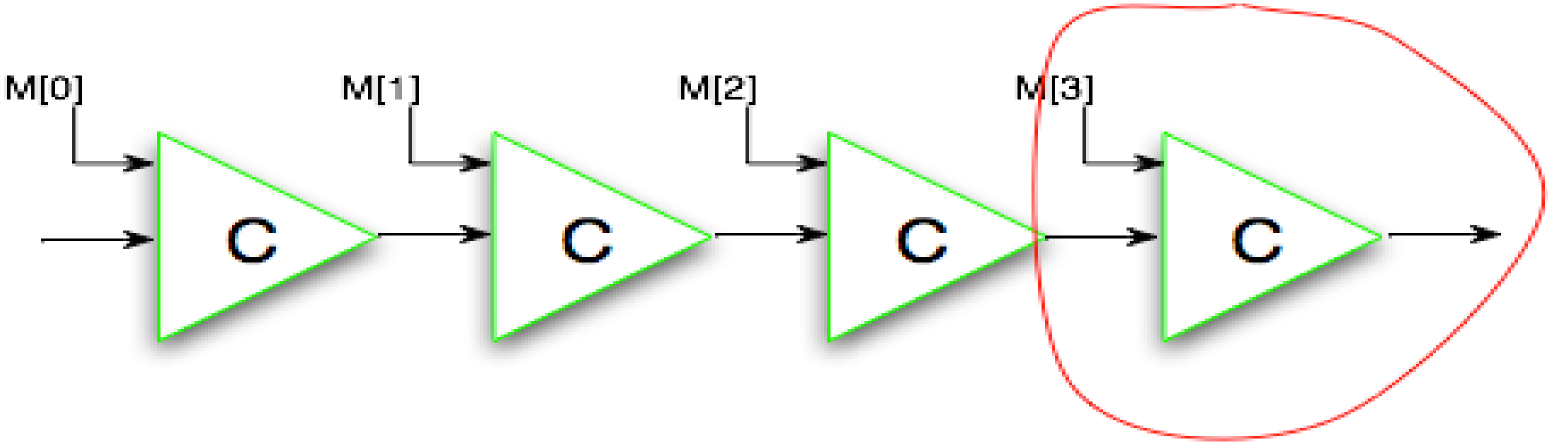
How to Attack a Hash Function (in one easy lesson)

John Kelsey, NIST, August 2006

Damgaard-Merkle Construction

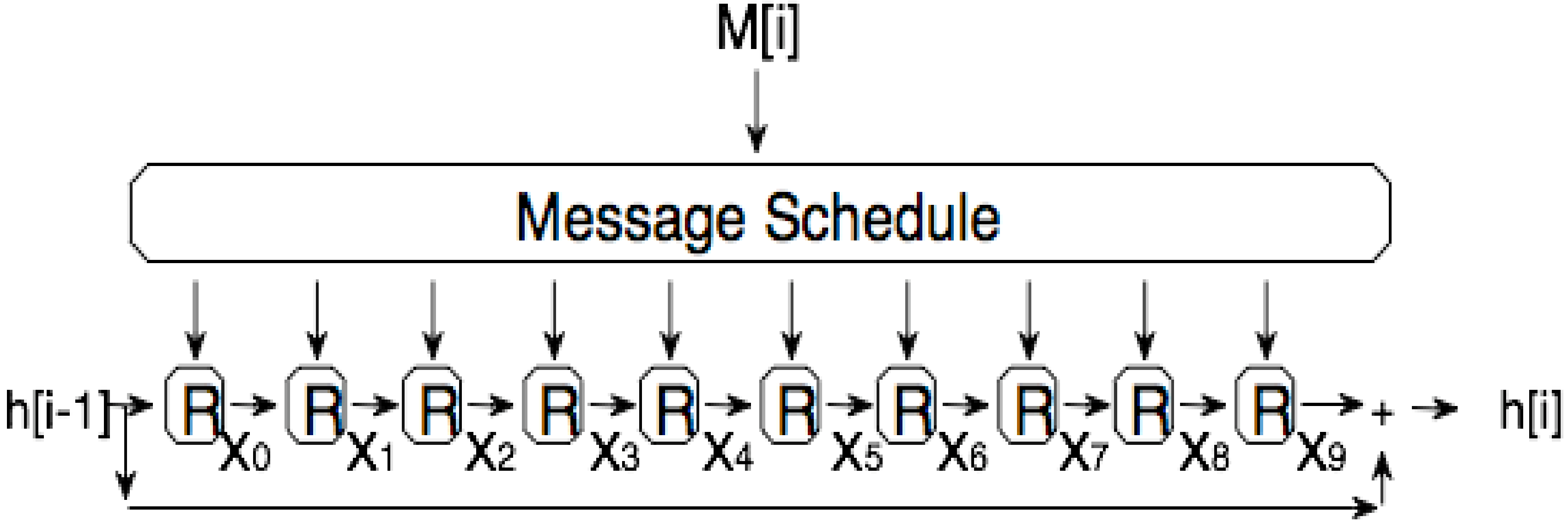
- Building a Hash Function from a Compression Function
  - Hash function takes variable length input
  - Compression function takes fixed length
  - Collision in hash function

