Observations on H-PRESENT-128

Niels Ferguson
Microsoft
H-PRESENT-128 hash function

- Proposed by Bogdanov, Leander, Paar, Poschmann, Robshaw, and Seurin at CHES 2008.
- Uses the PRESENT-128 64-bit block cipher in Hirose’s double-length mode.
Details

- All values 64 bits
- \((H_1, H_2)\) = chaining state
- \(M\) = message block
- \(c\) = public constant
Distinguisher for compression function

- For any $H_1$, $H_2$, $M$, compute
  $(A_1, A_2) = F(H_1, H_2, M)$ and
  $(B_1, B_2) = F(H_1 \oplus c, H_2, M)$
- Check
  $(A_1, A_2) = (B_2 \oplus c, B_1 \oplus c)$
- Requires 2 chosen plaintexts
Distinguisher for hash function

- $H_1' = H_2'$ is impossible
- Ideal hash function has equal output halves with prob. $2^{-64}$
- Distinguisher requires $2^{64}$ hash outputs.
Pseudo-preimage

- Generic exhaustive search: $2^{128}$ tries using $2^{129}$ block cipher enc.
- Trivial Improvement: pick M, H1, H2, compute only left half to reject wrong choice. Uses $2^{128}$ block cipher enc.
Pseudo-preimage

• Improvement 2: A single encryption can rule out both \( H_1 \) (using left half) and \( H_1 \oplus c \) (using right half)
• \( 2^{127} \) encryptions needed
• 4x faster then generic exhaustive search
Conclusion

• Distinguisher for compression function using 2 chosen inputs
• Distinguisher for hash function using $2^{64}$ hash outputs
• 4-fold speedup of pseudo-preimage search compared to generic algorithm