Side-Channel Countermeasure for SHA-3 At Almost-Zero Area Overhead

Mostafa Taha and Patrick Schaumont

Virginia Tech

This research was supported in part by the VT-MENA program of Egypt, and by NSF grant no. 1115839.

Outline

- Side-Channel Analysis
- SHA-3, the story and its applications
- SHA-3 Countermeasure
- New Design Concept

Outline

- Side-Channel Analysis
- SHA-3, the story and its applications
- SHA-3 Countermeasure
- New Design Concept

Side-Channel Analysis

- Information Leakage

Side-Channel Analysis

- Example
  - Password Checker (8 bytes)

`(I[0]=K[0])` is False ➔ one loop

`(I[0]=K[0])` is True
`(I[1]=K[1])` is False ➔ two loops

255 values of `I[0]` need one loop,
while one correct value needs two loops

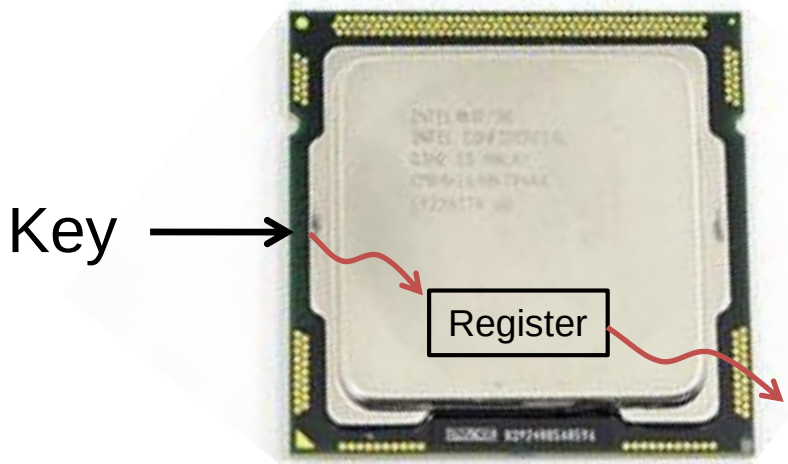
```
for  (i = 0 ; i < 8 ; i++){
        if(I[i] = K[i]){
                Password = true;
        }
        else{
                Password = false;
                break;
        }
}
```

Time

Brute Force security reduced from 256^8 to 256*8
If 1 try = 1 μsec ➔ from 585K years to 2 msec!

Side-Channel Analysis

- Information Leakage



- Computation Time
- Power Consumption
- Electromagnetic
- Acoustic Waves
- Photonic Emission
- Faulty Ciphertext

Side-Channel Analysis

- One Trace ➔ Simple ..... Analysis
- Many Traces ➔ Differential ..... Analysis

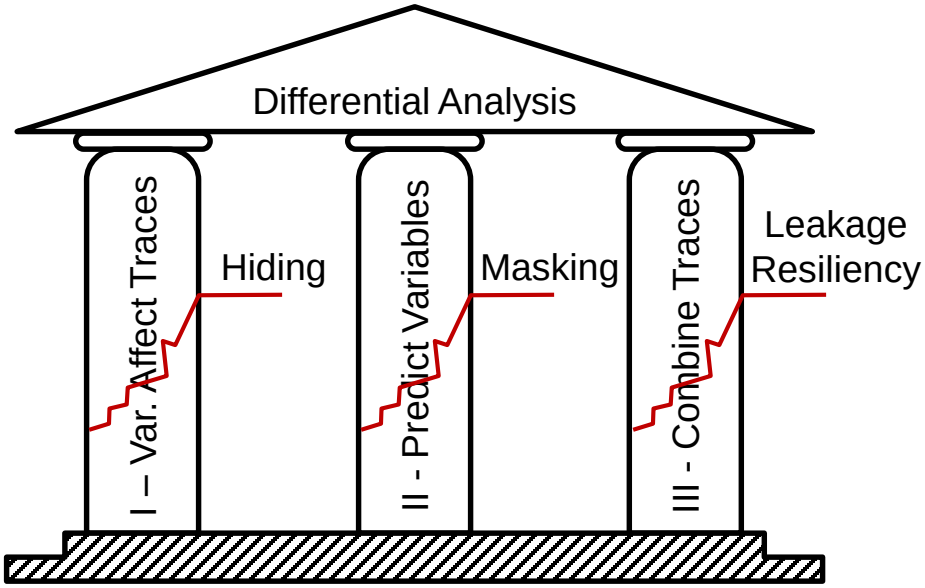
Actual Leakage
2
3
5
4
:

Hypothesis Table

K_0		K_1		K_2	
Variable	Eq. Leakage	Variable	Eq. Leakage	Variable	Eq. Leakage
0x0F	4	0x82	2	0xF1	5
0xAA	4	0x51	3	0x4E	4
0xD3	5	0xA3	4	0x0B	3
0x31	3	0xC7	5	0x92	3
:	:	:	:	:	:

Correlation

Side-Channel Analysis



Outline

- Side-Channel Analysis
- SHA-3, the story and its applications
- SHA-3 Countermeasure
- Why the countermeasure works!

