How to Attack a Hash Function
(in one easy lesson)

John Kelsey, NIST, August 2006
Damgaard-Merkle Construction

- Building a Hash Function from a Compression Function
  - Hash function takes variable length input
  - Compression function takes fixed length
  - Collision in hash function
    - $\implies$ Collision in compression function
Inside the Compression Fn.

- Sequence of rounds mix state with message
- Message schedule sends message to rounds
- Feedforward makes it hard to go backwards
Overview: Finding a Collision

- Find a differential path (roadmap to collision)
- Repeat:
  - Choose \( M, M^* \) to follow as far as possible
  - Check to see if it follows path to end
- Until we get a collision
The Differential Path:

We only care about differences, not value of $M, M^*$
Differential Path: Conditions vs Probabilities

Follow differential path by satisfying conditions

Follow differential path by getting lucky
Message Modification

• Choose M (and thus M*) to satisfy as many conditions as possible.
  – Simple: Free choice of message bits
  – Advanced: Message bits being altered may mess up earlier conditions
Switching to the Probability View

• At this point, we just see if the pair follows the differential path
  – Early Stopping
  – Backtracking/Free Bits of Message
  – Neutral Bits/Tunnels
Full Collision Attack

- Find differential path -> collision
- Use MM to follow path as far as possible
- Check if path followed after MM
- Repeat until a collision is found
Optimizing the Differential Path

- Finding a good differential path is key to these attacks
- Optimizing DP for message modification
Multiblock Collisions

- What if we can’t find a good differential path for a one-block collision?
  - Find a path for multi-block collision
  - Difference left from M0 is canceled by M1
  - More flexible differential paths
  - Use MM to add still more flexibility to start of path
Attack Tools Can Help….

- Finding differential paths
- Evaluating better/worse paths
- Satisfying conditions in message modification