CSE 484 (Winter 2010)

Symmetric Cryptography

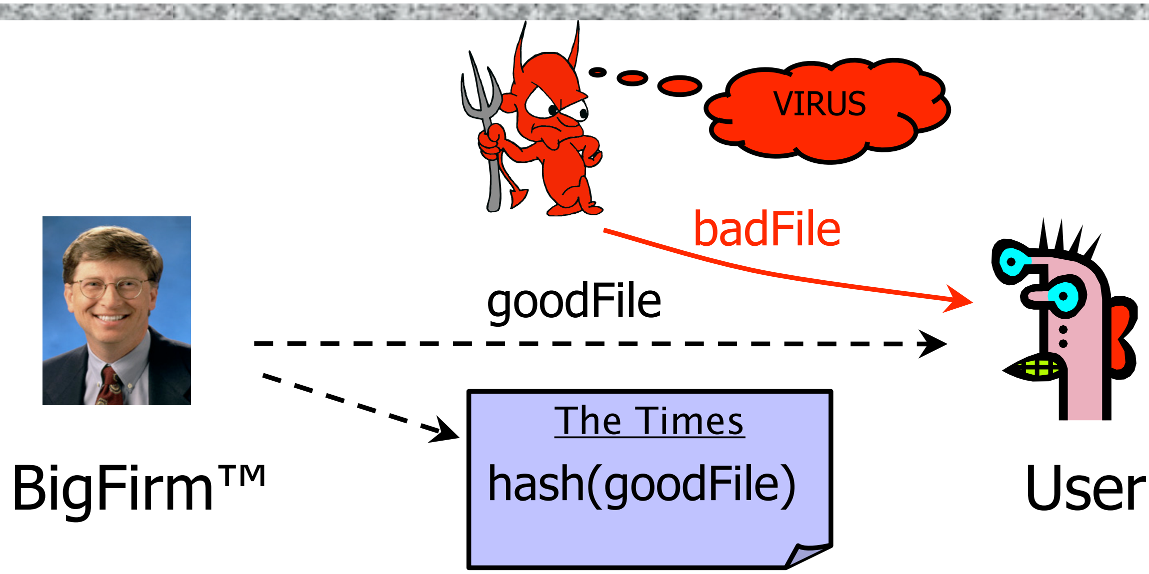
Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

Goals for Today

- Under the hood: Hash functions and MACs
- CELT -- Confidential course feedback opportunity

Integrity



Software manufacturer wants to ensure that the executable file is received by users without modification.

It sends out the file to users and publishes its hash in NY Times.

