



On handling cryptographic bottlenecks

by
Yacov Yacobi
Microsoft

Presentation of the RSA Conference
January 1997

Contents

- Computational bottlenecks
 - Algorithms
 - Protocols
 - * Clients: Back to Back EG-RSA,
 - * Servers: Signed hash trees,
- Architectural
 - Local CRL's,
 - Randomized audit.

Algorithms & Complexity

	Time		Space			
	Verify	Sign $T_s =$	Signature	Secret key	Scratch	
RSA	$T_v = 1\text{ms}$	30ms	$S = 512\text{bit}$	2s	a	
EG	T_s	On:off= $T_v/2:T_s$	S	2S	a	
Micali	$40T_v$	On:off= $T_s/40:T_s/20$	S	2S	a	
FS	$T_v/40$	$T_s/20$	S	160S	160a	

Numbers are for 512 bit secret.

VRA

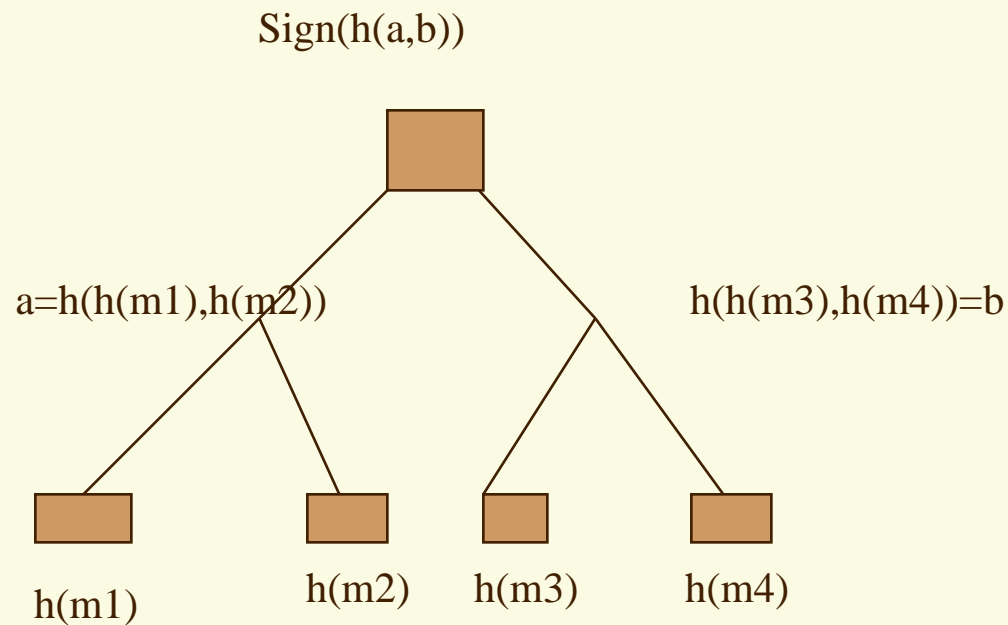
- Load a table with your favorable algorithm, (e.g. DES³),
- Run VRA at 6 machine cycles/block encryption,
- It is as strong as DES³.

Back to Back RSA/EG

	Secret	public
RSA	750	2
EG	1 on-line	750

Put the weak player on public=RSA, secret=EG.

Signed hash trees



A signature on individual message is the signed root+the path. Since hashing is 10,000 times faster than signature the cost of a signature is amortized over the batch.

Local CRL's

- Long (2 years) + short expirations (1 month),
- Store short hashes of revocations,
- 30M users, 4% annual revocations ==>
|CRL|=1 Mbyte
- Use broadcast channels for update revocations
(~1 in 25 seconds).
- This is as close to off-line as possible.