

DAKOTA

– hashing from a combination of modular arithmetic and symmetric cryptography

Lars R. Knudsen

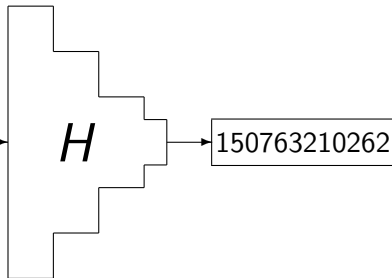
January 11, 2008

DaKoTa

- DaKoTa, a hash function co-designed by
- Damgård, Ivan B.
- Knudsen, Lars R.
- Thomsen, Søren S.

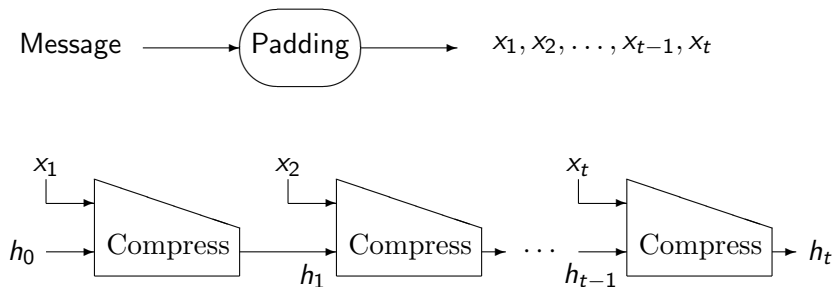
Definition - hash function

Aboriginal settlers arrived on the continent from Southeast Asia about 40,000 years before the first Europeans began exploration in the 17th century. No formal territorial claims were made until 1770, when Capt. James Cook took possession in the name of Great Britain. Six colonies were created in the late 18th and 19th centuries; they federated and became the Commonwealth of Australia in 1901. The new country took advantage of its natural resources to rapidly develop agricultural and manufacturing industries and to make a major contribution to the British effort in World Wars I and II. In recent



$$H : \{0, 1\}^* \rightarrow \{0, 1\}^n, \text{ for fixed value of } n$$

Iterated hash functions



Damgård, Merkle, 89: collision for Hash \Rightarrow collision for compress

History - brief

- 1980s: Hash functions based on block ciphers
- 1990s:
 - Dedicated, faster hash functions (kicked off by Rivest)
 - Many broken block cipher based hash functions
- 2000s:
 - Many dedicated schemes broken
 - Many new constructions, alternatives to Damgård-Merkle?
- Designs with proofs of security ?

Based on number theory

■ Factoring

- $n = pq$, $p \neq q$, large odd primes, α fixed, large order mod n .
- $H(x) = \alpha^x \bmod n$
- Collision: $H(x) = H(x') \Rightarrow x - x' = k\phi(n)$.

■ Discrete log

- Public primes: $p, q = \frac{p-1}{2}$, s.t. DLP(p) is hard
- Public primitive elements of Z_p : α, β (randomly chosen)
- $h(x, y) = \alpha^x \beta^y \bmod p$
- Collision for $h \Rightarrow$ compute $\log_\alpha(\beta)$

“GMR”

- Based on GMR 1984
- $n = pq$, $p \neq q$, large primes, a_0, a_1 squares modulo n
- Public: n, a_0, a_1 $h : \{0, 1\} \times \text{SQ}(n) \rightarrow \text{SQ}(n)$

$$h(b, y) = a_b y^2 \bmod n$$

- Collision gives “ $\sqrt{a_0 a_1^{-1}}$ ” \rightarrow factoring
- More efficient variants with more squares a_0, \dots, a_k , Damgård 87

Number-theoretic hash functions

- most schemes slow, e.g., no real speed-up for use in digital signature schemes
- some schemes have unfortunate algebraic properties (may interact badly with other public-key algorithms)
- open problem to devise efficient “provably” secure hash function

Newer constructions

- VSH - Very Smooth Hash
 - Contini, Lenstra, Steinfeld, 2005
 - collision \Rightarrow nontrivial modular square roots of very smooth numbers modulo N (composite)
 - efficient collision finder implies fast factoring algorithm
- LASH - A Lattice Based Hash Function
 - Bentahar, Page, Saarinen, Silverman, Smart 2006
 - based on the problem of finding small vectors in lattices

VSH - iterated hash function

- Let $N = pq$ be a public RSA modulus ($p \neq q$, both secret)
- Let p_1, \dots, p_k be public primes such that $\prod_{i=1}^k p_i < N$
 - Let $m = m_1, m_2, \dots, m_{\ell k}$ be message, $m_i \in \{0, 1\}$
 - $x_0 = 1$
 - $x_1 = x_0^2 (p_1^{m_1} p_2^{m_2} \dots p_k^{m_k}) \bmod N$
 - $x_{j+1} = x_j^2 \prod_{i=1}^k p_i^{m_{jk+i}} \bmod N$
 - $\text{Hash}(m) = x_\ell$