    ===> Collision in compression function



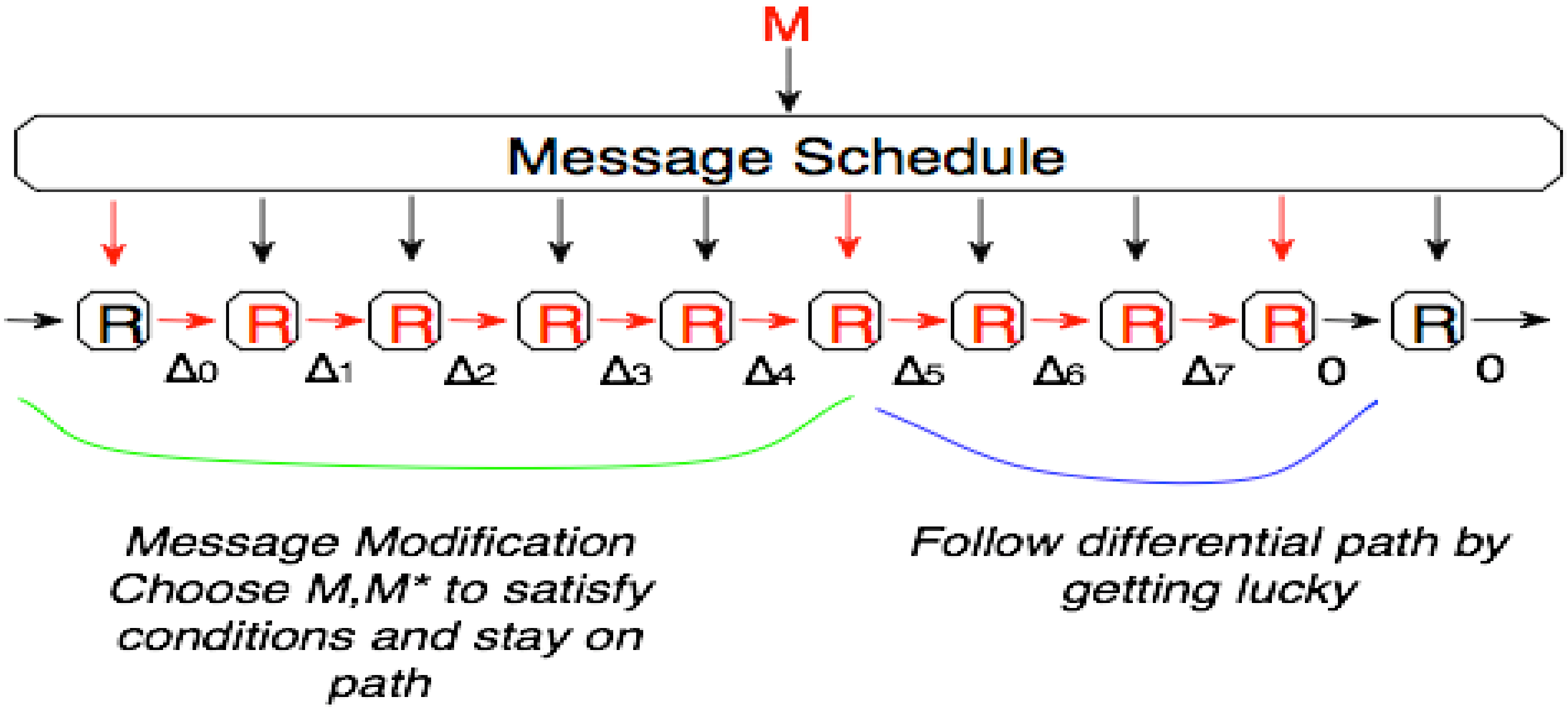
Inside the Compression Fn.

