Authentication Under Denial-of-Service Attacks

Bettina Weiss
bw@auto.tuwien.ac.at

Vienna University of Technology
Department of Computer-Aided Automation

http://www.auto.tuwien.ac.at/Projects/W2F/

Overview

- Motivation
- Security Methods

- Client Puzzles
- Attack Possibilities
- Problems

- Message Filters
- Authenticated Communication
- Attack Possibilities

- Conclusions

$\boxed{\text{Motivation}}$

- Project W2F: design a wireline/wireless field-bus which should be fault-tolerant and secure
- Distributed algorithms like consensus profit from authentication

  $\Rightarrow$ we use authenticated algorithms

- Ad-hoc network, mobile nodes

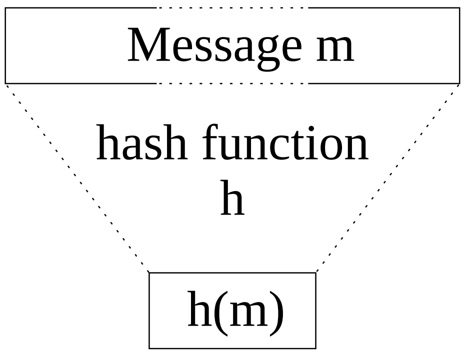
  $\Rightarrow$ denial-of-service attacks are likely

Unfortunately, authentication with signatures employing public keys is time-consuming $\Rightarrow$

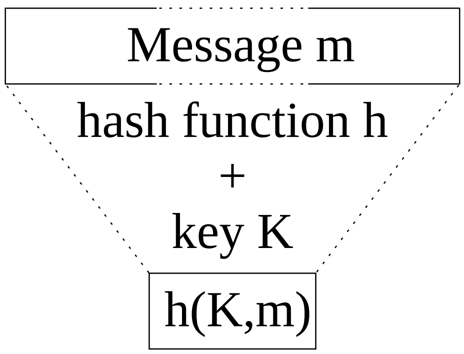
can be used for DoS attacks

Easy Security Methods

Hash Function



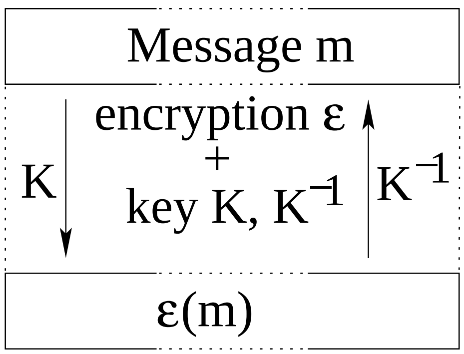
Message Authentication Code (MAC)



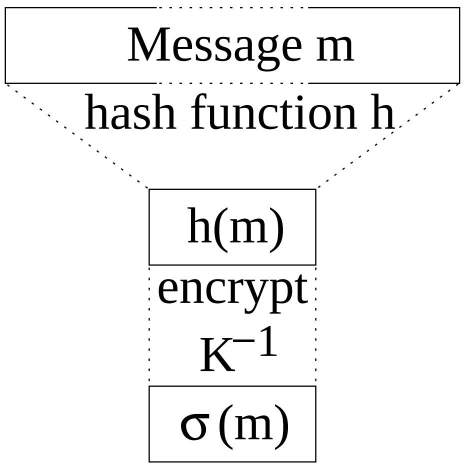
fast ($>$ 100MBit/s) $\Rightarrow$ can be done on-the-fly during message transmission/reception

Hard Security Methods

Encryption



Signatures



slow (< 1MBit/s) $\Rightarrow$ computation delays message

Client Puzzles

[Aura, Nikander and Leiwo 2000]

Idea: make attacker commit its resources first

Implementation: client must solve cryptographic problem by brute force: find X so that

$$\mathrm{h}(\mathrm{C},\ \mathrm{N}_S,\ \mathrm{N}_C,\ \mathrm{X}) = \underbrace{000\ldots000}_{k}\mathrm{Y}$$

