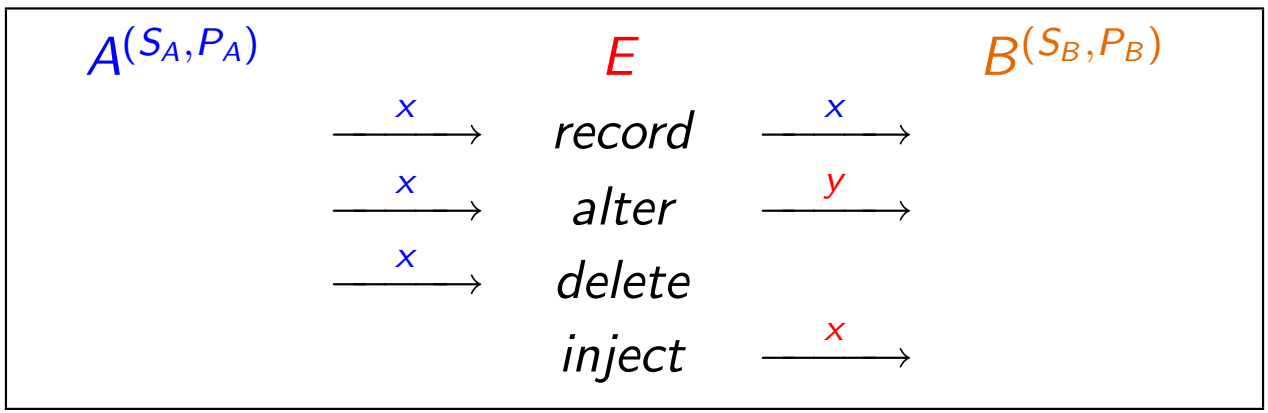
Question 1

What does "secure" mean for a key agreement protocol ?

Security model

Formal model of [Blake-Wilson-Johnson-Menezes 97], variant of [Bellare-Rogaway 94], considered for MQV.

Participants communicate through an insecure channel so that an **adversary** can



- *reveal* previous session keys computed,
- *corrupt* entities (i.e. get read/write access to its long-term secret)
- *initiate sessions*, etc.

Desirable attributes

Key-compromise impersonation: if A's secret is disclosed, E can impersonate A, but should not be able to impersonate other entities.

Known-key security: if some session keys are compromised, future sessions should still achieve their goal.

Forward secrecy: if long-term private keys are compromised, previous session keys should remain secret.

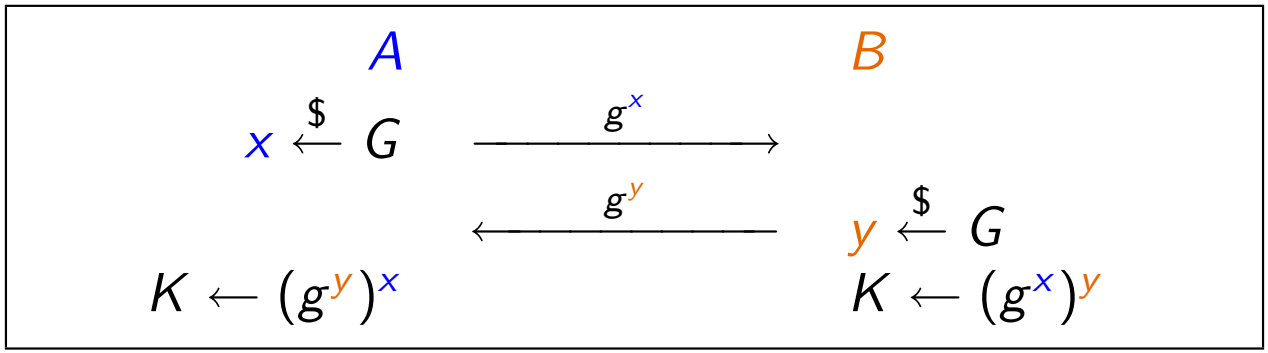
Key control: neither entity should be able to force the session key to a preselected value (or subset thereof).

Other attributes: unknown key-share, message independence, loss of information, etc., see [Blake-Wilson-Johnson-Menezes 97]

Question 2

Is the basic Diffie-Hellman protocol secure in the Canetti-Krawczyk model ?