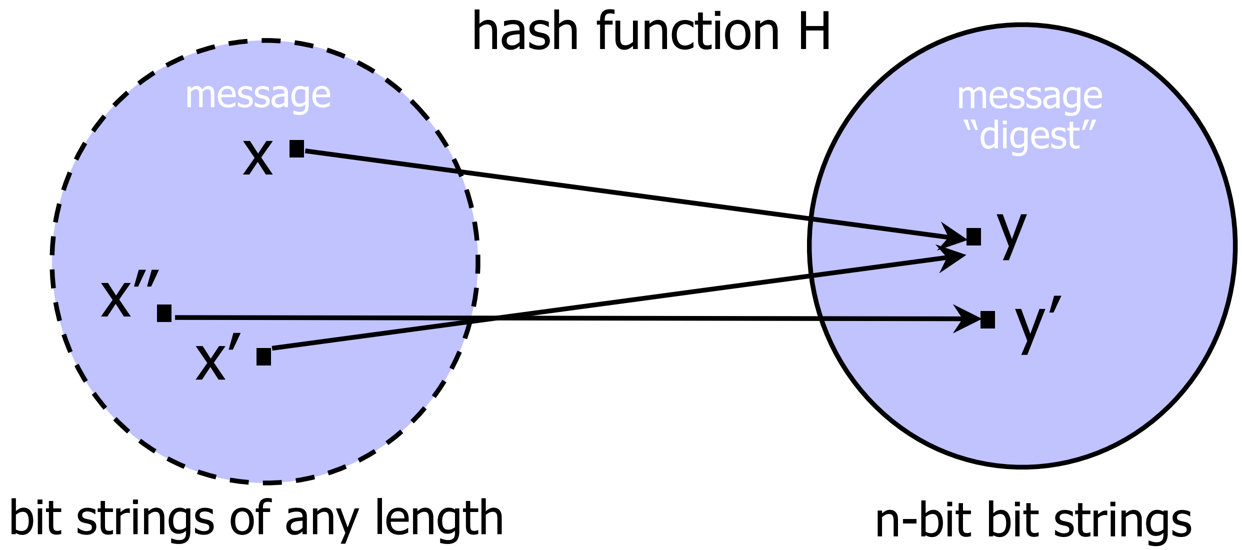
The goal is integrity, not secrecy

Idea: given goodFile and hash(goodFile), very hard to find badFile such that hash(goodFile)=hash(badFile)

Integrity vs. Secrecy

- Integrity: attacker cannot tamper with message
- Encryption does not always guarantee integrity
  - Intuition: attacker may able to modify message under encryption without learning what it is
    - One-time pad: given key K, encrypt M as $M \oplus K$
    - This guarantees perfect secrecy, but attacker can easily change unknown M under encryption to $M \oplus M'$ for any M'
    - Online auction: halve competitor's bid without learning its value
  - This is recognized by industry standards (e.g., PKCS)
    - "RSA encryption is intended primarily to provide confidentiality… It is not intended to provide integrity" (from RSA Labs Bulletin)

Hash Functions: Main Idea



- H is a lossy compression function
  - Collisions: h(x)=h(x') for distinct inputs x, x'
  - Result of hashing should "look random" (make this precise later)
    - Intuition: half of digest bits are "1"; any bit in digest is "1" half the time
- Cryptographic hash function needs a few properties…

One-Way

- Intuition: hash should be hard to invert
  - "Preimage resistance"
  - Let $h(x')=y\in\{0,1\}^n$ for a random x'
  - Given y, it should be hard to find any x such that h(x) =y
- How hard?
  - Brute-force: try every possible x, see if h(x)=y
  - SHA-1 (common hash function) has 160-bit output
    - Expect to try 2^{159} inputs before finding one that hashes to y.

Collision Resistance

- Should be hard to find distinct x, x′ such that h(x)=h(x′)
  - Brute-force collision search is only $O(2^{n/2})$, not $O(2^n)$
  - For SHA-1, this means $O(2^{80})$ vs. $O(2^{160})$
- Birthday paradox (informal)
  - Let t be the number of values x,x′,x″… we need to look at before finding the first pair x,x′ s.t. h(x)=h(x′)
  - What is probability of collision for each pair x,x′? $1/2^n$
  - How many pairs would we need to look at before finding the first collision? $O(2^n)$
  - How many pairs x,x′ total? $\text{Choose}(t,2)=t(t-1)/2 \sim O(t^2)$
  - What is t? $2^{n/2}$

One-Way vs. Collision Resistance

- One-wayness does <u>not</u> imply collision resistance
  - Suppose g is one-way
  - Define h(x) as g(x’) where x’ is x except the last bit
    - h is one-way (to invert h, must invert g)
    - Collisions for h are easy to find: for any x, h(x0)=h(x1)
- Collision resistance does <u>not</u> imply one-wayness
  - Suppose g is collision-resistant
  - Define h(x) to be 0x if x is n-bit long, 1g(x) otherwise
    - Collisions for h are hard to find: if y starts with 0, then there are no collisions, if y starts with 1, then must find collisions in g
    - h is not one way: half of all y’s (those whose first bit is 0) are easy to invert (how?); random y is invertible with probab. 1/2

Weak Collision Resistance

- Given randomly chosen x, hard to find x′ such that h(x)=h(x′)
  - Attacker must find collision for a specific x. By contrast, to break collision resistance, enough to find any collision.
  - Brute-force attack requires $O(2^n)$ time
  - AKA second-preimage collision resistance
- Weak collision resistance does not imply collision resistance

