Radbound University

Authenticated Key-Exchange using Digital Signatures.

Applied Cryptography – Spring 2024

Simona Samardjiska

May 13, 2024

Institute for Computing and Information Sciences
Radboud University

Summary

Last time:

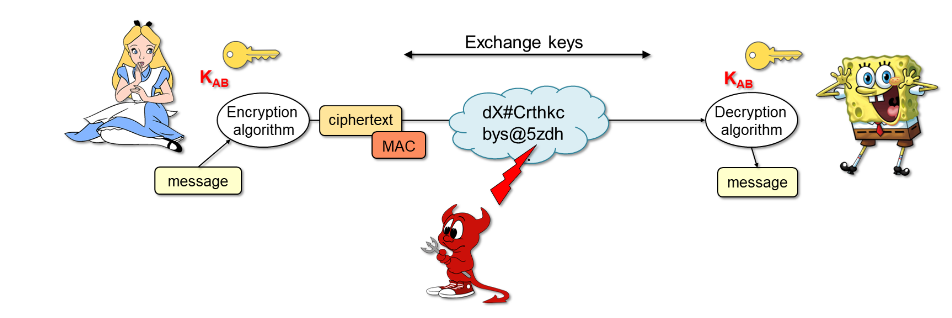
- Introduction to Post-Quantum Cryptography
- Hash-Based Signatures

Today:

- Authenticated Key-Exchange using Digital Signatures
- Post-Quantum cryptography chalenges and families

Authenticated Key-Exchange using Digital Signatures

Recall our everyday scenario



- Alice and Bob have **not** agreed on a joint key yet, but they want to communicate securely
  - They want to exchange symmetric keys over the public channel, first
  - They use public key cryptography for this, so that Eve can't learn the key
  - Typically using (Merkle-)Diffie-Hellman key exchange

Recall the magic of (Merkle-)Diffie-Hellman key exchange

Public-key based establishment of a shared secret

Alice's client		Bob's server
P, Q, G		P, Q, G
$a \xleftarrow{\$} \mathbb{Z}_q$		
$A \leftarrow G^a$	$\xrightarrow{\text{Alice};A}$	$b \xleftarrow{\$} \mathbb{Z}_q$
	$\xleftarrow{\text{Bob};B}$	$B \leftarrow G^b$
$K_{A,B} \leftarrow B^a$		$K_{B,A} \leftarrow A^b$

Alice and Bob arrive at the same shared secret $K_{A,B} = K_{B,A}$

$$K_{A,B} = (B^a) = (G^b)^a = G^{b \cdot a} = G^{a \cdot b} = (G^a)^b = A^b = K_{B,A}$$

- Alice and Bob derive key(s) from secret: $K \leftarrow KDF(K_{A,B})$
- They use K further in their communication for encryption or message authentication
- **Perfect Forward Secrecy (PFS)**
  - Once the session keys are destroyed they can not be recovered even by the parties that created them

Man-in-the-middle attack on (Merkle-)Diffie-Hellman Key Exchange

Alice's client		Eve		Bob's server
P, Q, G		P, Q, G		P, Q, G
$a \xleftarrow{\$} \mathbb{Z}_q$		$e \xleftarrow{\$} \mathbb{Z}_q$		$b \xleftarrow{\$} \mathbb{Z}_q$
$A \leftarrow G^a$	$\xrightarrow{\text{Alice};A}$	$E \leftarrow G^e$	$\xrightarrow{\text{Alice};E}$	
			$\xleftarrow{\text{Bob};B}$	$B \leftarrow G^b$
	$\xleftarrow{\text{Bob};E}$			
$K_{A,B} \leftarrow E^a$		$K_{A,B} \leftarrow A^e$		
		$K_{B,A} \leftarrow B^e$		$K_{B,A} \leftarrow E^b$

- Alice and Bob both unknowingly share a secret with Eve,
- In subsequent exchange protected with shared secrets
  - **Eve decrypts, can read plaintext, and re-encrypts**
  - Eve may **modify/delete messages and compute tags**
- **Problem: Alice and Bob can never be sure who sent the message**
- **Solution: Entity authentication/Identification**
  - Alice must verify B really comes from Bob and vice versa