Client		**Server**
		generate puzzle $\sigma_S(\mathrm{T_S}, \mathrm{k}, \mathrm{N_S})$
	$\xrightarrow{\text{Request}}$	
	$\xleftarrow{\sigma_S(\mathrm{T_S}, \mathrm{k}, \mathrm{N_S})}$	
compute X		
	$\xrightarrow{\sigma_C(\mathrm{S}, \mathrm{C}, \mathrm{N_S}, \mathrm{N_C}, \mathrm{X})}$	
		verify X authenticate C
	$\xleftarrow{\sigma_S(\mathrm{S}, \mathrm{C}, \mathrm{N_C})}$	
authenticate S		

Attack Possibilities

1. C $\rightarrow$ S: Request
2. S $\rightarrow$ C: $\sigma_S(\mathrm{T_S}, \mathrm{k}, \mathrm{N_S})$
3. C $\rightarrow$ S: $\sigma_C(\mathrm{S}, \mathrm{C}, \mathrm{N_S}, \mathrm{N_C}, \mathrm{X})$
4. S $\rightarrow$ C: $\sigma_S(\mathrm{S}, \mathrm{C}, \mathrm{N_C})$

---

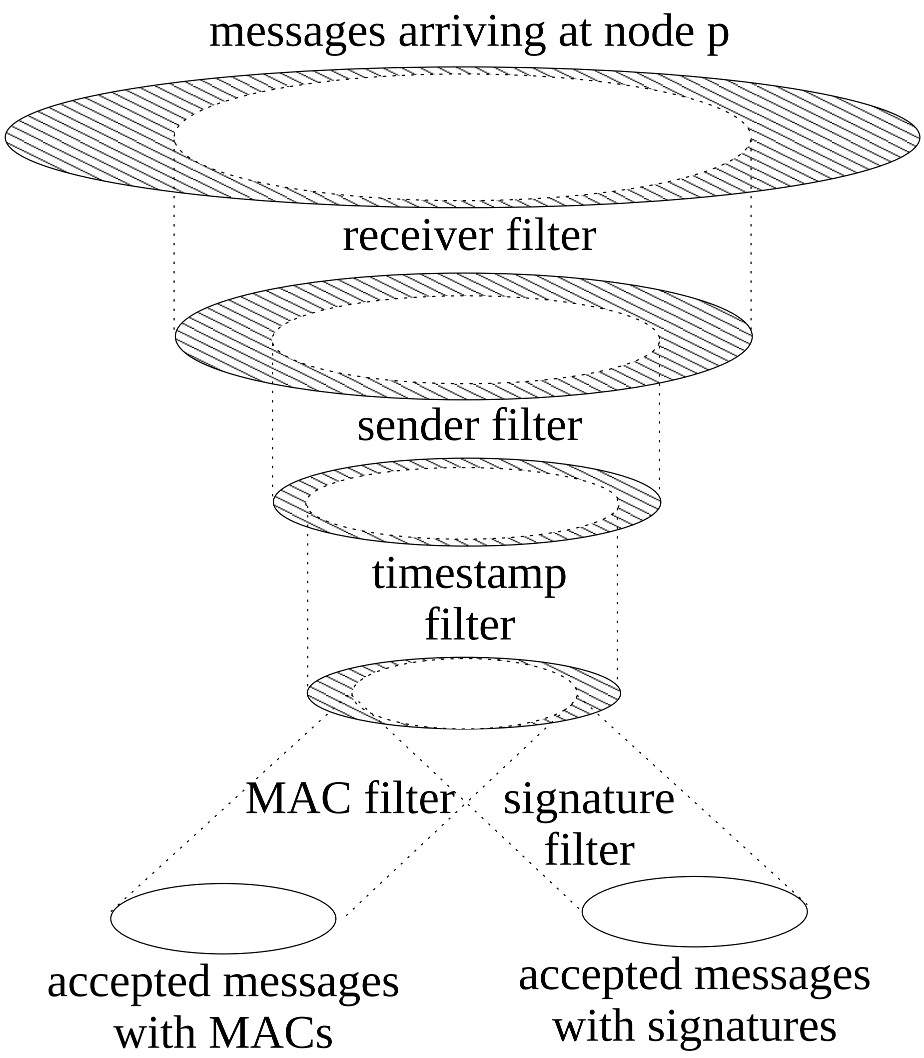
1. No attack on server
2. Attacker sends appropriate msg (plausible $\mathrm{T_S}$)
3. No attack on server
4. Attacker sends appropriate msg ($\mathrm{N_C}$)