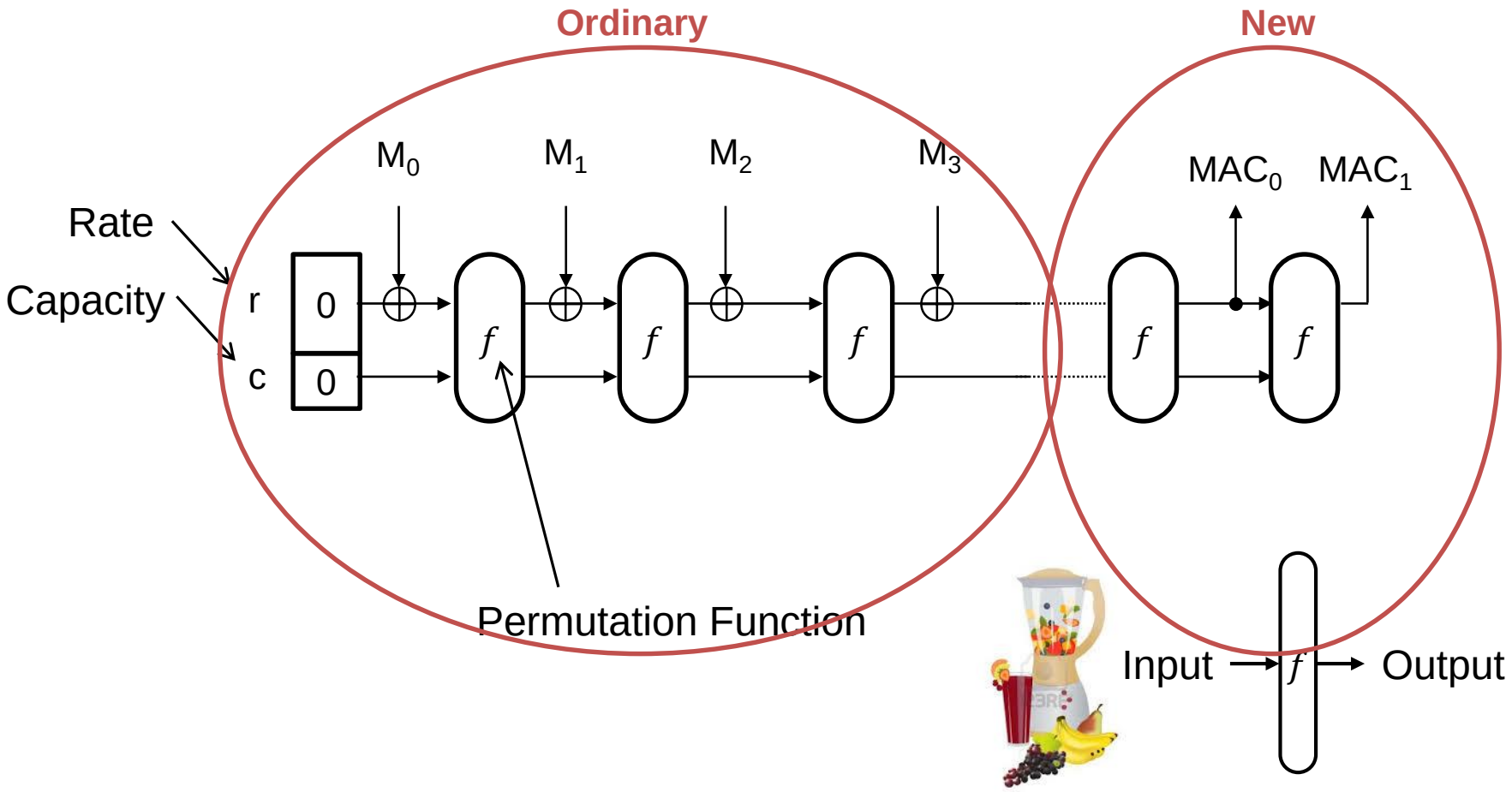
SHA-3 (The Story)