Let's try to fix it

- Alice has a long term signing key pair $(\mathsf{pk}_A, \mathsf{sk}_A)$ and Bob has $(\mathsf{pk}_B, \mathsf{sk}_B)$
- **Very important:**
  - Their long term keys are already authenticated! (for ex. out of band. Alternatively, certificates can be sent together with the identities!)
  - Alice knows pk_B belongs to Bob and Bob knows pk_A belongs to Alice

The fix:

Alice's client		Bob's server
$P, Q, G, \mathsf{pk}_B, \mathsf{sk}_A$		$P, Q, G, \mathsf{pk}_A, \mathsf{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q$		
$A \leftarrow G^a$	$\xrightarrow{\text{Alice}; A}$	$b \xleftarrow{\$} \mathbb{Z}_q$
$\mathsf{Vf}_{\mathsf{pk}_B}(\sigma_B)$	$\xleftarrow{\text{Bob};\ B;\ \sigma_B}$	$B \leftarrow G^b,\ \sigma_B = \mathsf{Sign}_{\mathsf{sk}_B}(A, B)$
$\sigma_A = \mathsf{Sign}_{\mathsf{sk}_A}(B, A)$	$\xrightarrow{\sigma_A}$	$\mathsf{Vf}_{\mathsf{pk}_A}(\sigma_A)$
$K_{A,B} \leftarrow B^a$		$K_{B,A} \leftarrow A^b$

- "Authentication-only" Station-to-Station Protocol (Diffie et al. '92)
  - **claimed mutual authentication!**
- Flawed! Man-in-the-middle attacks possible!

Man-in-the-middle attack on the "Auth.-only" STS protocol – Attack 1

Alice's client		Eve		Bob's server
$P, G, \mathrm{pk}_B, \mathrm{pk}_E, \mathrm{sk}_A$		$P, G, \mathrm{pk}_A, \mathrm{pk}_B, \mathrm{sk}_E$		$P, G, \mathrm{pk}_A, \mathrm{pk}_E, \mathrm{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q$				$b \xleftarrow{\$} \mathbb{Z}_q$
$A \leftarrow G^a$	$\xrightarrow{\text{Alice; } A}$		$\xrightarrow{\text{Alice; } A}$	
				$B \leftarrow G^b$
$\mathrm{Vf}_{\mathrm{pk}_E}(\sigma_E)$	$\xleftarrow{\text{Eve; } B;\ \sigma_E}$	$\sigma_E = \mathrm{Sign}_{\mathrm{sk}_E}(A, B)$	$\xleftarrow{\text{Bob; } B;\ \sigma_B}$	$\sigma_B = \mathrm{Sign}_{\mathrm{sk}_B}(A, B)$
$\sigma_A = \mathrm{Sign}_{\mathrm{sk}_A}(B, A)$	$\xrightarrow{\sigma_A}$		$\xrightarrow{\sigma_A}$	$\mathrm{Vf}_{\mathrm{pk}_A}(\sigma_A)$
$K_{A,B} \leftarrow B^a$				$K_{B,A} \leftarrow A^b$

- Alice thinks she is talking to Eve,
- Bob thinks he is talking to Alice
- So, no mutual authentication!, but Eve does not know the key
- **Identity misbinding attack**
  - but Eve can trick Alice into saying things not intended for Bob, but for Eve
  - and then just relay them to Bob
  - Think of: I agree to buy your house for 1/2 million (Eve has a mansion, and Bob a shed)

Man-in-the-middle attack on the “Auth.-only” STS protocol – Attack 2

Alice’s client		Eve		Bob’s server
$P, G, \mathsf{pk}_B, \mathsf{pk}_E, \mathsf{sk}_A$		$P, G, \mathsf{pk}_A, \mathsf{pk}_B, \mathsf{sk}_E$		$P, G, \mathsf{pk}_A, \mathsf{pk}_E, \mathsf{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q$				$b \xleftarrow{\$} \mathbb{Z}_q$
$A \leftarrow G^a$	$\xrightarrow{\text{Alice};\ A}$		$\xrightarrow{\text{Eve};\ A}$	
				$B \leftarrow G^b$
$\mathsf{Vf}_{\mathsf{pk}_B}(\sigma_B)$	$\xleftarrow{\text{Bob};\ B;\ \sigma_B}$		$\xleftarrow{\text{Bob};\ B;\ \sigma_B}$	$\sigma_B = \mathsf{Sign}_{\mathsf{sk}_B}(A, B)$
$\sigma_A = \mathsf{Sign}_{\mathsf{sk}_A}(B, A)$	$\xrightarrow{\sigma_A}$	$\sigma_E = \mathsf{Sign}_{\mathsf{sk}_E}(B, A)$	$\xrightarrow{\sigma_E}$	$\mathsf{Vf}_{\mathsf{pk}_E}(\sigma_E)$
$K_{A,B} \leftarrow B^a$				$K_{B,A} \leftarrow A^b$