Problems

- Mainly protects server
  $\Rightarrow$ try to protect clients even more
- Requires four messages for one request
  $\Rightarrow$ use different method for subsequent requests
- Client protection relies on synchronized clocks
  $\Rightarrow$ use fallback in case clock synchronization fails

Message Filters

Receiver, Sender, Timestamp, MAC or Signature



AUTHENTICATED COMMUNICATION

Enhancements:

- Alice sends nonce
- Sequence number in puzzle
- Shared secret key for further communication

Alice		**Bob**
		generate puzzle $\sigma_B(\mathrm{T_B}, \mathrm{k}, \mathrm{N_B}, \mathrm{S_B})$
	$[\mathrm{B, A, T^s_A, T^r_B}], \mathrm{N_A}$, `puzzle request` $\longrightarrow$	
	$\longleftarrow$ $[\mathrm{A, B, T^s_B, T^r_A}], \mathrm{N_A}, \sigma_B(\mathrm{T_B}, \mathrm{k}, \mathrm{N_B}, \mathrm{S_B})$	
compute X		
	$\sigma_A([\mathrm{B, A, T'^s_A, T'^r_B}], \mathrm{N_B}, \mathrm{N'_A}, \mathrm{X}, \epsilon_\mathrm{B}(\mathrm{K_{AB}}), \mathrm{Data_A})$ $\longrightarrow$	
		verify X authenticate Alice
	$\longleftarrow$ $\sigma^h_B([\mathrm{A, B, T'^s_B, T'^r_A}], \mathrm{N'_A}, \mathrm{Data_B})$	
authenticate Bob		

$\boxed{\textsc{Attack Possibilities}}$

1. $\mathrm{A} \rightarrow \mathrm{B}$: $[\mathrm{B}, \mathrm{A}, \mathrm{T}^{\mathrm{s}}_{\mathrm{A}}, \mathrm{T}^{\mathrm{r}}_{\mathrm{B}}], \mathrm{N}_{\mathrm{A}}$, `puzzle request`
2. $\mathrm{B} \rightarrow \mathrm{A}$: $[\mathrm{A}, \mathrm{B}, \mathrm{T}^{\mathrm{s}}_{\mathrm{B}}, \mathrm{T}^{\mathrm{r}}_{\mathrm{A}}], \mathrm{N}_{\mathrm{A}}, \sigma_B(\mathrm{T}_{\mathrm{B}}, \mathrm{k}, \mathrm{N}_{\mathrm{B}}, \mathrm{S}_{\mathrm{B}})$
3. $\mathrm{A} \rightarrow \mathrm{B}$: $\sigma_A([\mathrm{B}, \mathrm{A}, \mathrm{T}'^{\mathrm{s}}_{\mathrm{A}}, \mathrm{T}'^{\mathrm{r}}_{\mathrm{B}}], \mathrm{N}_{\mathrm{B}}, \mathrm{N}'_{\mathrm{A}}, \mathrm{X}, \epsilon_{\mathrm{B}}(\mathrm{K}_{\mathrm{AB}}), \mathrm{Data}_{\mathrm{A}})$
4. $\mathrm{B} \rightarrow \mathrm{A}$: $\sigma^h_B([\mathrm{A}, \mathrm{B}, \mathrm{T}'^{\mathrm{s}}_{\mathrm{B}}, \mathrm{T}'^{\mathrm{r}}_{\mathrm{A}}], \mathrm{N}'_{\mathrm{A}}, \mathrm{Data}_{\mathrm{B}})$

---

1. No attack on Bob
2. Attacker sends appropriate msg (N_{A}, plausible T_{B}, plausible S_{B})
3. No attack on Bob
4. No attack on Alice

CONCLUSIONS

- Algorithm for authenticated communication
- Client puzzle for first contact, MAC for remainder of communication
- Better protection of initiator of first contact through MAC
- Hardware message filters for discarding wrong messages as soon as possible

Future Work

- Better algorithm for first contact?
- Formal Analysis
- Comparison of costs to attacker and defender [Meadows 2001]