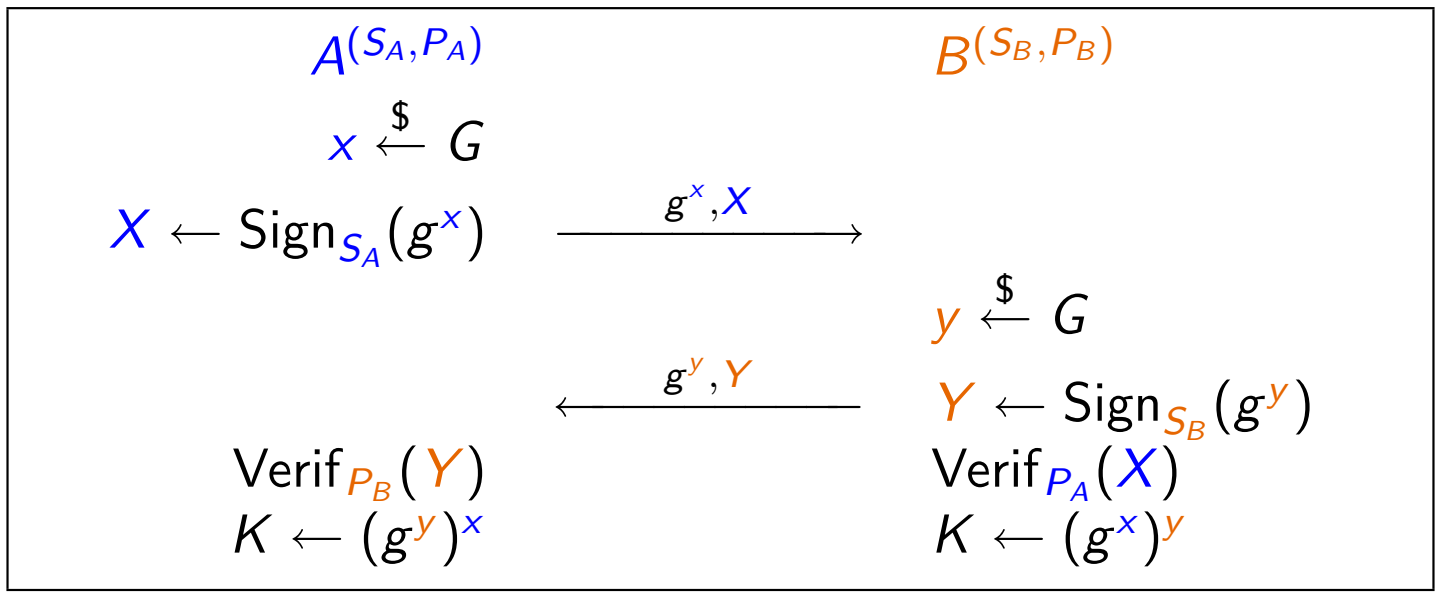
Basic Diffie-Hellman



A has no assurance that she communicates with B (since has *no authenticated information*).

Secrecy for a passive adversary only.

Authenticated Diffie-Hellman



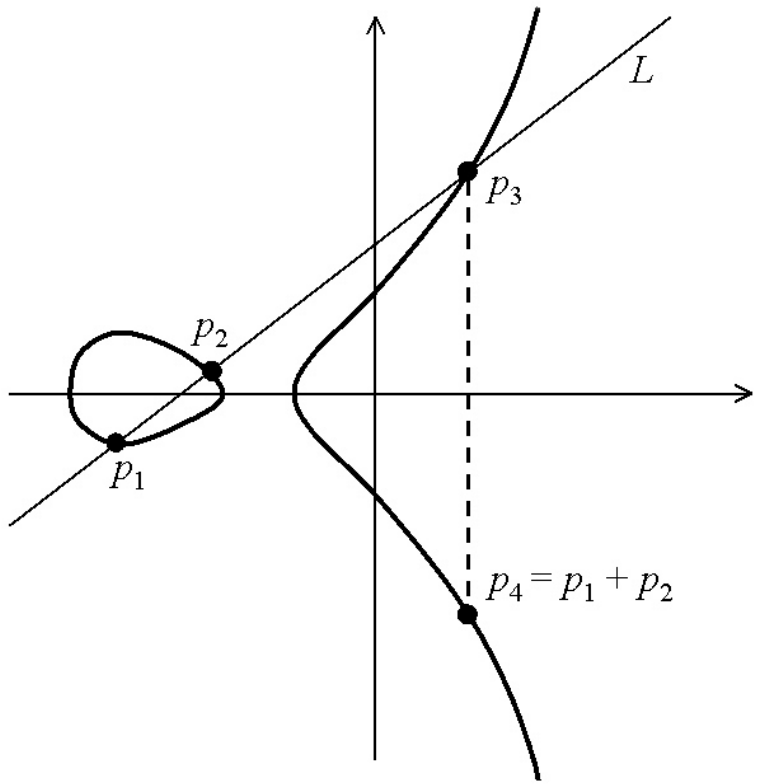
Leakage of an x allows impersonation.