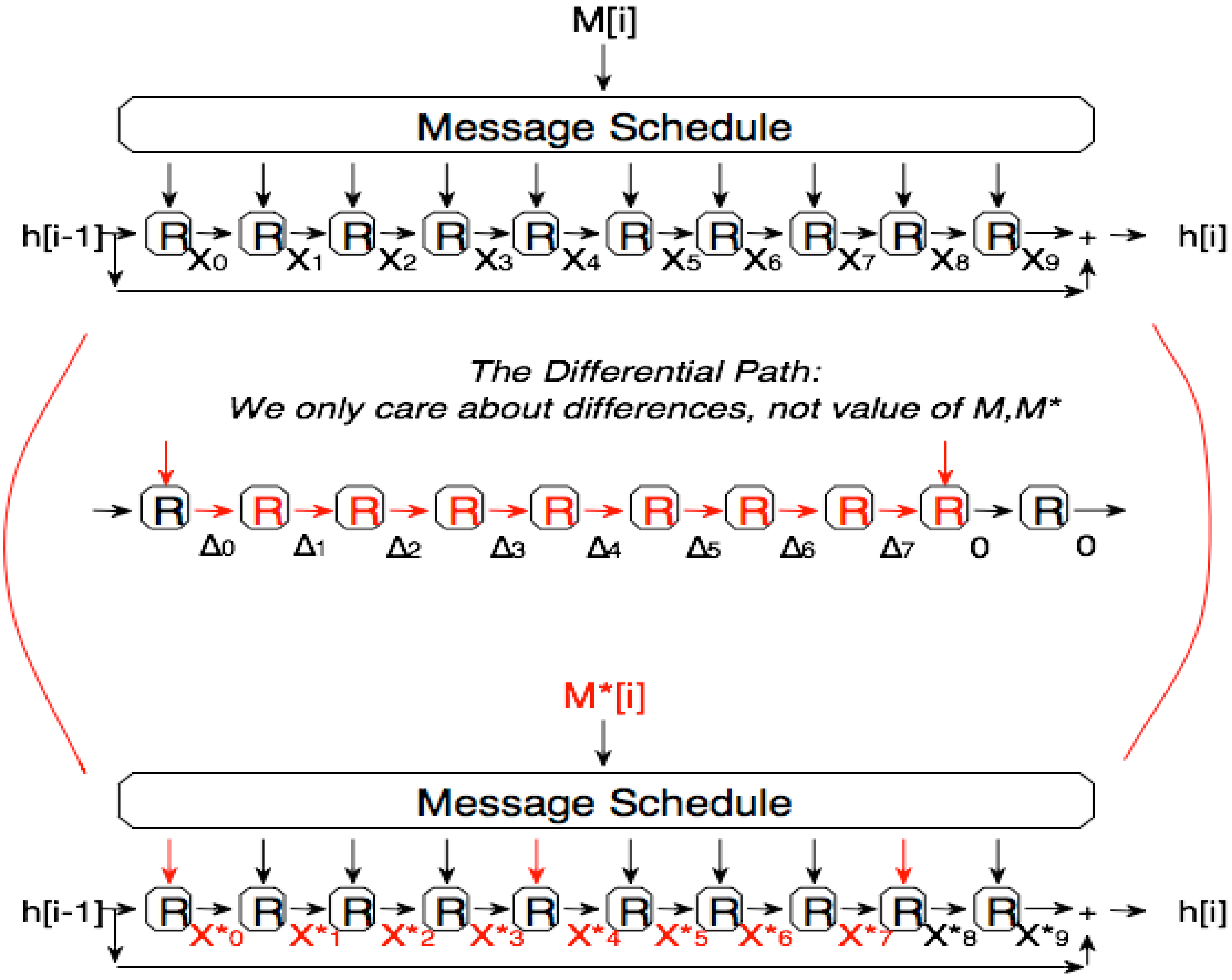
- Sequence of rounds mix state with message
- Message schedule sends message to rounds
- Feedforward makes it hard to go backwards



