Hongjun Wu$^{1,2}$

$^1$Institute for Infocomm Research
$^2$Nanyang Technological University
• Design
• Security
• Performance
Innovative Design

- New compression function structure
- Proposed the generalized AES design method
- Combining the best of AES and Serpent
Design: New compression function structure

$M^{(i)}$: $m$ bits
$H^{(i)}$: $2m$ bits

New, simple

efficient

=> does not discard

part of the output of $E$
Design: Proposed the generalized AES design method

**SPN + MDS code** (to a multi-dimensional array)

=> A simple, efficient and flexible approach to design a large permutation (block cipher) from small components by increasing dimension

Examples:
- AES (2D, 128 bits) => 3D (512 bits) => 4D (2048 bits);
- JH (8D, 1024 bits) → bit-slice
Design: Combining the best of AES and Serpent

AES
SPN + MDS code

Serpent
Bit-slice fast software implementation

JH
Security

The generalized AES design:
SPN + MDS (to a multi-dimensional array)

• Advantages
  – Analyze a small function to find the best attack
  – Verify the attack on a small function
Security: Differential attack

a compression function in JH involves:

512 message bits, 9216 Sboxes

a differential path in JH involves more than 600 active Sboxes => strong against differential attack
Security: Other attacks

• Preimage attack on the mode
  – meet-in-the-middle collision search
  – $2^{507}$ computations + $2^{507}$ memory + $2^{526}$ memory accesses
    (Bhattacharyya et al, FSE 2010)
  – more expensive than brute force
Security: Other attacks (contd.)

- Rebound attacks on JH (Rijmen et al. FSE 2010)
  - Semi-free-start collision
    - 16 out of 35.5 rounds \((2^{178} \text{ computations} + 2^{101} \text{ memory})\)
  - Semi-free-start near collision
    - 22 out of 35.5 rounds \((2^{156} \text{ computations} + 2^{143} \text{ memory})\)

- My opinions on rebound attacks
  - Rebound attack is not a threat to JH
    - As stated in the original JH submission, JH compression function is strong against the attack from the middle
  - Rebound attack is only useful for Matyas–Meyer–Oseas-like structure
Security: Other attacks (contd.)

- 820/1024-bit near collision for 10 out of 35.5 rounds
  \( (2^{23} \text{ computations}, \text{Turan et al., the 2nd SHA-3 conference}) \)
Security: Proof

- JH
  - Indifferentiable with less than $2^{n/3}$ queries ($n = 512$)
    Low bound (Bhattacharyya et al, FSE 2010)
Performance

• Hardware or resource constrained platform
  – Identical round functions
  – Small components
  – Compute round constants on-the-fly

• Fast software
  – Bit-slice with seven different round functions

• Easy to implement for software and hardware
  – The difficult part is to derive the bit-slice description
    => I did it already (two years ago in the submission)
Performance: Fast software

Bit-slice; suitable for the 128-bit SIMD instruction set (available on many platforms):

- compute 128 Sboxes in parallel
- compute 128 MDS codes in parallel

ebash results:
about 17 cycles/byte on the common Intel & AMD processors;
very close to that of SHA-256
Performance: Fast software (contd.)

- More efficient on the incoming microprocessors with 256-bit AVX instruction set
  - Intel (Sandy Bridge Q4 2010)
  - AMD (Bulldozer 2011)

  **Compute 256 Sboxes in parallel**

- The size of data register has been gradually increasing
  8 bits $\rightarrow$ 16 bits $\rightarrow$ 32 bits $\rightarrow$ 64 bits $\rightarrow$ 128 bits $\rightarrow$ 256 bits $\rightarrow$
Performance: Hardware

• ASIC: Tillich et al (the 2nd SHA-3 conference)
Performance: Hardware (contd.)

- Gaj et al (the 2nd SHA-3 conference)
Performance: Hardware (contd.)

- Gaj et al. (the 2nd SHA-3 conference)

Throughput vs. Area Normalized to Results for SHA-512 and Averaged over 7 FPGA Families – 512-bit variants
Performance: Hardware (contd.)

• Baldwin et al. (the 2nd SHA-3 conference)
  – JH is one of the top three candidates in FPGA implementation
Conclusion

- **Design**
  - New compression function structure
  - The generalized AES design method
  - Combining the best of AES and Serpent

- **Strong**

- **Efficient for software and hardware**