- SHA-3 is a hashing standard

- SHA-0, SHA-1, SHA-2 ⟶ SHA-3
- Competition started in Nov. 2007 and ended in Oct. 2012 with Keccak as the winner.
- Chosen for
  - Superior performance in hardware
  - The *Sponge* construction

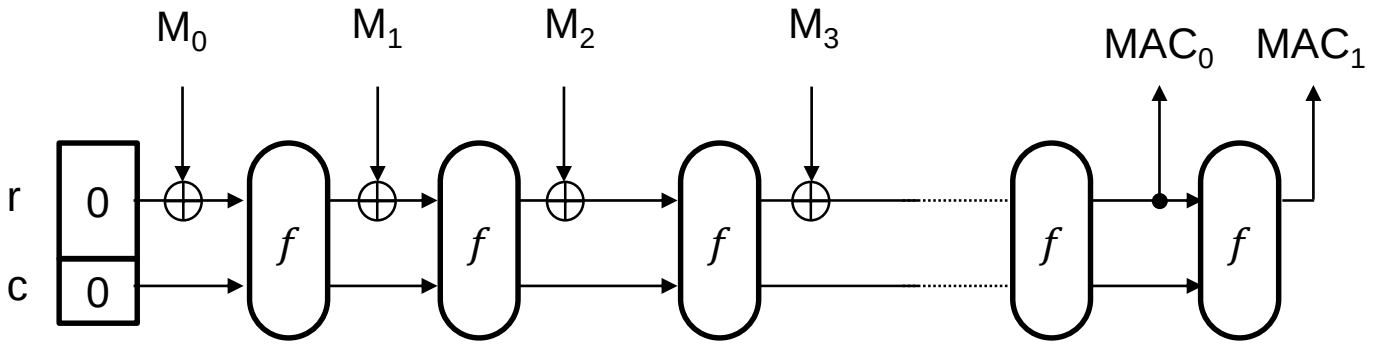
NIST

SHA-3 (The *Sponge* construction)



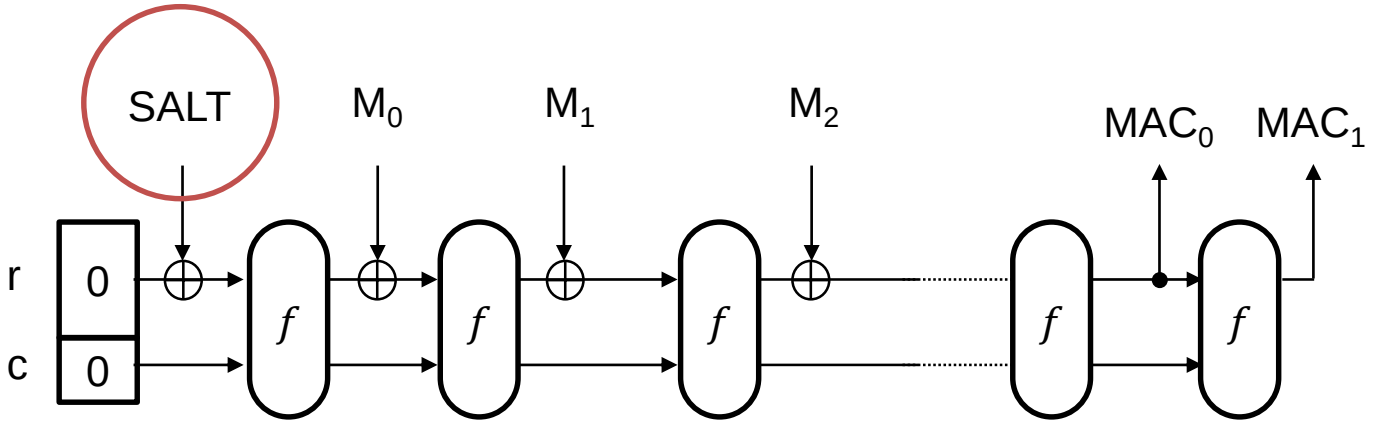
SHA-3 (Applications)