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

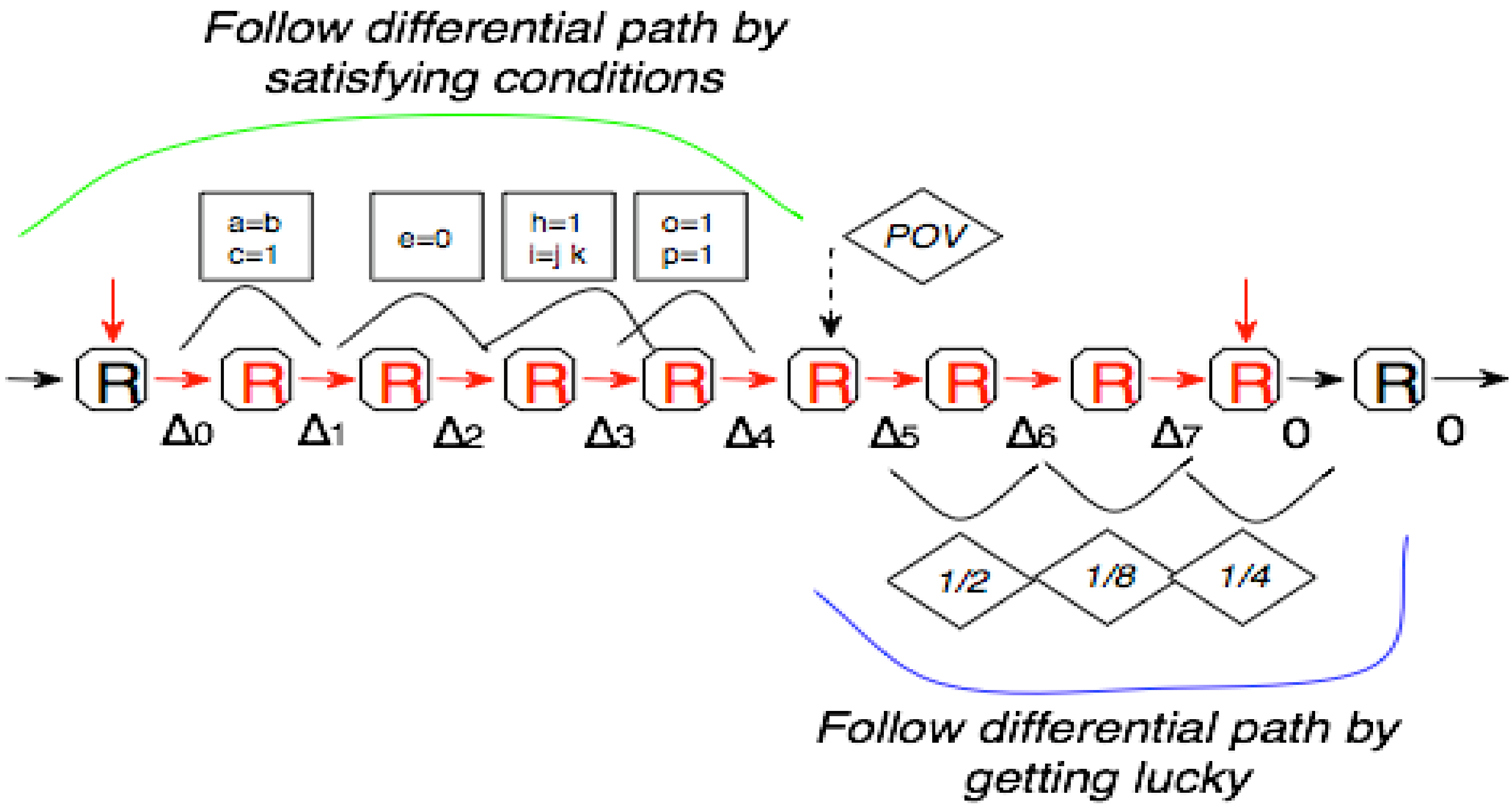

M
Message Schedule
R
Δ0
Δ1
Δ2
Δ3
Δ4
Δ5
Δ6
Δ7
0
0
Message Modification Choose M,M* to satisfy conditions and stay on path
Follow differential path by getting lucky



M[i]
Message Schedule
h[i-1]
R R R R R R R R R R
X_0 X_1 X_2 X_3 X_4 X_5 X_6 X_7 X_8 X_9
+
h[i]
R R R R R R R R R R
Δ_0 Δ_1 Δ_2 Δ_3 Δ_4 Δ_5 Δ_6 Δ_7 0 0
M*[i]
Message Schedule
h[i-1]
R R R R R R R R R R
X^*_0 X^*_1 X^*_2 X^*_3 X^*_4 X^*_5 X^*_6 X^*_7 X^*_8 X^*_9
+
h[i]