- Alice thinks she is talking to Bob,
- Bob thinks he is talking to Eve
- So, no mutual authentication!, but Eve does not know the key
- Again, **Identity misbinding attack**
- but Bob thinks everything from Alice comes from Eve
- Similar effect as previously, but now Eve intercepts the initial message from Alice intended for Bob
- in Attack 1, Eve uses a legitimate initial message from Alice

A fixed protocol – ISO-9796

Alice's client		Bob's server
$P, Q, G, \mathsf{pk}_B, \mathsf{sk}_A$		$P, Q, G, \mathsf{pk}_A, \mathsf{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q,\ A \leftarrow G^a$	$\xrightarrow{\text{Alice}; A}$	$b \xleftarrow{\$} \mathbb{Z}_q$
$\mathsf{Vf}_{\mathsf{pk}_B}(\sigma_B)$	$\xleftarrow{\text{Bob};\ B;\ \sigma_B}$	$B \leftarrow G^b,\ \sigma_B = \mathsf{Sign}_{\mathsf{sk}_B}(A, B, \mathit{Alice})$
$\sigma_A = \mathsf{Sign}_{\mathsf{sk}_A}(B, A, \mathit{Bob})$	$\xrightarrow{\sigma_A}$	$\mathsf{Vf}_{\mathsf{pk}_A}(\sigma_A)$
$K_{A,B} \leftarrow B^a$		$K_{B,A} \leftarrow A^b$

- Proven secure [Canetti-Krawzyk '01]
- **Include the identity of the receiver in the signature**
- The identities of Alice and Bob are bound to the key $K_{A,B}$ and the previous attacks don't work
  - Eve can't just relay the message sent by Bob (Attack 2)
  - Bob; B; $\mathsf{Sign}_{\mathsf{sk}_B}(A, B, \mathit{Eve})$ needs to be Bob; B; $\mathsf{Sign}_{\mathsf{sk}_B}(A, B, \mathit{Alice})$
  - Where does Attack 1 fail?
- Does including the identity of the sender accomplish the same? (See homework :))

Issues with ISO-9796

- Proven secure but...
- **Neither initiator nor responder privacy protection**: both Alice and Bob need the identity of the peer before being able to proceed with the protocol
  - For ex, roaming users/browsers need initiator identity protection
  - Or NFC cards need responder identity protection
  - Can't be fixed for active attackers (as are Attacks 1 and 2), as the identity of the peer must be known before authentication (to be included in the signature)
- **Non-repudiability**: by signing the identity of the peer, one leaves a non-deniable proof of communication with that peer
- How to fix these issues?
  - Can we have identity protection for both peers, or even at least for one of them?
  - Major issue in many protocols today
  - What about TLS 1.2?

TLS 1.2 ephimeral version (TLS-DHE)

Alice's client		Bob's server
P, Q, G, sk_A		P, Q, G, sk_B
Ciphersuite negotiation		
$R_A \leftarrow \mathcal{R}$	$\xrightarrow{\text{supported suites; } R_A}$	
	$\xleftarrow{\text{selected suites; } R_B}$	$R_B \leftarrow \mathcal{R}$
Key exchange		
$\mathsf{Vf}_{\mathsf{pk}_B}(\sigma_B)$	$\xleftarrow{\text{Bob } (\mathsf{pk}_B);\ \sigma_B, B}$	$b \xleftarrow{\$} \mathbb{Z}_q,\ B \leftarrow G^b,\ \sigma_B = \mathsf{Sign}_{\mathsf{sk}_B}(B)$
$a \xleftarrow{\$} \mathbb{Z}_q, A \leftarrow G^a,\ \sigma_A = \mathsf{Sign}_{\mathsf{sk}_A}(A)$	$\xrightarrow{\text{Alice } (\mathsf{pk}_A);\ \sigma_A, A}$	$\mathsf{Vf}_{\mathsf{pk}_A}(\sigma_A)$
$K_{conf} \leftarrow B^a$		$K_{conf} \leftarrow A^b$
$K_{A,B} \leftarrow (K_{conf}, R_A, R_B)$		$K_{B,A} \leftarrow KDF(K_{conf}, R_A, R_B)$
Authentication		
	$\xleftarrow{MAC_{K_{A,B}}(finnish_B)}$	$finnish_B \leftarrow 1 \vert transcript$
$finnish_B \leftarrow 0 \vert transcript$	$\xrightarrow{MAC_{K_{A,B}}(finnish_A)}$	

