CMPSCI 677 Operating Systems **Spring 2017**

Lecture 25: April 30

Lecturer: Prashant Shenoy *Scribe:* ***Daniel Saunders***

25.1 Distributed Security

25.1.1 Authentication using public keys

AP 4.0 uses symmetric keys for authenticantion. **Question**: can we use public keys? *symmetry*: DA(EA(n)) = EA(DA(n)).

AP 5.0:
A to B: msg = "I am A"
B to A: once in a lifetime value n
A to B: msg = DA(n)
B computes: If EA(DA(n)) = n
then A is verified
else A is fradulent

25.1.2 Man-in-the-middle attack

Trudy impersonates as Alice to Bob and as Bob to Alice.

Alice		Trudy		Bob
	"I am A"		"I am A"	
			nonce n	
			DT(n)	
			send me ET	
	nonce n			
	DA(n)			
	send me EA			
	EA			

Bob sends data using ET, and Trudy decrypts and forwards it using EA (Trudy *transparently* intercepts every message).

25.1.3 Digital signatures using public keys

Goals of digital signatures:

- Sender cannot repudiate message never sent ("I never sent that").

- Receiver cannot fake a received message.

Suppose A wants B to "sign" a message M.

B send DB(M) to A
A computes if EB(DM(A)) = M
then B has signed M

Question: Can B plausibly deny having sent M?

25.1.4 Message digests

Encypting and decrypting entire messages using digital signatures is computationally expensive. Routers routinely exchange data, which do not need encryption, but do require authentication and to verify that data hasn't changed.

A message digest is a compact summary of a message, like a checksum. A has function H converts a variable length string to a fixed length hash value. The user will then digitally sign $H(M)$, and send both M and $DA(H(M))$. The receiver can verify who sent the message and that it has been changed.

Important property of H: Given a digest x, it is infeasible to find a message y for which $H(y) = x$. Also, it is infeasible to find any two messages such that $H(x) = H(y)$ (hash collision).

25.2 Bitcoin