The Differential Path:
*We only care about differences, not value of M,M**

Differential Path: Conditions vs Probabilities

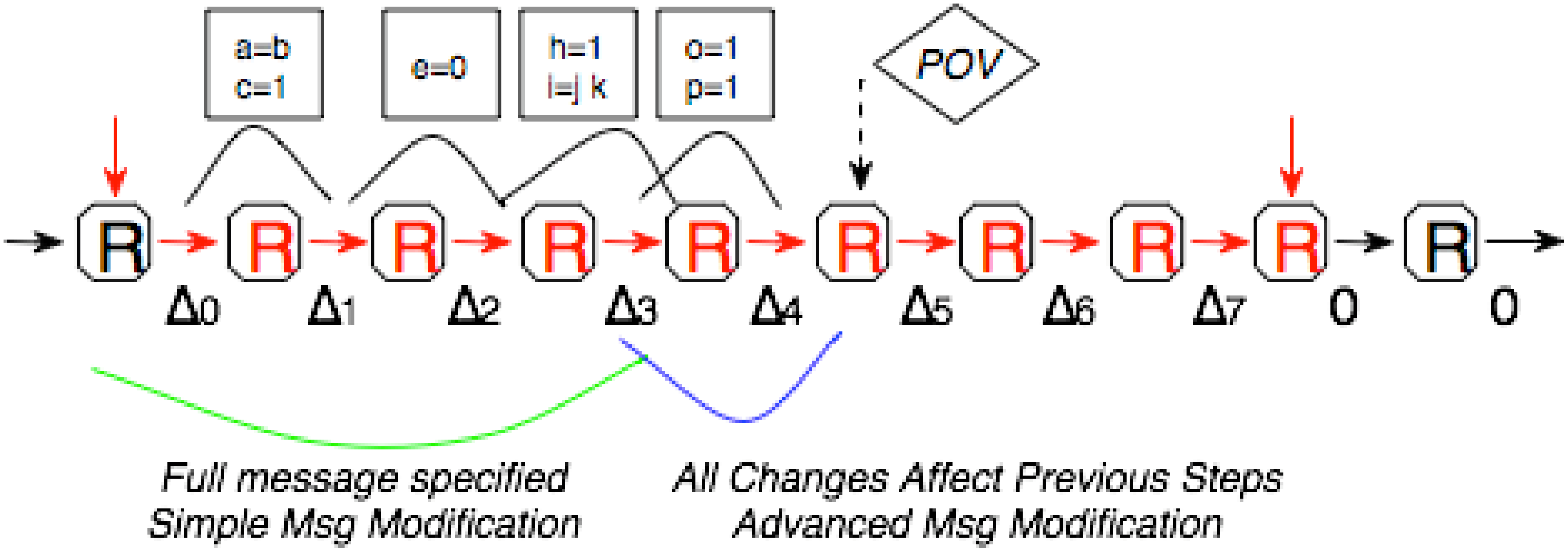


Follow differential path by satisfying conditions

Follow differential path by getting lucky

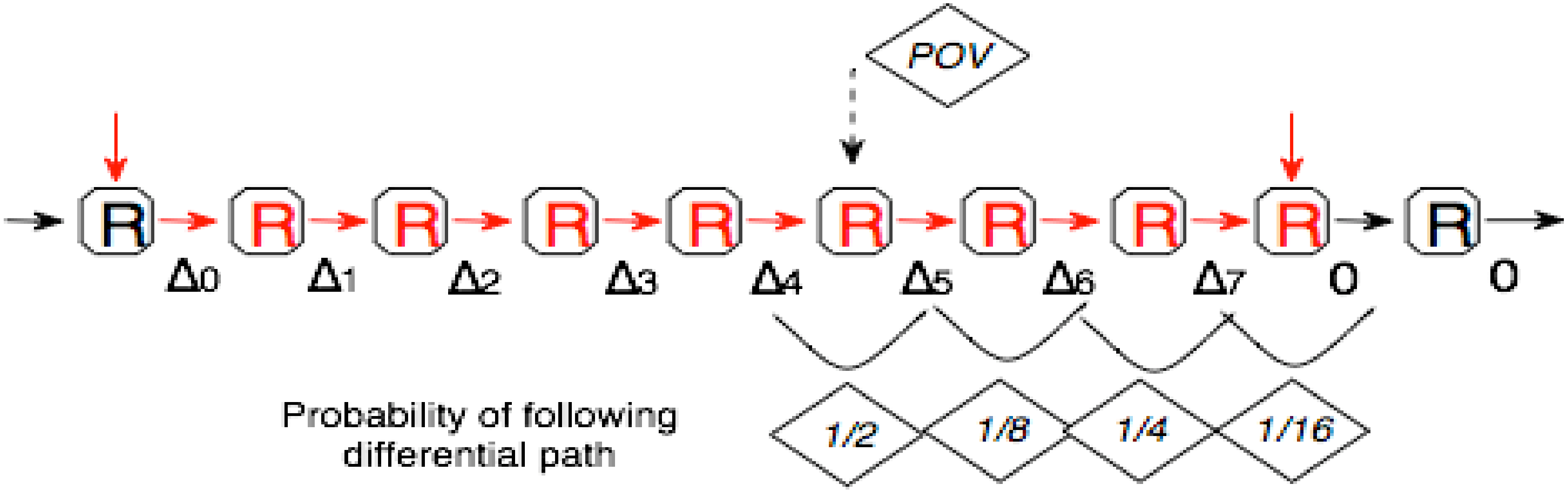