- Regular Hashing



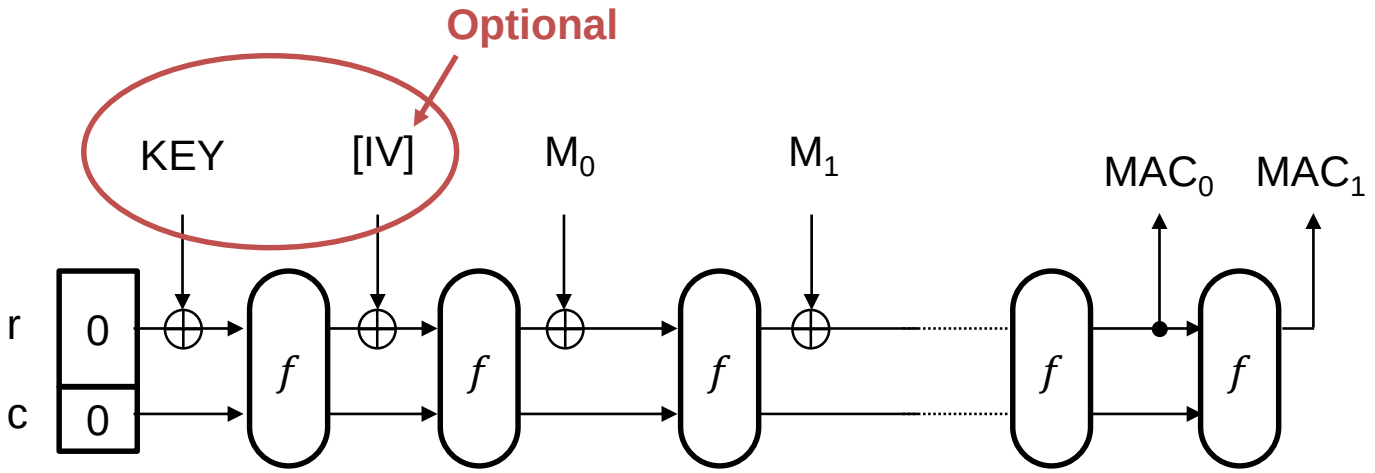
SHA-3 (Applications)

- Salted Hashing



SHA-3 (Applications)

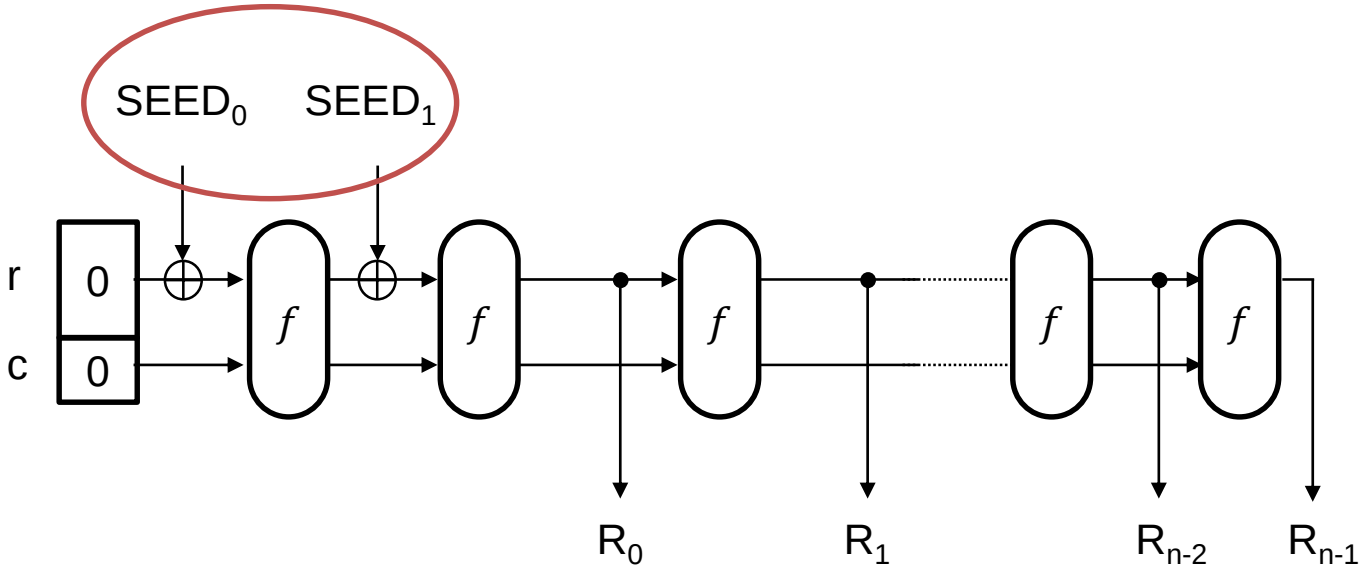
- Message Authentication Codes



- Hashing, salted hashing and MACs are ordinary applications. But!