Elliptic curves

Idea

[Miller 85, Koblitz 87]

Plane curve E, with an abelian group ($E(\mathbb{F}_q)$, $+$), in which DL is hard.



Cryptographic curves

Curve E of equation

$$y^2 = x^3 + ax + b.$$

$E(\mathbb{F}_q) =$ set of points with (affine) coordinates in $\mathbb{F}_q \times \mathbb{F}_q$ (and $\mathcal{O}$).

Well-defined addition law (coordinates-dependent).

Typically, $q = p$ or $q = 2^d$ for crypto curves.

Elliptic Curve Diffie-Hellman Problem (ECDLP):
Given a *base point* P, and $Q = rP$, find the random integer r.

Domain parameters

- A **field size** q (typically $q = 2^d$),
- A **curve**, i.e. two field-elements $a, b \in \mathbb{F}_q$,
- A **base point** $P = (x_P, y_P) \in E(\mathbb{F}_q)$,

$\Rightarrow$ constraints on E (special classes, etc.), P (order, etc.), e.g.

$$\mathrm{ord}(P) = n \in \mathbb{P},\ \#E(\mathbb{F}_q) = np,\ n > 4\sqrt{q}.$$

In practice, curve selected *at random*, then pass through a process of domain parameter validation.

Key pair generation

Private key: integer n
Public key: $Q = nP$

$\Rightarrow$ finding the private key reduces to ECDL.

Pair chosen *at random*, need a **public key validation**, to check consistency of (n, Q):

- $Q \stackrel{?}{=} \mathcal{O}$
- $(x_Q, y_Q) \in \mathbb{F}_q \times \mathbb{F}_q$?
- $Q \in E(\mathbb{F}_q)$?
- $nQ \stackrel{?}{=} \mathcal{O}$

The MQV protocols

MQV

[Menezes-Qu-Vanstone 95]

Key idea: A and B use ephemeral key pairs, in addition to their long-term pairs.

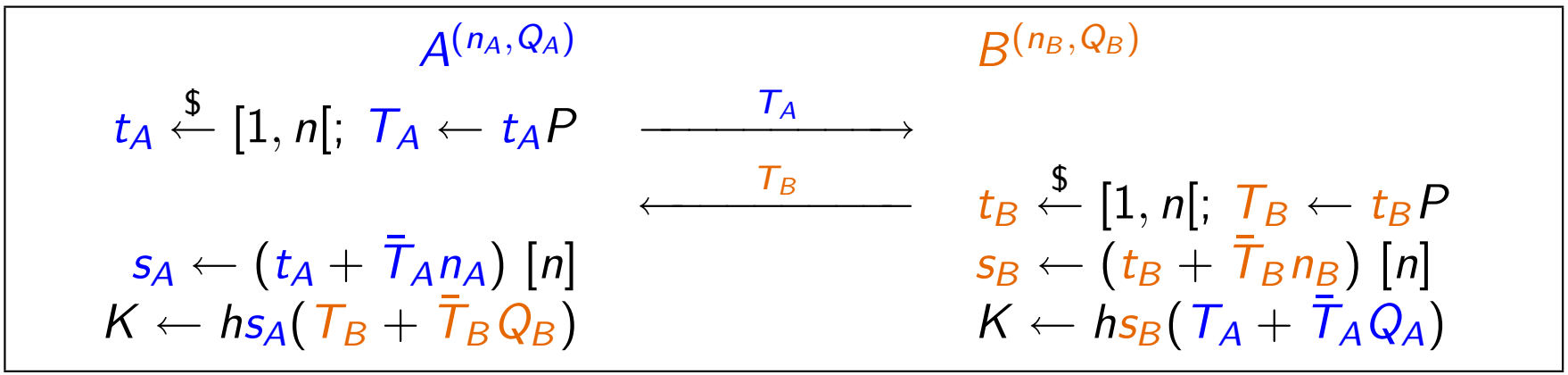
Use of a mapping $E(\mathbb{F}_q) \mapsto \mathbb{Z}/n\mathbb{Z}$, *converting* a point to an integer,

$$E(\mathbb{F}_q) \ni Q \mapsto \bar{Q} \in \mathbb{Z}/n\mathbb{Z},$$

by considering the binary representation of Q's x-coordinate.