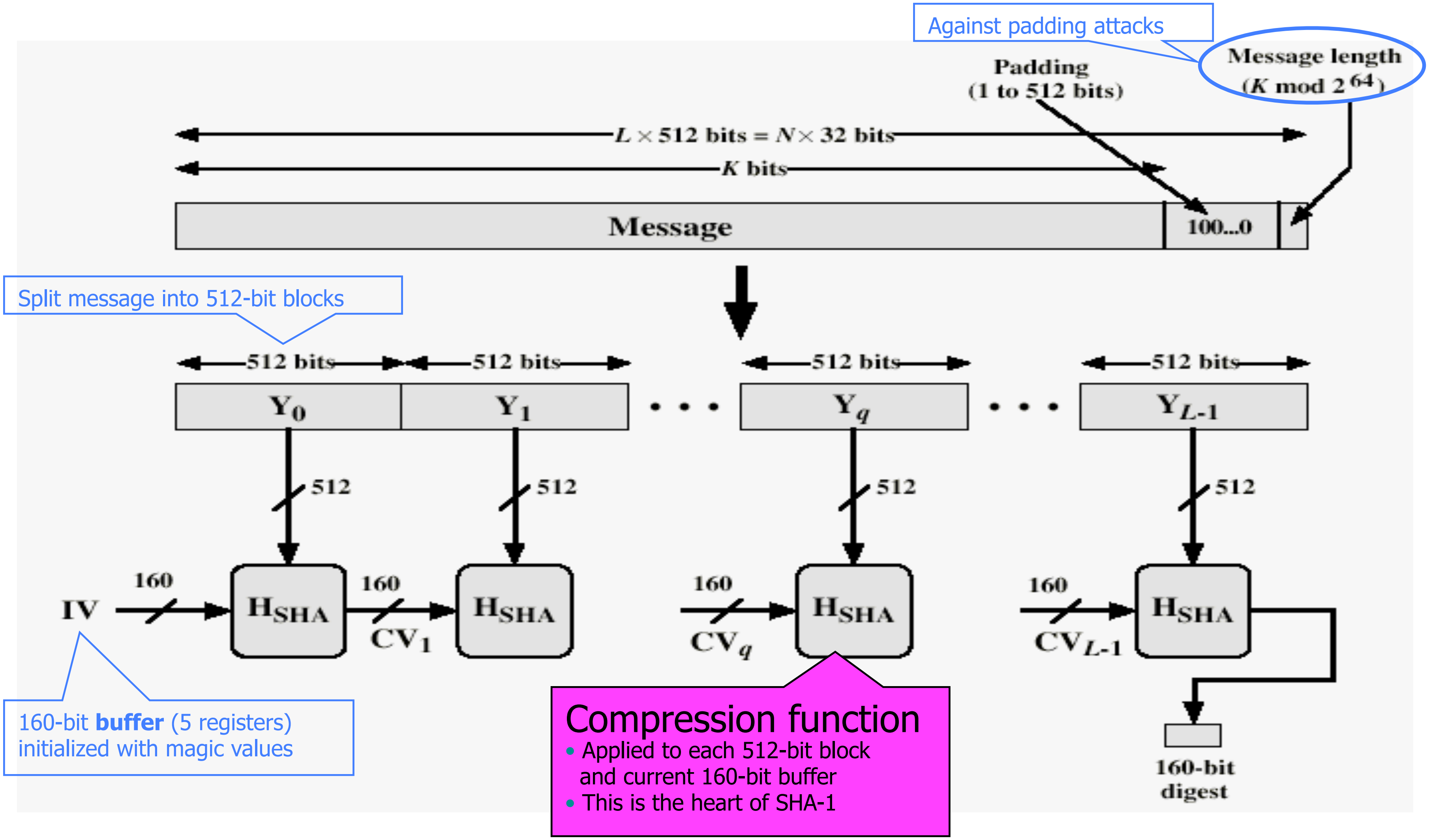
Which Property Do We Need?