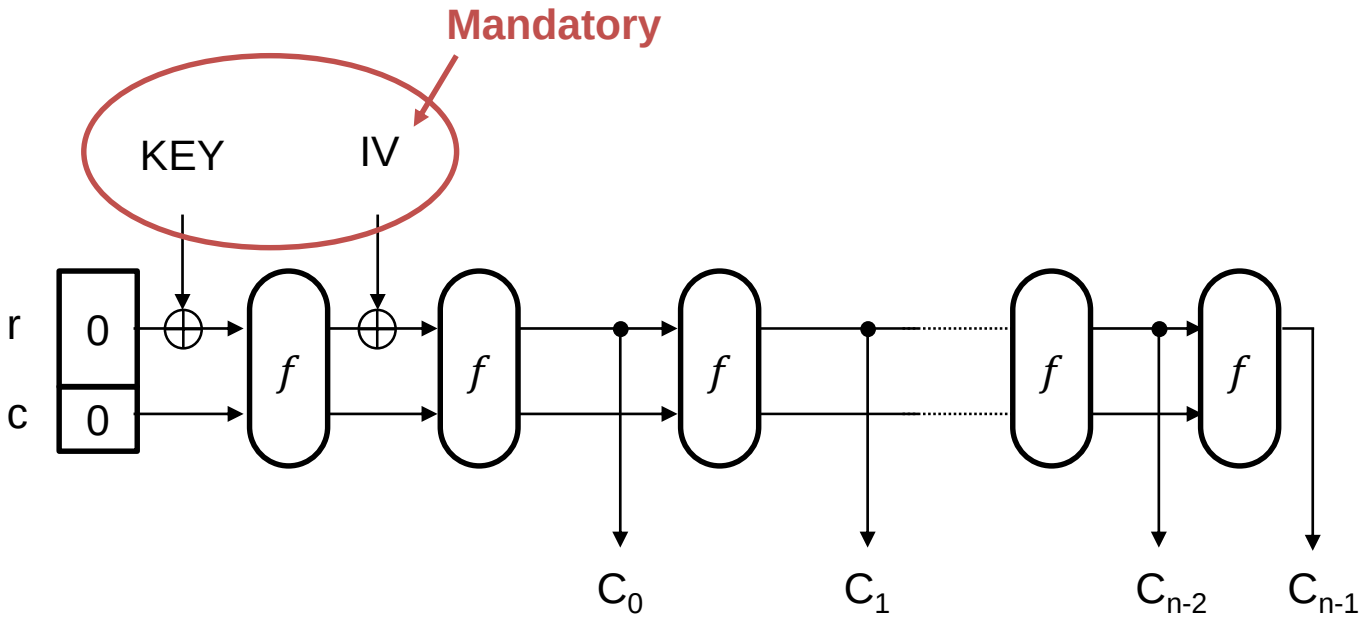
SHA-3 (Applications)

- Random Number Generation New!



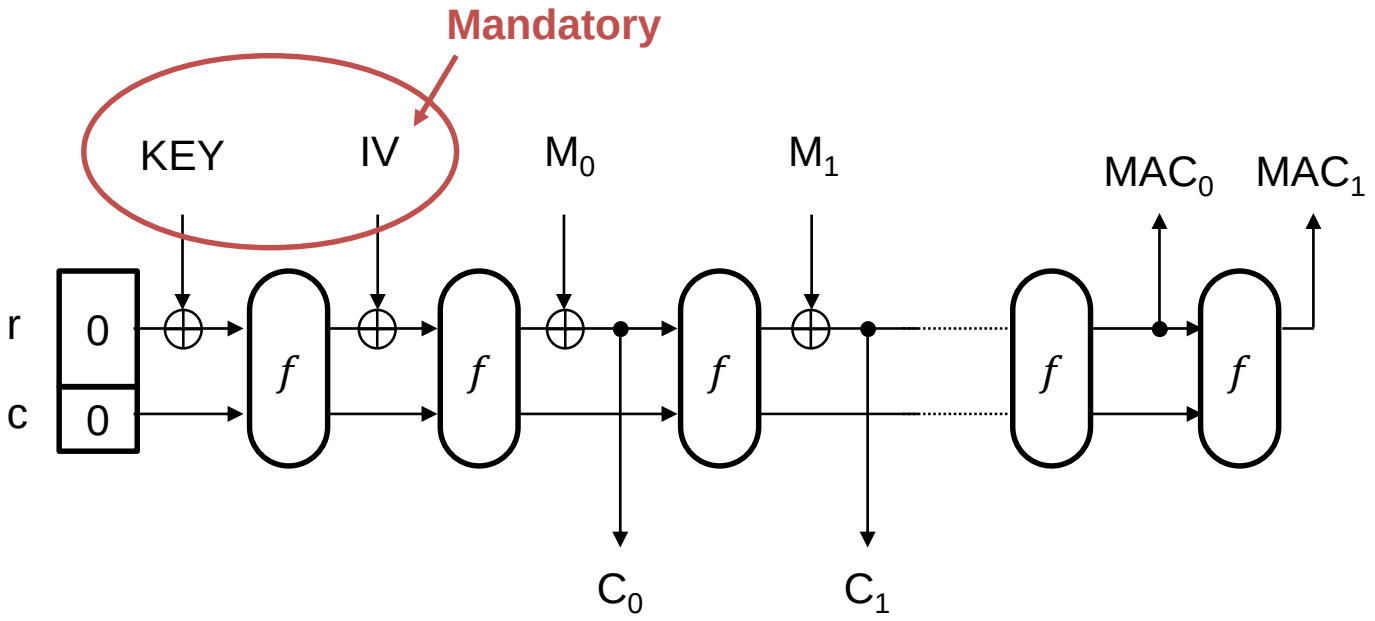
SHA-3 (Applications)