The protocol

1. A and B generate *ephemeral* key pairs (t_A, T_A), (t_B, T_B),
2. They run a DH with *implicit signatures* s_A and s_B.



With $h = \#E(\mathbb{F}_q)/n$ (*cofactor*).

Post-processing of K with a key derivation function (e.g. hash).

The protocol

$A^{(n_A, Q_A)}$		$B^{(n_B, Q_B)}$
$t_A \xleftarrow{\$} [1, n[;\ T_A \leftarrow t_A P$	$\xrightarrow{T_A}$	
	$\xleftarrow{T_B}$	$t_B \xleftarrow{\$} [1, n[;\ T_B \leftarrow t_B P$
$s_A \leftarrow (t_A + \bar{T}_A n_A)\ [n]$		$s_B \leftarrow (t_B + \bar{T}_B n_B)\ [n]$
$K \leftarrow h s_A (T_B + \bar{T}_B Q_B)$		$K \leftarrow h s_B (T_A + \bar{T}_A Q_A)$

Correctness (symmetric):

$$K = h s_A (T_B + \bar{T}_B Q_B) = h s_A (t_B + \bar{T}_B n_B) P = h s_A s_B P.$$

The protocol

$A^{(n_A, Q_A)}$		$B^{(n_B, Q_B)}$
$t_A \xleftarrow{\$} [1, n[;\ T_A \leftarrow t_A P$	$\xrightarrow{T_A}$	
	$\xleftarrow{T_B}$	$t_B \xleftarrow{\$} [1, n[;\ T_B \leftarrow t_B P$
$s_A \leftarrow (t_A + \bar{T}_A n_A)\ [n]$		$s_B \leftarrow (t_B + \bar{T}_B n_B)\ [n]$
$K \leftarrow h s_A (T_B + \bar{T}_B Q_B)$		$K \leftarrow h s_B (T_A + \bar{T}_A Q_A)$

Why use of the *cofactor* $h = \#E(\mathbb{F}_q)/n$?
Ensures that K is a point in the subgroup of order n in $E(\mathbb{F}_q)$.

Questions 3 & 4

Does MQV require a third (trusted) party ?

Why using $H(K)$ as effective session key rather than K ?

Performance attributes

2 *passes*.

2.5 *scalar multiplications* ($\mathbb{Z} \times E(\mathbb{F}_q) \mapsto E(\mathbb{F}_q)$) per party,
e.g. for A:

- $t_A \times P$ (1 multiplication)
- $\bar{T}_B \times Q_B$ (0.5 mul.)
- $s_A \times (\dots)$ (1 mul.)

"0.5" because $\bar{T}_B$ uses only *half the bits* of the x-coordinate of T_B.

Role-symmetric (messages have same structure for each entity).

Non-interactive (messages independent of each other).

Security attributes

No security reduction; *"appears to have the security attributes of known-key security, forward secrecy, key-compromise impersonation, and key control (. . .)."* [Menezes-Qu-Vanstone 95]

Attacks in [Krawczyk 05], see **PART II**.

Security reductions to a DH-like problem
in [Kunz-Jacques-Pointcheval-06].

To be continued...

PART II

The “insecurity” of MQV

Summary

[Krawczyk 05]

- argues that MQV has *several weaknesses*,
- defines HMQV to fix some of them,
- presents rigorous analysis, with security *proofs*.

Uses the Canetti-Krawczyk model [Canetti-Krawczyk 01], including e.g. **state-reveal** queries.

MQV weaknesses

(with paraphrases of *Menezes' response* [Menezes 05])

- Security depends on the group representation
  - *Exploits unrealistic "contrived group representations", and representations allowed by standards are okay.*
- Existence of unknown key-share attacks
  - *There exists simple countermeasures, mentioned in latest standards.*
- Lack of perfect forward secrecy
  - *Not proper to MQV, but to any 2-pass protocol, including HMQV.*
- Possibility of key-compromise impersonation
  - *Need very powerful adversary (getting A's secret and B's inner state), applies to HMQV as well.*

+ observations that key validations require expensive extra-computation (including from the CA).

Unknown key-share attack

[Kaliski 01]

Idea: coerce A and B to establish a key s.t. B doesn't know that the key is shared with A (but believes it is with E).