- UNIX passwords stored as hash(password)
  - One-wayness: hard to recover password
  - Second-preimage resistance: hard to recover "equivalent" passwd
- Integrity of software distribution
  - Weak collision resistance
  - But software images are not really random… maybe need full collision resistance
- Auction bidding
  - Alice wants to bid B, sends H(B), later reveals B
  - One-wayness: rival bidders should not recover B
  - Collision resistance: Alice should not be able to change her mind to bid B' such that H(B)=H(B')

Common Hash Functions

- MD5
  - 128-bit output
  - Designed by Ron Rivest, used very widely
  - Collision-resistance broken (summer of 2004)
- RIPEMD-160
  - 160-bit variant of MD5
- SHA-1 (Secure Hash Algorithm)
  - 160-bit output
  - US government (NIST) standard as of 1993-95
  - Also recently broken! (Theoretically -- not practical.)
- SHA-256, SHA-512, SHA-224, SHA-384
- SHA-3: Forthcoming.

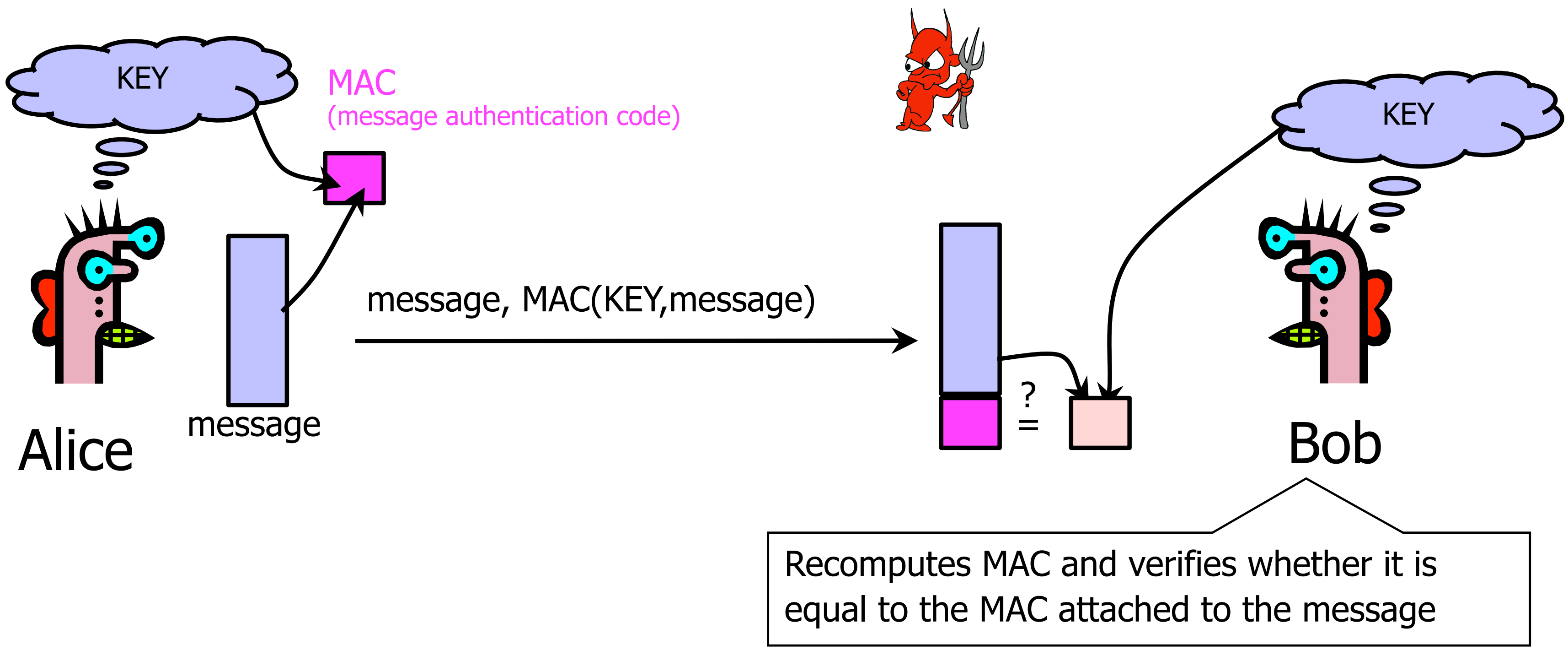
Basic Structure of SHA-1 (Not Required)



How Strong Is SHA-1?