- Stream Encryption New!



SHA-3 (Applications)

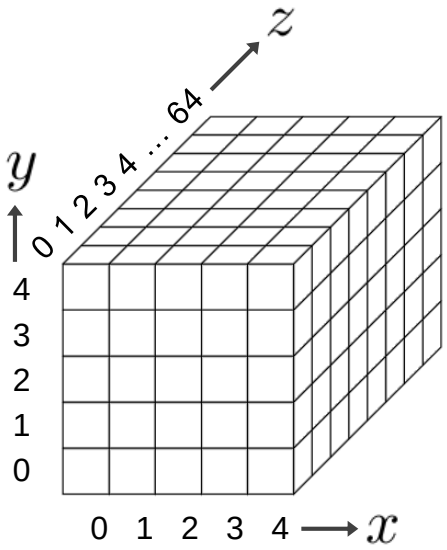
- Authenticated Encryption New!



And many more.

SHA-3

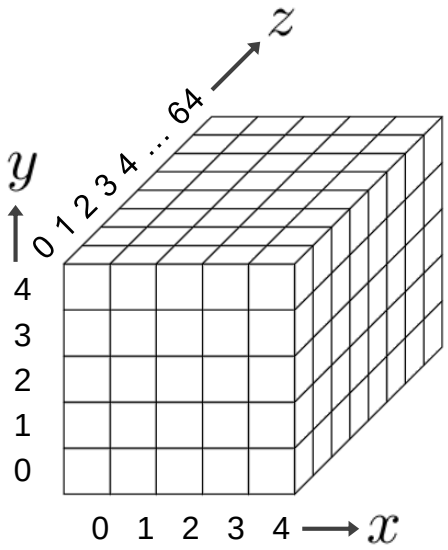
- Keccak permutation function



State
1600 bits

24 Rounds of

$$\iota \circ \chi \circ \pi \circ \rho \circ \theta$$



State
1600 bits

Also, PHOTON, QUARK, SPONGENT,…

SHA-3 (Single Core)

Regular Hashing

Salted Hashing

RNG

MAC-generation

Stream Encryption

Authenticated Encryption

One HW Core

SHA-3 (Single Core)