Message Modification

- 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

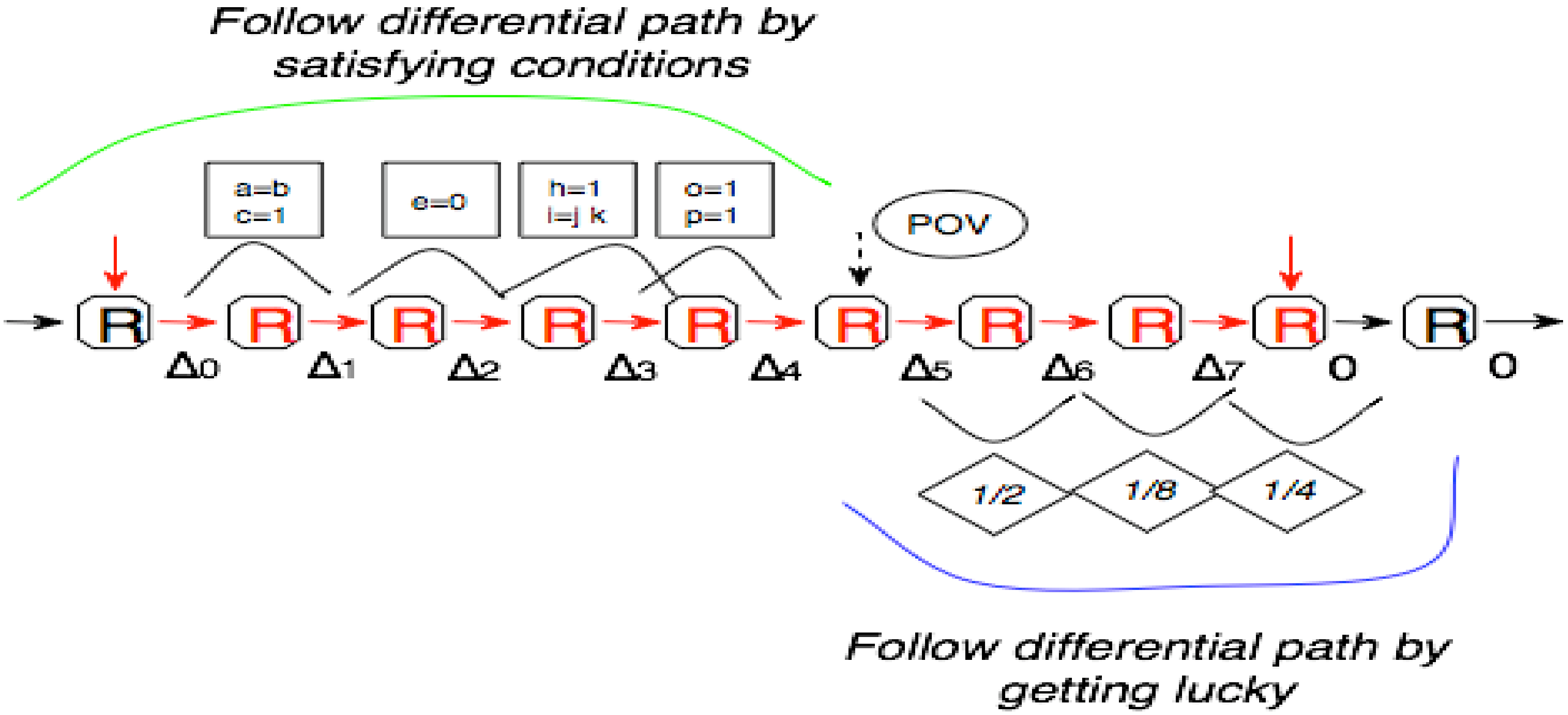
- At this point, we just see if the pair follows the differential path
  - – Early Stopping
  - – Backtracking/Free Bits of Message
  - – Neutral Bits/Tunnels