TLS 1.2 ephimeral version (TLS-DHE)

- Misbinding not possible, why? (due to last phase)
- No identity protection, certificates sent in clear!
- Everybody can see the website you are browsing to
- Good thing - no signing of peer identity

Disclaimer:

- Some messages on the previous slide are overly simplified. Be ware!
  - For instance, certificates are sent in separate messages
  - The server may or may not ask for the clients certificate, client authentication is optional.
  - Several other values and keys are generated (there is premaster key, master key and session key)
- You will here more on the complications and other issues with TLS 1.2 next time
  - No authentication of ciphersuite, TLS version etc.
- Now let's go back to the simplicity of the STS protocol

Recall the Auth.only STS protocol and Attack 2

Alice's client		Bob's server
$P, Q, G, \mathsf{pk}_B, \mathsf{sk}_A$		$P, Q, G, \mathsf{pk}_A, \mathsf{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q$		
$A \leftarrow G^a$	$\xrightarrow{\text{Alice}; A}$	$b \xleftarrow{\$} \mathbb{Z}_q$
$\mathsf{Vf}_{\mathsf{pk}_B}(\sigma_B)$	$\xleftarrow{\text{Bob};\ B;\ \sigma_B}$	$B \leftarrow G^b,\ \sigma_B = \mathsf{Sign}_{\mathsf{sk}_B}(A, B)$
$\sigma_A = \mathsf{Sign}_{\mathsf{sk}_A}(B, A)$	$\xrightarrow{\sigma_A}$	$\mathsf{Vf}_{\mathsf{pk}_A}(\sigma_A)$
$K_{A,B} \leftarrow B^a$		$K_{B,A} \leftarrow A^b$

Recall the Auth.only STS protocol and Attack 2

Alice's client		Eve		Bob's server
$P, G, \mathsf{pk}_B, \mathsf{pk}_E, \mathsf{sk}_A$		$P, G, \mathsf{pk}_A, \mathsf{pk}_B, \mathsf{sk}_E$		$P, G, \mathsf{pk}_A, \mathsf{pk}_E, \mathsf{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q$				$b \xleftarrow{\$} \mathbb{Z}_q$
$A \leftarrow G^a$	$\xrightarrow{\text{Alice; } A}$		$\xrightarrow{\text{Eve; } A}$	
				$B \leftarrow G^b$
$\mathsf{Vf}_{\mathsf{pk}_B}(\sigma_B)$	$\xleftarrow{\text{Bob; } B;\ \sigma_B}$		$\xleftarrow{\text{Bob; } B;\ \sigma_B}$	$\sigma_B = \mathsf{Sign}_{\mathsf{sk}_B}(A, B)$
$\sigma_A = \mathsf{Sign}_{\mathsf{sk}_A}(B, A)$	$\xrightarrow{\sigma_A}$	$\sigma_E = \mathsf{Sign}_{\mathsf{sk}_E}(B, A)$	$\xrightarrow{\sigma_E}$	$\mathsf{Vf}_{\mathsf{pk}_E}(\sigma_E)$
$K_{A,B} \leftarrow B^a$				$K_{B,A} \leftarrow A^b$

- Recall that although no authentication is provided, Eve does not know the exchanged key $K_{A,B}$
- So why don't we use $K_{A,B}$ somehow to protect against the ID-misbinding?
- The full version of the STS protocol does exactly this

STS protocol (full version)