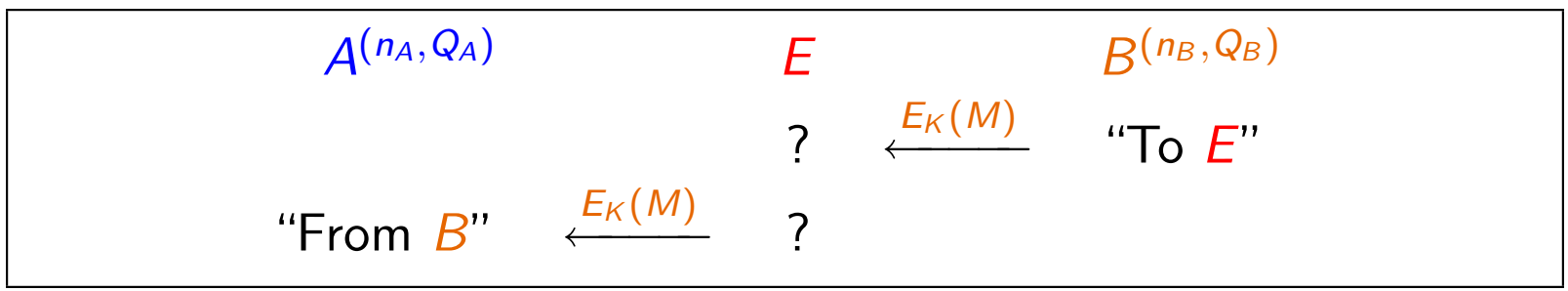
E doesn't know K !

Also called *source-substitution attack*.

Unknown key-share attack

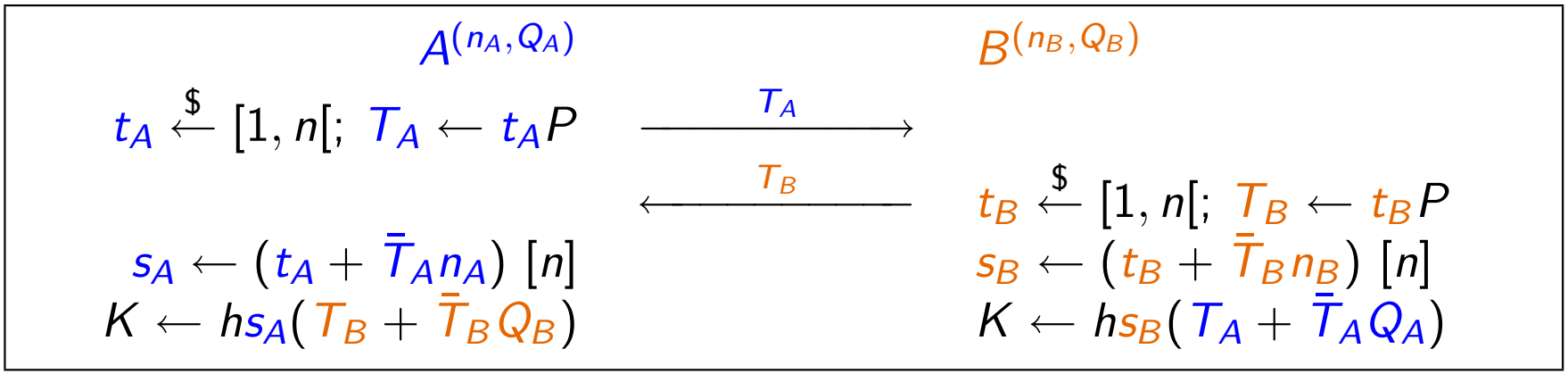
After the attack: A and B share a secret K (unknown by E), but B believes K is shared with E.

Ex. of application: B sends a message M (e.g. *"are you free tonight?"*) to E, protected with K