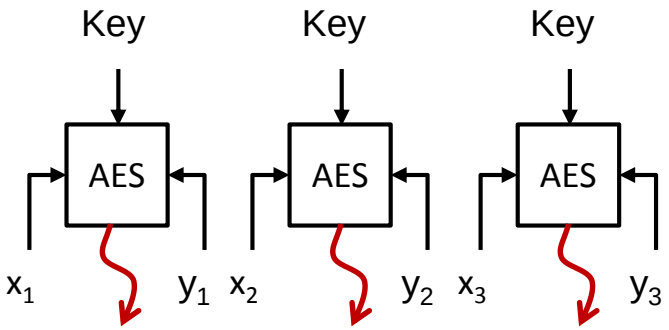
Regular Hashing
Salted Hashing
RNG

MAC-generation
Stream Encryption
Authenticated Encryption

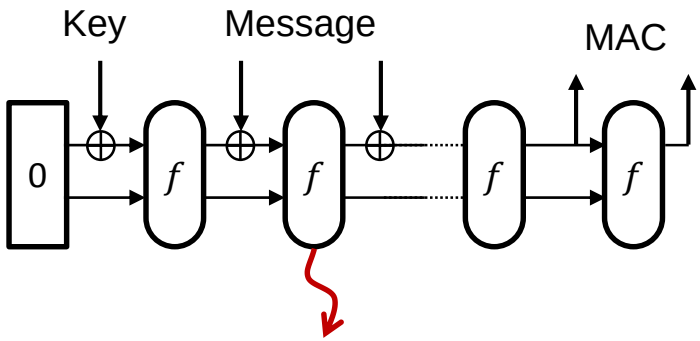
Side-Channel Protection
3x implementation cost

SHA-3 (Single Core)

- Permutation Functions



AES Block-cipher



MAC-Keccak

SHA-3 (Single Core)

Regular Hashing		
Salted Hashing		
RNG		
MAC-generation (m-1)	+	1
Stream Encryption (m-1)	+	1
Authenticated Encryption (m-1)	+	1

Design a new countermeasure

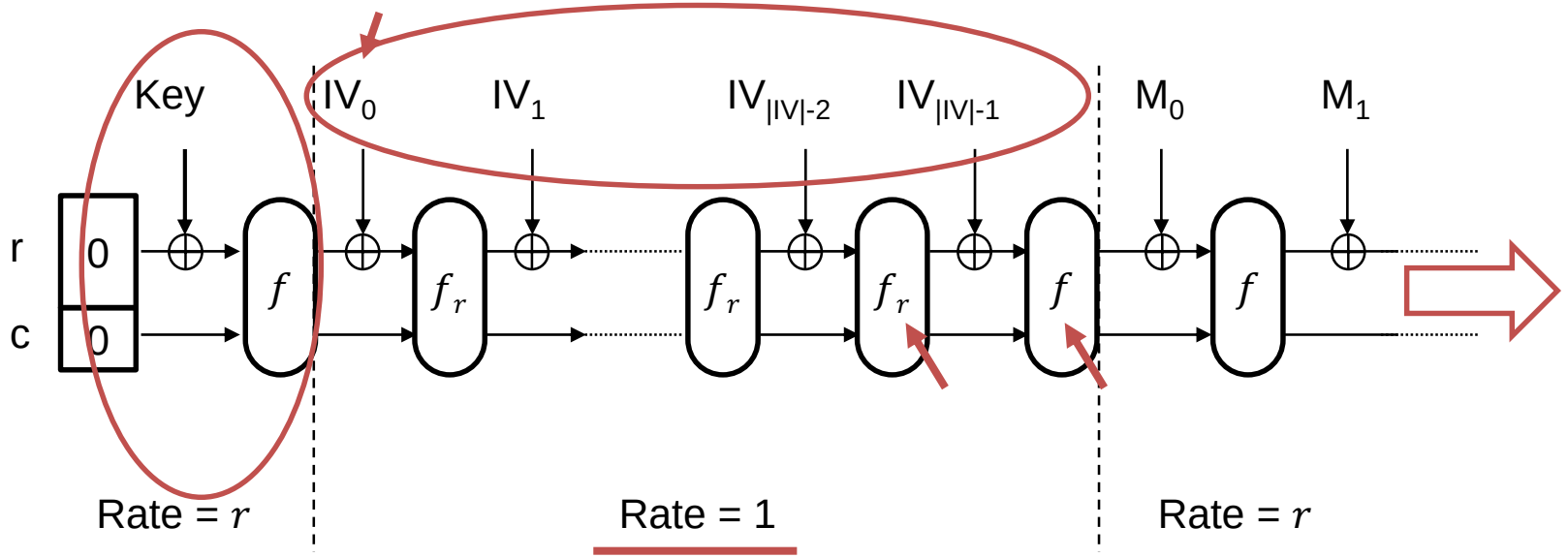
- Minimal changes to the implementation
- Can be turned ON / OFF New!

Outline

- Side-Channel Analysis
- SHA-3, the story and its applications
- SHA-3 Countermeasure
- Why the countermeasure works!

