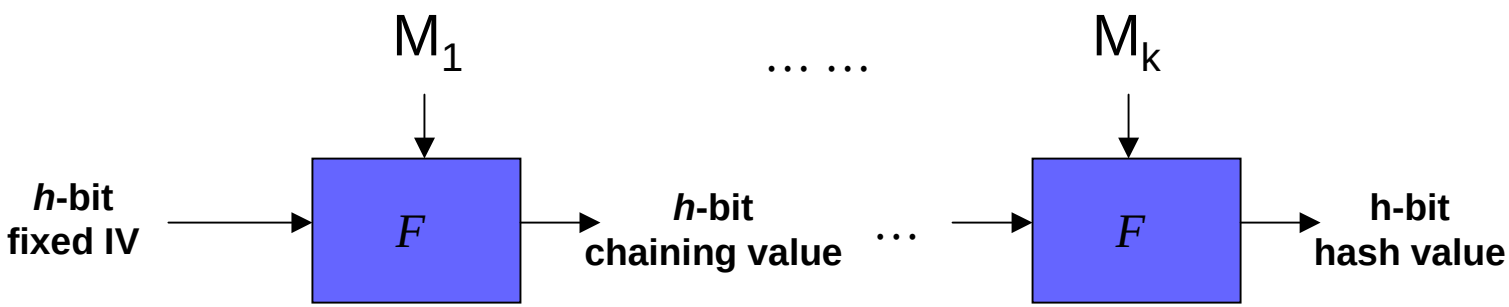
NIST Hash Function Standards Status and Plans

Dec. 6, 2005

Bill Burr
Manager, Security Technology Group
NIST
willliam.burr@nist.gov

Merkle-Damgard Hash Functions

- Take a long message, break it into blocks (typ. 512 bits)
  - M_1, M_2, M_3…M_k (pad out last block)
- Let *F* be a “compression function” that operates on a block and the current *h*-bit state and “mixes” the block into the state
- Last output of compression function is the *h*-bit hash value or message digest.



Hash Function Applications

- Digital Signatures
  - Message digest represents message; private key applied to digest
  - Proof to 3rd party is where most problems with collisions occur
- Keyed hashes
  - Random number generation
    - H(counter||masterkey)
  - Message authentication codes
- One way transformations
  - Protect passwords
    - H(password||username||salt)
  - Forward secrecy
- Key derivation
  - Mix entropy up and shrink
    - H(mastersecret||“encryptkey”||clientname||hostname||random)

Hash Function Properties

- Collision resistant
  - Can’t find any two different messages with the same hash value
- One Way
  - Given only a hash value, can’t construct a message (or “preimage”) that generates the hash. An attack that generates a second message with the same hash value as a given message is called a “second pre-image” attack.

Finding Hash Collisions

- Find two messages with the same digest
- Birthday “paradox”
  - Given a population of *n* equally probable values, we need roughly $\sqrt{n}$ random samples to expect to find a single collision
- Therefore any attack that finds a collision in much under $2^{n/2}$ operations is said to “break” the collision resistance property of the hash function

Finding Preimages

- Work backward from message digest to find a message that will produce it
- Expect to have to hash about 2^n messages to find an unknown pre-image for any particular selected message digest value
  - Any attack that finds a preimage in significantly under 2^n operations is a break of the one-way property or preimage resistance of a hash function.

Digital Signatures

Digital signatures are perhaps the most demanding of the many applications for hash functions

- Hash the message, then apply private key to the hash to generate the signature

- Potentially subject to collision attacks and second pre-image attacks
- Collisions must be found before the signature is applied
  - Can't do a collision attack on old signed messages
- Second pre-image attack can be done any time after the message is signed.

Signature Collision Attack

- Find 2 messages with opposite meanings and the same digest value
  - I agree that…
  - I do not concur…
- Sign one, then repudiate the signature by claiming that you signed the other
- Collisions have to be found *before you sign*
- Doesn't help to forge a signature with an unknown private key
- Should require $2^{n/2}$ work
  - By best current estimates SHA-1 gives about 63-bit security against collisions
    - *It was supposed to be* 2^{80}

Signature Second Preimage Attack

- Take a signed message and find a second message with the same message digest (the second preimage)
- You have just forged a signature for the second message
- Much harder than collision attack
  - 2^n versus $2^{n/2}$ operations
  - For SHA-1 about 160-bit security against a second preimage
- Can do any time after the first signature is created

Attack Summary

- Collision attack
  - Allows signer to repudiate signature
  - Must do before signing
  - $2^{n/2}$ operations – comparatively easy (but we make hashes big enough that it's still should be impractical)
- Second preimage attack
  - Allows anybody to forge a signature
  - Can do anytime after first signature
  - 2^{n} operations – comparatively hard
- We don't want to allow either one

Currently Used Hash Functions

- Only two in wide use in US today
  - MD5
    - Invented by Ron Rivest circa 1992
    - 128-bit hash
    - “Almost broken” by Hans Dobbertin circa 1995
    - Fully broken by collision attack Wang *et. al.* 2004
  - SHA-1
    - Developed by NSA circa 1995
    - “Apparently minor” revision of SHA-0
    - 160-bit hash
    - Broken Feb. 2005 by Xiaouyan Wang