Getting to DAKOTA

- $n = pq$, $p \neq q$, large primes, $p \equiv q \equiv 3 \pmod{4}$
- Public: n, a_0, a_1
- $h : \{0, 1\} \times \text{SQ}(n) \rightarrow \text{SQ}(n)$

$$h(b, y) = a_b y^2 \pmod{n}$$

Getting to DAKOTA

- $n = pq$, $p \neq q$, large primes, $p \equiv q \equiv 3 \pmod{4}$
- Public: n, f
- $h : \{0, 1\} \times \text{SQ}(n) \rightarrow \text{SQ}(n)$
- $f : \{0, 1\} \rightarrow \text{SQ}(n)$

$$h(b, y) = f(b) y^2 \pmod{n}$$

Getting closer to DAKOTA

- $n = pq$, $p \neq q$, large primes, $p \equiv q \equiv 3 \pmod{4}$
- Public: n, f
- $h : \{0, 1\}^k \times \text{SQ}(n) \rightarrow \text{SQ}(n)$
- $f : \{0, 1\}^k \rightarrow \text{SQ}(n)$

$$h(x, y) = f(x) y^2 \pmod{n}$$

Getting even closer to DAKOTA

- $n = pq$, $p \neq q$, large primes, $p \equiv q \equiv 3 \pmod{4}$
- Public: n, f
- $h : \{0, 1\}^k \times \text{SQ}(n) \rightarrow \text{SQ}(n)$
- $f : \{0, 1\}^k \rightarrow \mathbf{Z}_n$

$$h(x, y) = f(x)^2 y^2 \pmod{n}$$

Arriving at DAKOTA

- $n = pq$, $p \neq q$, large primes, $p \equiv q \equiv 3 \pmod{4}$
- Public: n, f
- $h : \{0, 1\}^k \times \text{SQ}(n) \rightarrow \text{SQ}(n)$
- $f : \{0, 1\}^k \rightarrow \mathbf{Z}_n$

$$h(x, y) = (f(x) y)^2 \pmod{n}$$

DAKOTA- an iterated hash function

- $n = pq$, $p \equiv q \equiv 3 \pmod{4}$, public: n, f

- $h : \{0, 1\}^k \times \text{SQ}(n) \rightarrow \text{SQ}(n)$ $f : \{0, 1\}^k \rightarrow \mathbf{Z}_n$

$$h(x, y) = (f(x) y)^2 \pmod{n}$$

- Choose $r \in \mathbf{Z}_n^*$, let $s = r^2 \pmod{n}$
- Split padded message x into k -bit words, x_1, \dots, x_t
- Set $y_0 = s$, then compute

$$y_i = h(x_i, y_{i-1}) = (f(x_i) y_{i-1})^2 \pmod{n}$$

- Hash of x is then y_t .

DAKOTA- an iterated hash function

$$h(x, y) = (f(x)y)^2 \bmod n$$

Assumption

Consider probabilistic polynomial time algorithm with input f, n , and output x, \tilde{x}, z . Probability is negligible that

$$x \neq \tilde{x} \quad \text{and} \quad f(x)/f(\tilde{x}) = \pm z^2 \bmod n$$

Theorem

Hash function H is collision intractable under Assumption

find collision with prob $\epsilon \rightarrow$ break Assumption with prob $\epsilon/2$.

DAKOTA- Assumption

$$h(x, y) = (f(x)y)^2 \bmod n$$

Assumption

Consider probabilistic polynomial time algorithm with input f, n , and output x, \tilde{x}, z . Probability is negligible that

$$x \neq \tilde{x} \quad \text{and} \quad f(x)/f(\tilde{x}) = \pm z^2 \bmod n$$

- f must be one-way: choose z, \tilde{x} , compute x
- f must be coll. resistant: find collision for f , let $z = 1$
- no circular argument?, since f does (need to) not compress
- $f(x) = x^2 \bmod n$ for $x > n/2$?? No! $x = z\tilde{x}$ breaks it
- $f(x) = x^3 \bmod n$
- $f(x) = x^2 + c \bmod n$ for $x > n/2$?? $f(x) = x^3 + c \bmod n$

DAKOTA- Proposal 1 for f $h(x, y) = (f(x)y)^2 \bmod n$

$$f : \{0, 1\}^k \rightarrow \mathbf{Z}_n$$

- Let n and n' be 1025-bit resp. 1024-bit RSA moduli
- Let $k = 1022$
- Let $u = x^2 \bmod n'$, where $x < n'/2$
- Let $v = E_{\kappa_1}(u) = v_1 \mid \cdots \mid v_8$, where E is CBC encryption using AES
- Let $f(x) = E_{\kappa_2}(v_8 \mid \cdots \mid v_1)$
- f is one-way, collision-resistant

DAKOTA- Proposal 2 for f $h(x, y) = (f(x)y)^2 \bmod n$

$$f : \{0, 1\}^k \rightarrow \{0, 1\}^k$$

- Let n be 1025-bit RSA-modulus, let $k = 1024$
- $f(x) = g(x) \oplus x$, where g is permutation of 1024 bits
- proposal for g :
 - transform x into 8×8 matrix A with 16-bit values
 - Do 4 times

$$A \leftarrow E(A)^T,$$

where E is AES encryption (fixed key) of every column

DAKOTA- Performance

$$h(x, y) = (f(x)y)^2 \bmod n$$

Hash function	Approximate speed (cycles/byte)
SHA-256	21.5
VSH	840
DAKOTA (Proposal 1)	400
DAKOTA (Proposal 2)	345

The end

Thank you