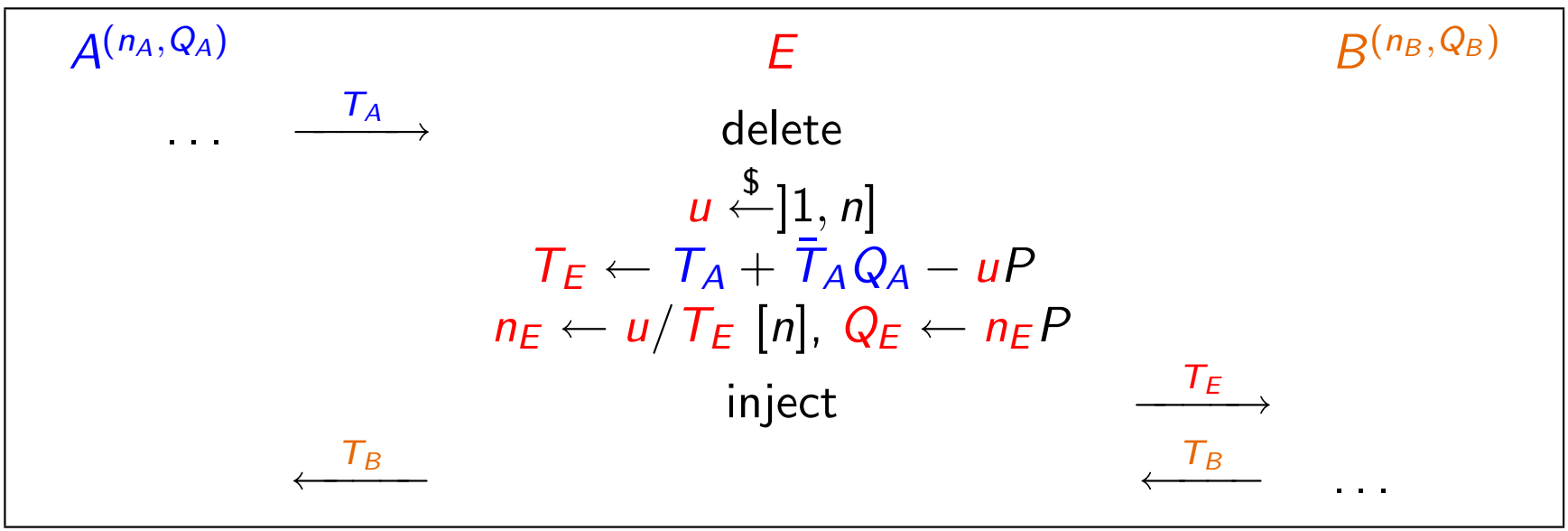


A knows she shares K with B, hence she will believe that the message is sent by B, while B intends to send it to E.

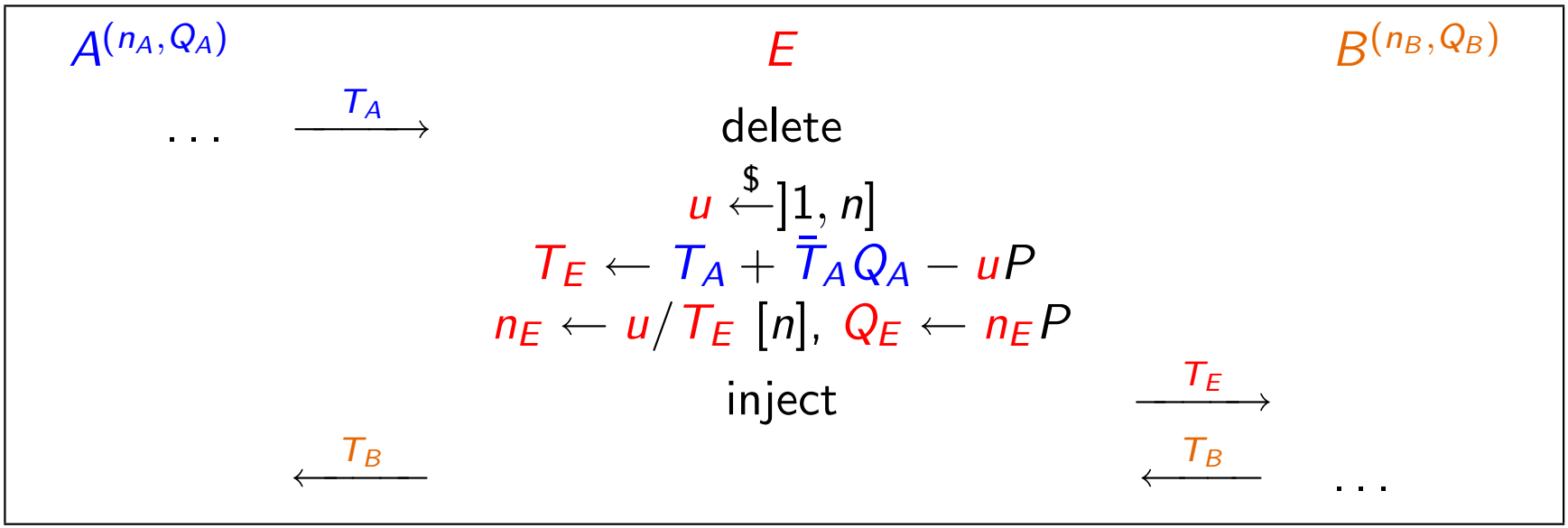
Unknown key-share attack



Online attack (E has to get (n_E, Q_E) certified!):