- Every bit of output depends on every bit of input
  - Very important property for collision-resistance
- Brute-force inversion requires 2^{160} ops, birthday attack on collision resistance requires 2^{80} ops
- Some very recent weaknesses (2005)
  - Collisions can be found in 2^{63} ops

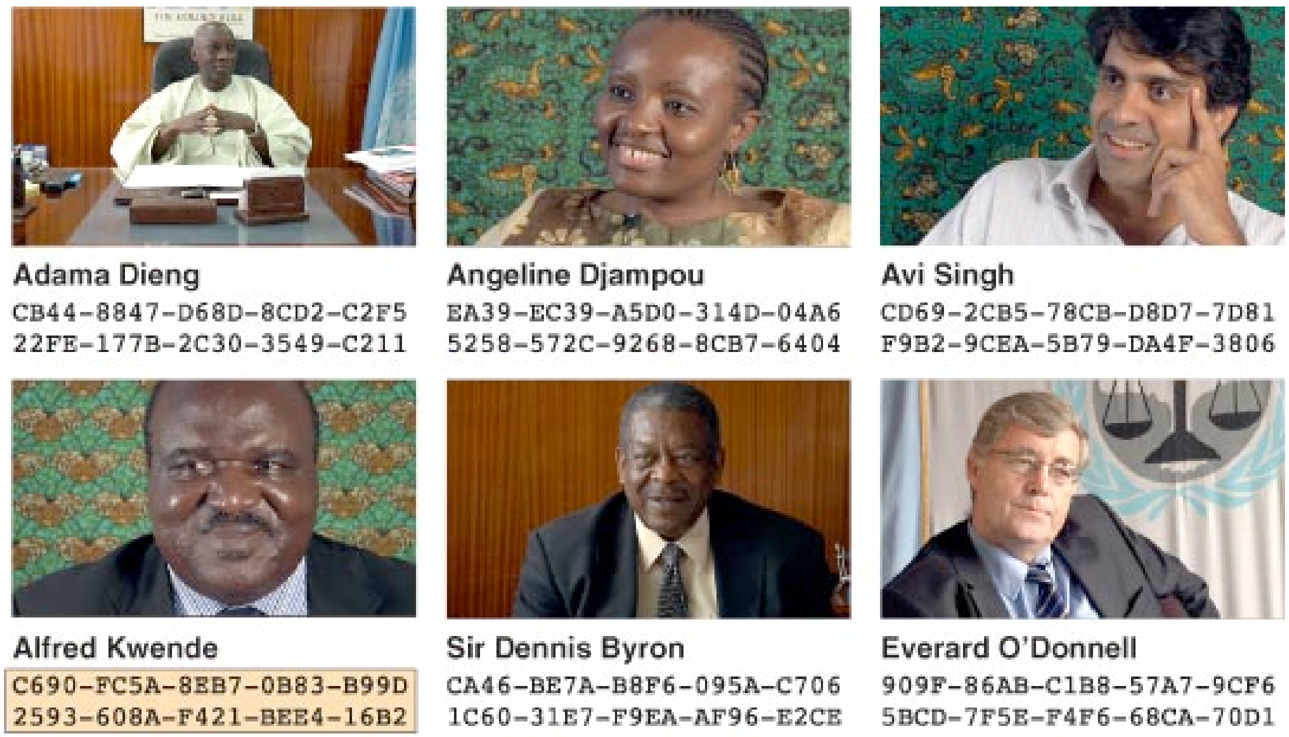
Authentication Without Encryption



Integrity and authentication: only someone who knows KEY can compute MAC for a given message

International Criminal Tribunal for Rwanda

- http://www.nytimes.com/2009/01/27/science/27arch.html?_r=1&ref=science



- Credits: Alexei Czeskis, Karl Koscher, Batya Friedman

HMAC

- Construct MAC by applying a cryptographic hash function to message and key
  - Could also use encryption instead of hashing, but…
  - Hashing is faster than encryption in software
  - Library code for hash functions widely available
  - Can easily replace one hash function with another
  - There used to be US export restrictions on encryption
- Invented by Bellare, Canetti, and Krawczyk (1996)
  - HMAC strength established by cryptographic analysis
- Mandatory for IP security, also used in SSL/TLS

Structure of HMAC