Probability of following differential path

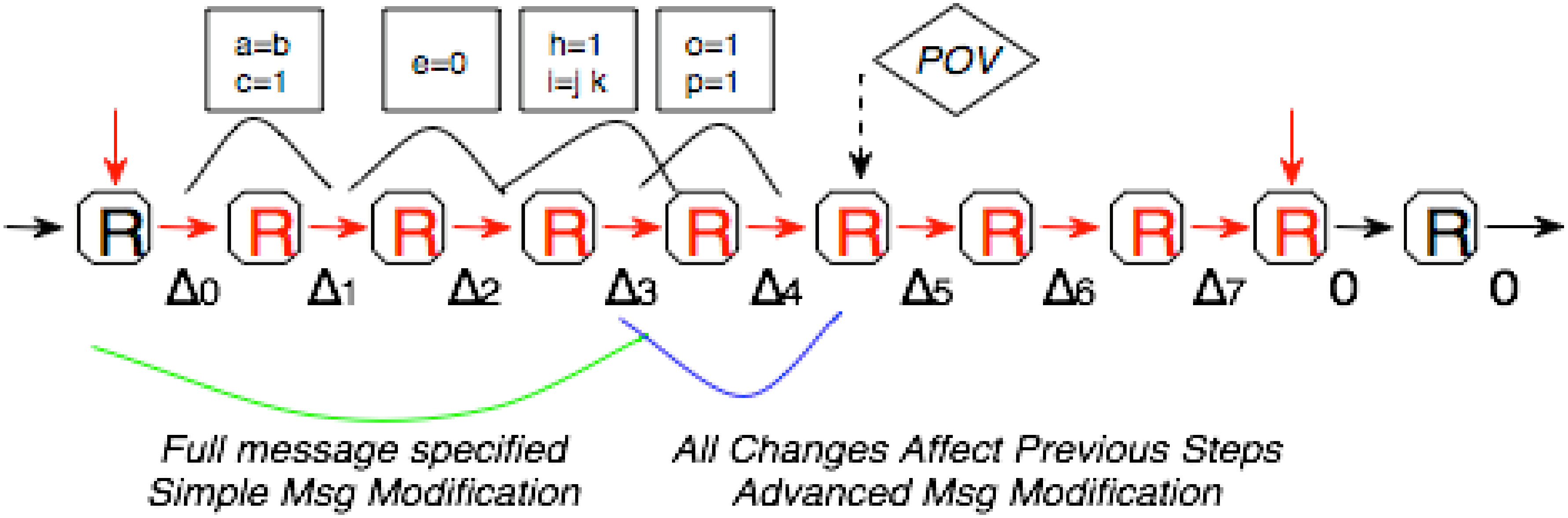
Full Collision Attack

- 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

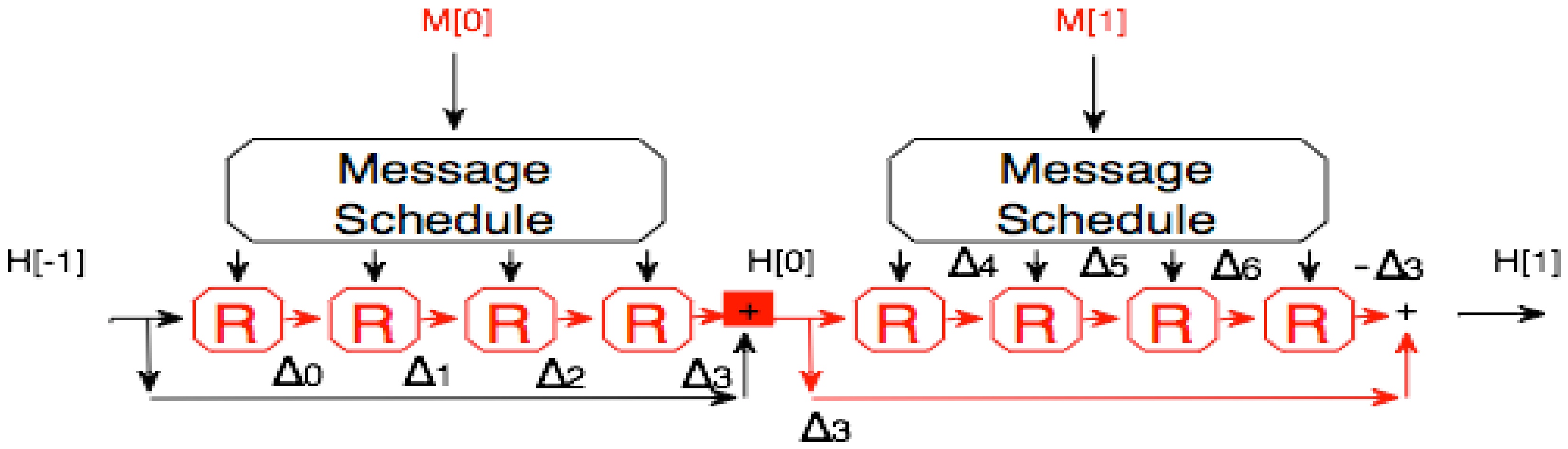

Follow differential path by satisfying conditions
a=b
c=1
e=0
h=1
i=j k
o=1
p=1
POV
R
R
R
R
R
R
R
R
R
R
R
Δ0
Δ1
Δ2
Δ3
Δ4
Δ5
Δ6
Δ7
0
0
1/2
1/8
1/4
Follow differential path by getting lucky