Alice's client		Bob's server
$P, Q, G, \mathsf{pk}_B, \mathsf{sk}_A$		$P, Q, G, \mathsf{pk}_A, \mathsf{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q$, $A \leftarrow G^a$	$\xrightarrow{\text{Alice};A}$	$b \xleftarrow{\$} \mathbb{Z}_q$
		$B \leftarrow G^b$, $K_{B,A} \leftarrow A^b$
	$\xleftarrow{\text{Bob};\ B;\ C_B}$	$\sigma_B = \mathsf{Sign}_{\mathsf{sk}_B}(A, B)$, $C_B = \mathsf{Enc}_{K_{A,B}}(\sigma_B)$
$K_{A,B} \leftarrow B^a$, $\sigma_B = \mathsf{Dec}_{K_{A,B}}(C_B)$		
$\mathsf{Vf}_{\mathsf{pk}_B}(\sigma_B)$, $\sigma_A = \mathsf{Sign}_{\mathsf{sk}_A}(B, A)$		
$C_A = \mathsf{Enc}_{K_{A,B}}(\sigma_A)$	$\xrightarrow{\sigma_A}$	$\sigma_A = \mathsf{Dec}_{K_{A,B}}(C_A)$, $\mathsf{Vf}_{\mathsf{pk}_A}(\sigma_A)$

- The encryption of the signatures protects the identities!
- Hence the identities don't have to be included in the signatures.
  - If public keys are not a priori authenticated, certificates can be included in the encrypted messages.
- However, this is not a good approach.... (encryption is not for authentication!)
  - Eve can register Alice's public key as her own and again perform an ID-missbinding attack
- Proof of knowledge of $K_{A,B}$ is not enough, it should be bound to the identity of the parties

SIGMA (SIGn and MAc) protocols [Hugo Krawczyk '03]

Alice's client		Bob's server
$P, G, \mathsf{pk}_B, \mathsf{sk}_A$		$P, G, \mathsf{pk}_A, \mathsf{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q,\ A \leftarrow G^a$	$\xrightarrow{\cancel{\text{Alice};}\ A}$	$b \xleftarrow{\$} \mathbb{Z}_q$
		$B \leftarrow G^b,\ K_{B,A} \leftarrow A^b$
		$K = KDF(K_{B,A}),\ t_B = MAC_K(Bob)$
$\mathsf{Vf}_{\mathsf{pk}_B}(\sigma_B),\ K_{A,B} \leftarrow B^a$	$\xleftarrow{\text{Bob};\ B;\ \sigma_B;\ t_B}$	$\sigma_B = \mathsf{Sign}_{\mathsf{sk}_B}(A, B, \cancel{Alice})$
$K = KDF(K_{A,B}),\ t_A = MAC_K(Alice)$		
$\sigma_A = \mathsf{Sign}_{\mathsf{sk}_A}(B, A, \cancel{Bob})$	$\xrightarrow{\sigma_A;\ t_A}$	$\mathsf{Vf}_{\mathsf{pk}_A}(\sigma_A)$

- We keep the signing as in the STS protocol, since it prevents MiM attacks
- But, remove the identities from the signature, to provide repudiability
- And Alice does not advertize her identity
    - **This is basically STS, so misbinding possible!**
- Solution?:
    - **Alice and Bob MAC their own identity!**
    - From the MAC, Alice can see that Bob's identity was not replaced by Eve's

SIGMA-I protocol (initiator identity protection)

Alice's client		Bob's server
$P, G, \mathrm{pk}_B, \mathrm{sk}_A$		$P, G, \mathrm{pk}_A, \mathrm{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q,\ A \leftarrow G^a$	$\xrightarrow{A}$	$b \xleftarrow{\$} \mathbb{Z}_q,\ B \leftarrow G^b,\ K_{B,A} \leftarrow A^b,$
		$K_M; K_E = KDF(K_{B,A}),\ t_B = MAC_{K_M}(Bob),$
$K_{A,B} \leftarrow B^a,\ K_M; K_E = KDF(K_{A,B}),$	$\xleftarrow{B;\ C_B}$	$\sigma_B = \mathrm{Sign}_{\mathrm{sk}_B}(A, B),\ C_B = Enc_{K_E}(Bob, \sigma_B, t_B)$
signature and MAC verification of Bob's id		
$t_A = MAC_{K_M}(Alice),\ \sigma_A = \mathrm{Sign}_{\mathrm{sk}_A}(B, A)$		
$C_A = Enc_{K_E}(Alice, \sigma_A, t_A)$	$\xrightarrow{C_A}$	signature and MAC verification of Alice's id