MD5

- NIST never felt 128-bits was enough for a digital signature, so never adopted MD5
- “Nearly broken” in 1995 by Hans Dobbertin
  - Found collisions in the compression function itself
  - We were warned:
    - “The presented attack does not yet threaten practical applications of MD5, but it comes rather close… Therefore we suggest that in the future MD5 should no longer be implemented in applications like signature schemes where a collision-resident hash function is required.”
      - *Cryptobytes* Summer 1996

SHA-1

- FIPS 180-1, 160-bit message digest
- Compression function has an initial block expansion and 80 “rounds” of mixing
- SHA-1derived from SHA-0
  - Apparently minor revision: adds a rotate to the initial block expansion
    - This turns out to block recent differential hash collisions attacks
- NIST has planned for several years to end federal use of SHA-1 by end of 2010 in favor of SHA-256, to forestall future brute force collision attacks

Recent Hash Breaks

- August 2004: Eli Biham, Fari Chen, Antoine Joux, Xiaoyun Wang, Xuejia Lai, Dengguo Feng, and Hongbo Yu presented successful full collision attacks on MD4, MD5, HAVAL-128, HAVAL-160, RIPEND and SHA-0 at Crypto 2004.
  - Of these only MD5 is widely used in the US today
  - ***All these algorithms are broken now*** and should not be used to generate signatures
- Feb. 2005 Prof. Wang announced she can find SHA-1 collisions with 2^{69} work – it should be 2^{80}
  - Her current estimate is 2^{63}
  - This is still a lot of computation for a collision, ***but not as much as we want***

NIST Hash Workshop

- Oct 31 – Nov. 1
  - http://www.csrc.nist.gov/pki/HashWorkshop/index.html
- About 180 attendees
- Status of attacks on SHA-1 & SHA-2
  - Generic attacks on all MD hashes
  - SHA-1 & SHA-2
    - Impacts and workarounds
    - How deadly and much farther will they go?
- New Designs and design criteria
- Where do we want to go from here?
  - How hard is it to change?
  - How soon is it needed?
  - What are the requirements for a new hash standard?

Workshop Summary: SHA-1 Collisions

- Current best estimate 2^{63}
  - Still a fair amount of work
    - How much farther will it go?
  - Would be nice to verify this result
    - May be dangerous to do so
- How important are collisions? Two extremes:
  - Relatively minor, only matter for rare instances where we have to prove to a 3rd party (e.g. PKI - but PKI is a failure anyhow), or;
  - Canary in the mineshaft, crack in the dyke – a warning of much bigger dangers close at hand

Workshop Summary: SHA-1 Policy

- Getting rid of MD5 is highest priority
- OK to continue using SHA-1 a few more years in old apps (really have to) but new apps must use something else (SHA2?)
  - But we don't want apps to roll their own crypto
    - SHA2 support doesn't arrive until Vista
      - Long tail to XP
  - Can't issue only SHA2 certs (if you believe PKI still lives) until clients can do SHA2

Workshop Summary: SHA2

- Very little analysis yet - rather complex
- May well be theoretical break within a decade
- Probably won't be a practical attack within a decade
- Not very efficient in hardware
- Can fix problems with more rounds
  - Need to be more conservative with number of rounds generally (think block cipher)
- Does NIST have a choice for relatively near term?

Workshop Sum: General Observations

- MD hash as random oracle => trouble
- Algorithm agility is needed
  - Resilience: several hash standards
- But: algorithm agility “sucks” in hardware
- So: we should overbuild
- But: everybody pays all the time for that

Workshop Summary: The Future

- Still confused about what all we want
- Beyond MD: block “generic attacks”
- Maybe we need more specialized functions
  - MACs, Digital Signatures, PRFs, KDF?
- Better design
  - Higher hamming weights
  - Better compression functions
- Provable security?
  - Number theoretic or equivalent to breaking something?
- Improve protocols to rely less on hash properties

Future Hash Standard Strategy

- For reasonably long term, not a crash program
  - Still discussing requirements/criteria
  - Not as mature as block cipher design in late 90s
- Flesh out requirements & criteria
  - additional workshop(s) ; competition for competition?
  - Tag the next onto Crypto2006?
- Competition
  - Probably 2 stages as with AES
- Selection
  - How many?

Bottom line

- Collisions facilitate repudiation but not forgery
- Take this seriously:
  - Don’t use MD5 for signatures
- SHA-1 not as badly broken but needs to be replaced
  - End use of SHA-1 and 1024 RSA by 2010
  - Stop issuing new certificates with SHA-1 by 2008
- NIST plans to phase out all 80-bit crypto by end of 2010
- FIPS 180-2 already in place with SHA-224, SHA-256, SHA-384 and SHA-512
  - Not much public analysis of these algorithms yet
  - Hash work shop at NIST Oct. 31 - Nov. 1
    - Near and long term response

The Future – not just hash functions

- Stop using 80-bit equivalent crypto by 2010
  - Don't rely on 2key TDEA, SAH-1 (for signatures), 160-bit ECDSA, r 1024-bit RSA, 1024bit DSA, 1024-bit Diffie-Hellman after Dec 31 2010
- 112-bit crypto should be good until 2030
- After 2030 use 128-bit strength crypto
  - Hard to say what the real date will be this far in the future
  - Quantum computing could change all this
    - Probably not a big impact on hashes

Questions ?

NIST
National Institute of Standards and Technology