Optimizing the Differential Path

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



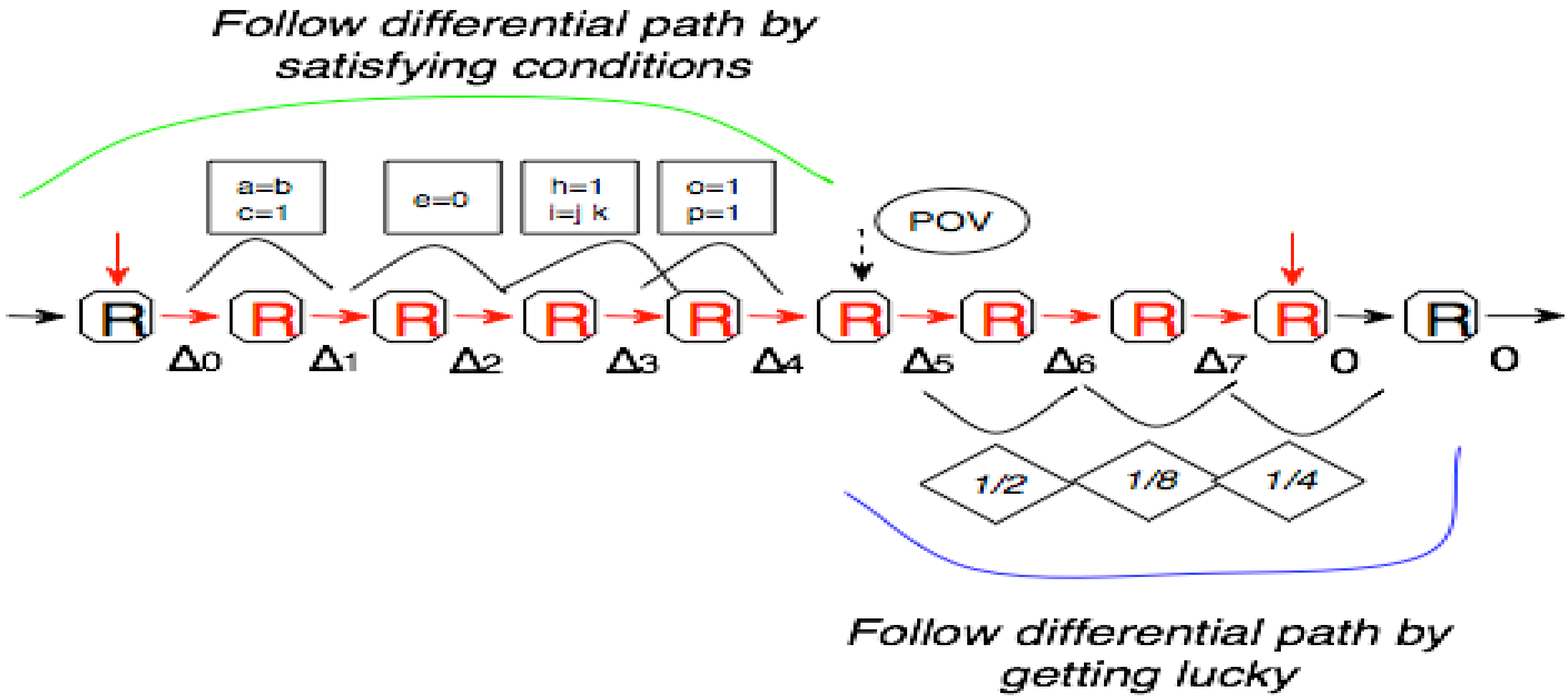
Multiblock Collisions



- 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



Follow differential path by satisfying conditions
a=b
c=1
e=0
h=1
i=j k
o=1
p=1
POV
R
Δ0
Δ1
Δ2
Δ3
Δ4
Δ5
Δ6
Δ7
O
O
1/2
1/8
1/4
Follow differential path by getting lucky