*For compactness, signature and MAC verification, decryption steps omitted (but are performed)

- No identities in clear $\Rightarrow$ protection against passive attackers
- Alice can verify the identity of Bob before disclosing her own identity $\Rightarrow$ **identity protection of initiator** against active attackers
- Signing $\Rightarrow$ MiM attacks prevented
- MAC of own identity $\Rightarrow$ identity misbinding attacks prevented
- **Used in TLS 1.3 handshake**

SIGMA-R protocol (responder identity protection)

Alice's client		Bob's server
$P, G, \mathrm{pk}_B, \mathrm{sk}_A$		$P, G, \mathrm{pk}_A, \mathrm{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q,\ A \leftarrow G^a$	$\xrightarrow{A}$	$b \xleftarrow{\$} \mathbb{Z}_q,\ B \leftarrow G^b$
$K_{A,B} \leftarrow B^a,\ K_M; K_E = KDF(K_{A,B}),$	$\xleftarrow{B}$	
$t_A = MAC_{K_M}(Alice),\ \sigma_A = \mathrm{Sign}_{\mathrm{sk}_A}(B, A)$		
$C_A = Enc_{K_E}(Alice, \sigma_A, t_A)$	$\xrightarrow{C_A}$	$K_{B,A} \leftarrow A^b,\ K'_M; K'_E = KDF(K_{B,A})$
		$t_B = MAC_{K'_M}(Bob),\ \sigma_B = \mathrm{Sign}_{\mathrm{sk}_B}(A, B)$
	$\xleftarrow{C_B}$	$C_B = Enc_{K'_E}(Bob, \sigma_B, t_B)$

*For compactness, signature and MAC verification, decryption steps omitted (but are performed)

- Bob can verify the identity of Bob before disclosing his own identity $\Rightarrow$ **identity protection of responder** against active attackers
- Alice and Bob use different $K_M; K_E$ and $K'_M; K'_E$ to prevent **reflection attacks**
  - Hint: Protocol is completely symmetric!

SIGMA protocol in IKE (Internet Key Exchange) main mode

Alice's client		Bob's server
$P, G, \mathsf{pk}_B, \mathsf{sk}_A$		$P, G, \mathsf{pk}_A, \mathsf{sk}_B$
$a \xleftarrow{\$} \mathbb{Z}_q,\ A \leftarrow G^a$	$\xrightarrow{A}$	$b \xleftarrow{\$} \mathbb{Z}_q,\ B \leftarrow G^b$
$K_{A,B} \leftarrow B^a,\ K_M; K_E = KDF(K_{A,B}),$	$\xleftarrow{B}$	
$t_A = MAC_{K_M}(Alice),\ \sigma_A = \mathsf{Sign}_{\mathsf{sk}_A}(B, A, t_A)$		
$C_A = Enc_{K_E}(Alice, \sigma_A, \cancel{t_A})$	$\xrightarrow{C_A}$	$K_{B,A} \leftarrow A^b,\ K'_M; K'_E = KDF(K_{B,A})$
		$t_B = MAC_{K'_M}(Bob),\ \sigma_B = \mathsf{Sign}_{\mathsf{sk}_B}(A, B, t_B)$
	$\xleftarrow{C_B}$	$C_B = Enc_{K'_E}(Bob, \sigma_B, \cancel{t_B})$

*For compactness, signature and MAC verification, decryption steps omitted (but are performed)

**Difference from SIGMA-R in green

- IKE is core AKE protocol of IPSec – IP Security [RFC2401-12]
- above is IKE v1
- Aggressive mode exists without identity protection
- IKE v2 slight differences – A, B not included inside of MAC, only identity

Conclussions and further considerations

SIGMA protocols are part of IKE and TLS 1.3

- provably secure
- fast
- robust
- minimize number of rounds
- simpler parameters
- identity protection

Protocols have other issues (for ex. TLS is not only the handshake):

- Ciphersuite negotiation?
- Side-channel protection?
- DOS protection?
- Formal analysis?
- Key derivation?
- Use of PKI?
- Post-quantum versions?

Summary

Today:

- Authenticated Key-Exchange using Digital Signatures

Next time:

- **"Transport layer security (TLS)"** Thom Wiggers, PQShield