A MAC Based on Matrix Groups

Matthew Cary, University of Washington

Joint work with Ramarathnam Venkatesan, Microsoft Research
Our Results

- A New Universal Hash
  - Fast: < 4 cycles/byte hashed peak rate
  - Secure: collision probability tunable from $2^{-54}$ to $2^{-108}$
  - Low Set-up Cost: < 1/2 generated key of UMAC

- Novel Hash Construction
  - Matrix groups easily analyzed
  - Algebraic structure reduces generated key
  - Adroit choice of group keeps hash fast
Talk Outline

• What’s a secure hash?
• What’s a MAC?
• Why Universal Hashing?
• What’s our construction?
• How well does it work?
• Where do we go from here?

February 24, 2003: Matrix Group MAC
Hash Functions

Reduce large files to small identifiers

Different documents don’t collide
  • Any impossible goal

a.k.a imprint, fingerprint, digest
Keyed Hash Functions

- **Family** of hash functions $\mathcal{H}_1, \mathcal{H}_2, \ldots$ parameterized by key

- $\mathcal{H}_k$ is a good hash for any fixed key $k$
Secure Hash Functions

- Collisions *infeasible* to find in practice
- Many constructions based on block ciphers (DES, MD5, etc.)

Hash functions used for cryptography
Uses of Hash Functions

- Checksum for network transmission
- Searching and Comparison
- *Data Integrity and Message Authentication*
- Examples: CRC, MD5, SHA-1, UMAC
How to prevent an adversary from tampering with a document?
Secure keyed hash sent with document detects tampering

MAC = Message Authentication Code = Secure Keyed Hash
The Problem

• Traditional secure MACs are slow
  ★ Based on block cipher construction
  ★ Must over-design for security

• How to speed MAC computation without sacrificing security?
  ★ Move from Megabyte per second to Gigabyte per second rates
Universal Hash Functions

Any pair of documents unlikely to collide if key chosen randomly

\[ \forall \text{ documents } X, Y, \Pr_k[\mathcal{H}_k(X) = \mathcal{H}_k(Y)] < 2^{-(\text{hash size})+c} \]

- Insecure: may be easy to find collisions for a fixed key
Universal Hashing for MACs

Universal Hash

Ensures no collisions

Secure Cipher

Hides hash structure from adversary

Two-stage process separates *collision-resistance* from *security*
Universal Hashing for MACs

- Document hashed using fast universal hash
  - 0.5–4 cycle/byte speeds possible
  - Collisions analysis with randomness of key, not adversarial model

- Only final hash enciphered with secure cipher
  - Final hash small, so over-designed cipher has little impact on overall MAC rate
  - Adversarial analysis separate from collision analysis
  - Security and efficiency need not trade-off
Universal Hashing for MACs

Well-known and Accepted Technique

- [Carter, Wegman 1981] — Universal hashing
- [Krawczyk 1994] — CRC hashing
- [Rogaway 1995] — Bucket hashing
- [Halevi, Krawczyk 1997] — MMH hash
- [Black et. al. 1999] — UMAC
Related Work: UMAC

- UMAC [Black et. al. 99]
  - Large key combined with document using simple arithmetic operations
  - Generate hash key from MAC with pseudo-random generator (e.g. RC4)
  - Hash efficiently implemented with SIMD (e.g. Intel MMX) instructions
  - MAC rates under a cycle per byte

- Our construction
  - More sophisticated hash uses key randomness better without sacrificing (much) efficiency
  - Generated key < 1/2 size of UMAC's
  - MAC rates of 2–4 cycles per byte
  - Scalable collision parameters
Our Construction

Break input into blocks

$h_0 \xrightarrow{\mathcal{H}_{\text{block}}} \mathcal{H}_{\text{block}} \xrightarrow{\mathcal{H}_{\text{block}}} \mathcal{H}_{\text{block}} \xrightarrow{\mathcal{H}_{\text{block}}} \mathcal{H}_{\text{block}} \xrightarrow{\mathcal{H}_{\text{block}}} \mathcal{H}_{\text{block}} \xrightarrow{\mathcal{H}_{\text{block}}} \mathcal{H}_{\text{block}} \xrightarrow{\mathcal{H}_{\text{block}}} \mathcal{H}_{\text{block}}$

Secure Hash

Output hash value

Block Hashing and Inter-Block Chaining

February 24, 2003: Matrix Group MAC
Block Hashing

Block words combined with key words to produce vectors using 32 → 64 bit multiply

Sum vectors to get \( \sigma \)

Vectors embedded into matrices which are multiplied to give block hash

\[
\left( \begin{array}{c}
M \\
0
\end{array} \right) \begin{bmatrix} u \\ v \end{bmatrix} \\
\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}
\]

\((u, v)\) and \(\sigma\) form output hash
Matrices for the Block Hash

- $H_{\text{block}}$ described using $2 \times 2$ matrix sequence $A = (A_1, A_2, \ldots)$
- Block $X$ is sequence of words $(x_1, x_2, \ldots)$
- Each block word $x_i$ and key word $k_i$ combine to form vector:
  
  $x_i \times k_i \rightarrow \begin{pmatrix} u_i \\ v_i \end{pmatrix}$

- Block matrix product formed by combining $A_i$ with $i^{\text{th}}$ vector:
  
  $\begin{pmatrix} A_i & u_i \\ v_i & 0 \\ 0 & 0 \end{pmatrix} \rightarrow B_i$

- Last column $(u, v)$ of block matrix used in $H_{\text{block}}$:
  
  $\prod B_i = \begin{pmatrix} B & u \\ u & v \\ 0 & 0 \end{pmatrix}$, \quad $(u, v) \rightarrow H_{\text{block}}(X)$
Implementation and Collisions

• Choose $A$ to be repeating sequence of following matrices:

$$\begin{pmatrix} -1 & 1 \\ 1 & -2 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 3 \\ 1 & 2 \end{pmatrix}$$

• $A$ makes a good hash:

  ★ Lemma: Any short subsequence of $A$ is “nearly” invertible

  ★ Lemma: Near invertibility $\Rightarrow$ last column of $\prod B_i$ is nearly universal hash, and $(u, v)$ and $\sigma$ behave independently

  ★ Theorem: $(u, v, \sigma)$ are nearly universal hash

• $A$ is an efficient hash:

  ★ Small entries mean products can be computed with small number of additions or subtractions, not full matrix multiplies
## Performance

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Security (Bits)</th>
<th>Peak Rate (cycles/byte)</th>
<th>Key Size (8 Kbyte Message)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>108</td>
<td>3.7</td>
<td>13.6 Kbits</td>
</tr>
<tr>
<td>Ours</td>
<td>54</td>
<td>2.0</td>
<td>6.8 Kbits</td>
</tr>
<tr>
<td>UMAC</td>
<td>60</td>
<td>0.98</td>
<td>32 Kbits</td>
</tr>
<tr>
<td>SHA-1</td>
<td>80</td>
<td>12.6</td>
<td>512 bits</td>
</tr>
</tbody>
</table>

Data for other algorithms taken from [Black et. al. 99]
Future Work

- Improve inter-block chaining to reduce key size

- Find better matrix sequences
  - Partial results toward a sequence with general proof and 0/1 entries

- MACs for short messages
  - Application: authenticating wireless packets