Unknown key-share attack

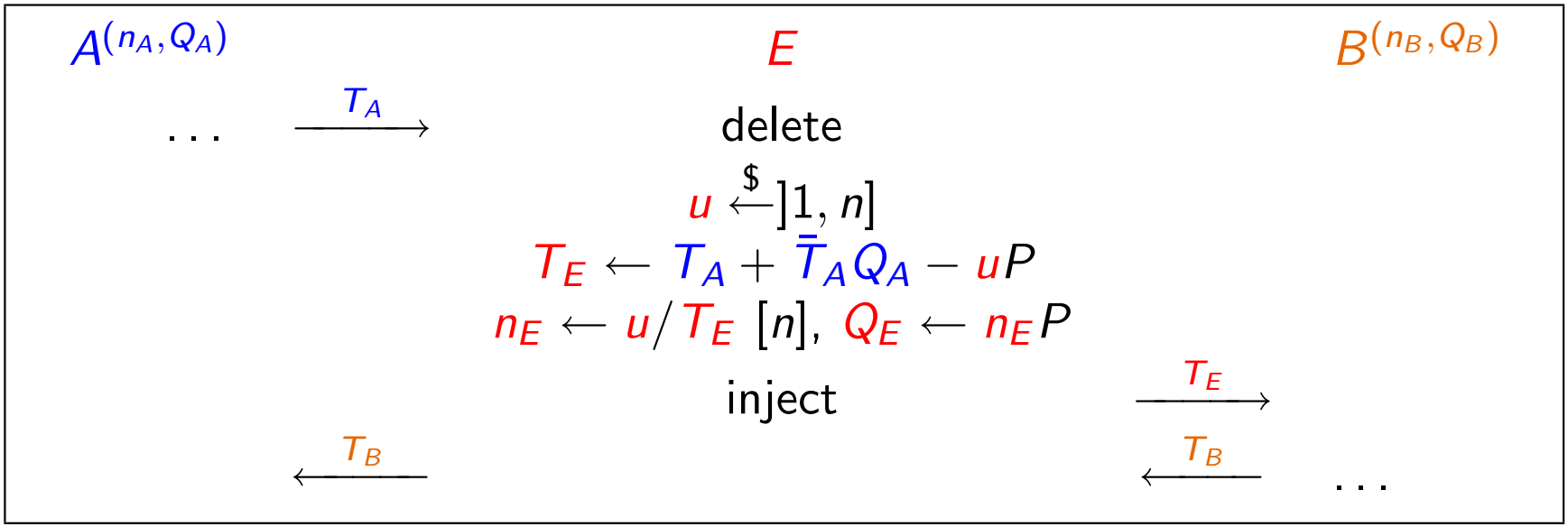


Key trick: B computes

$$K \leftarrow hs_B(T_E + \bar{T}_E Q_E) = hs_B(T_A + \bar{T}_A Q_A).$$

$\Rightarrow$ valid key shared with A.
E does *not know* K, since $t_E = \log T_E$ unknown.

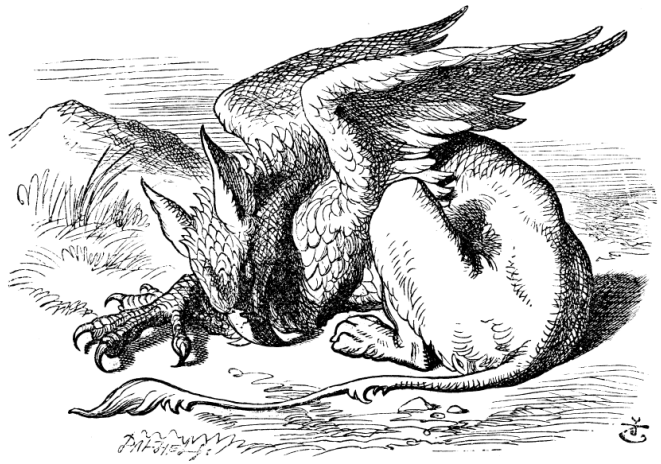
Unknown key-share attack



Countermeasures:

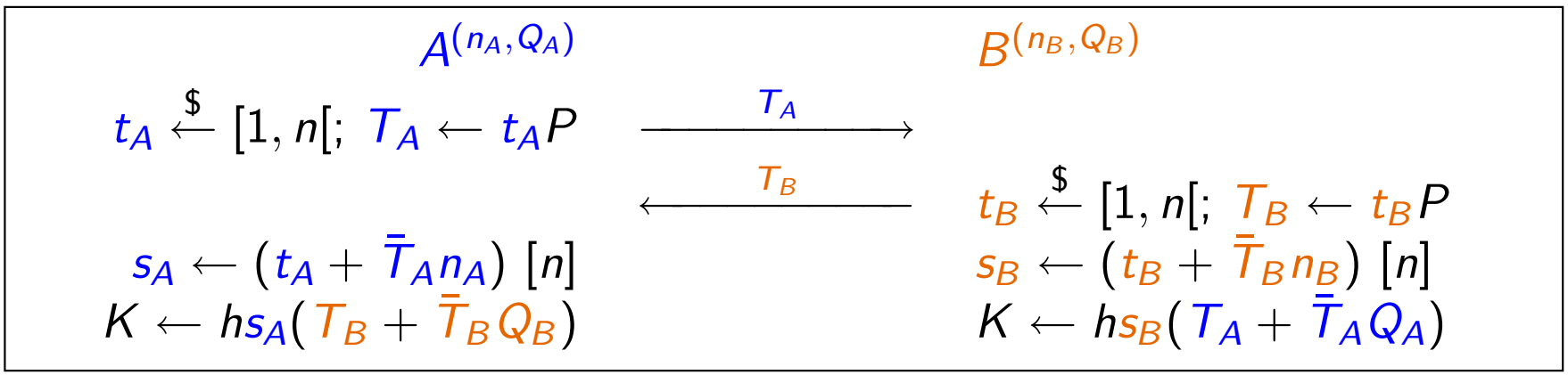
- *Ephemeral key commitment*: exchange $H(T_A)$ and $H(T_B)$ before T_A and T_B.
- *Delay detection*: since E's operations relatively long.
- *Certificate aging*: require "old enough" certificates.

HMQV



The protocol

MQV:



HMQV:
Replace $\bar{T}_A$ by $\bar{H}(T_A, B)$, and $\bar{T}_B$ by $\bar{H}(T_B, A)$.

Objective:

- need no assumption on the *group representation*,
- avoid the need for *key validation*.

Advantages over MQV

- Extensive analysis, *security proofs*
- Same or better *performance*, compared to MQV
- Additional *security features*

Proofs: security in the Canetti-Krawczyl model, KCI, weak forward secrecy, resilience to ephemeral exponents leakage.
w.r.t. CDH, GDH, KEA1, in the *random oracle model*

The “insecurity” of HMQV

Attacks on HMQV

[Menezes 05] presented

- attacks on HMQV, contradicting the security proof,
- flaws in 2 proofs.

Attacks exploit omission of key validation, and need knowledge of ephemeral exponents.

Flaws in the proof because of no key validation.
"Menezes' claim that the proof of XCR signature is flawed is incorrect"[Krawczyk 05]

Small-subgroup attack

Key idea: combine **state-reveal** and **session-key** queries, to get relations

$$\begin{aligned} n_B &= c_1 \mod t_1 \\ n_B &= c_2 \mod t_2 \\ \dots &\quad \dots \\ n_B &= c_k \mod t_k \end{aligned}$$

and then find n_B by the CRT's isomorphism.

Details in [Menezes 05]

CONCLUSION

Summary

Remarkable facts that

- MQV had no security proof (until [Kunz-Jacques-Pointcheval-06]),
- while HMQV had some...
- but partially incorrect.

Still, HMQV benefits of

- a rigorous analysis,
- great performance (as MQV).

Final words

Designing/analysing AK protocols is particularly difficult;

- multitude of attack scenarios, problem of deciding their realism,
- complexity of formal models, and analysis within,
- importance of the temporal dimension.

+ delicate issue of *"provable security"* (see Koblitz's arguments, and subsequent tentative responses)...

The odd couple: MQV and HMQV

Jean-Philippe Aumasson

n|w University of Applied Sciences Northwestern Switzerland
School of Engineering