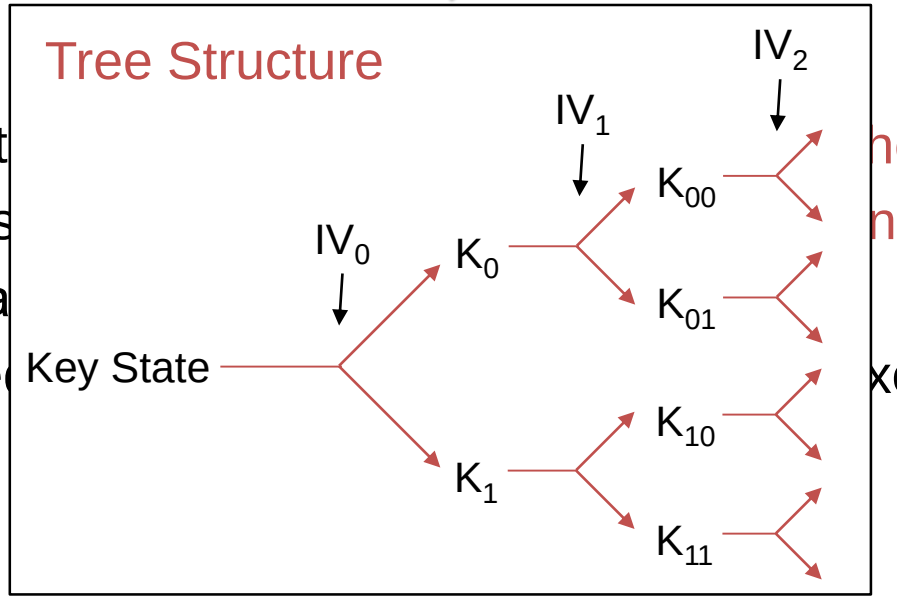
SHA-3 Countermeasure

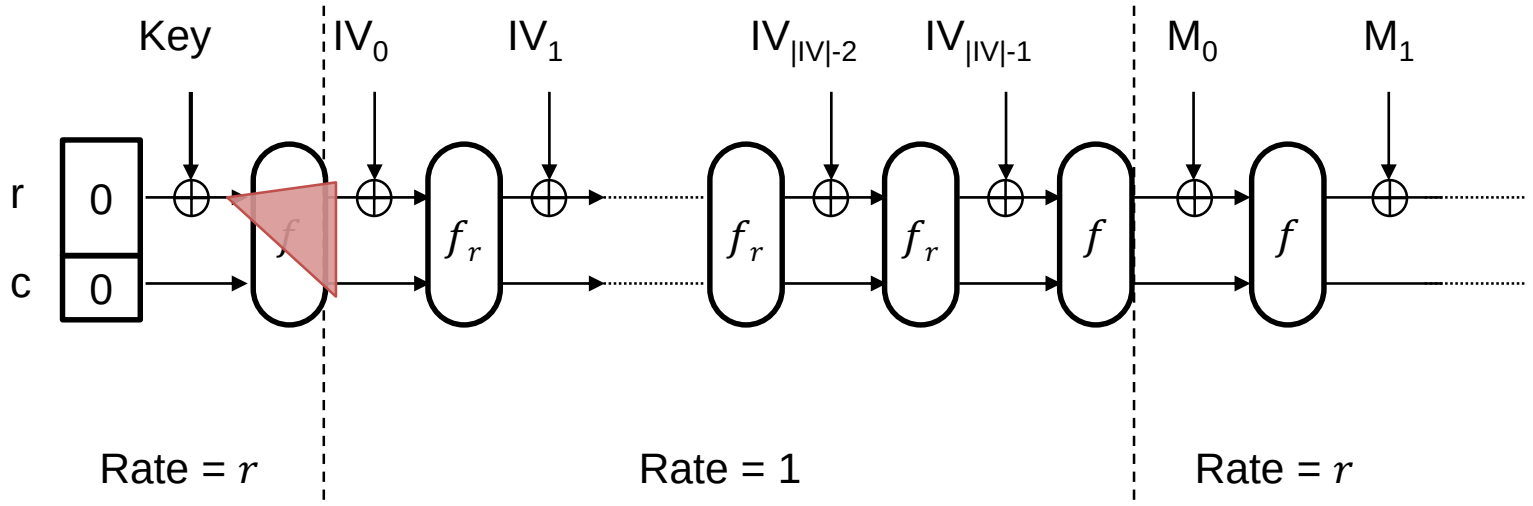
1. The Key goes to a separate input.
2. Mandate the use of IV in all keyed applications.
3. Squeeze the rate to “one bit” during IV.
4. Use Round-Reduced version of Keccak during IV, except the last bit.
5. Then, proceed normally.



SHA-3 Countermeasure, WHY?

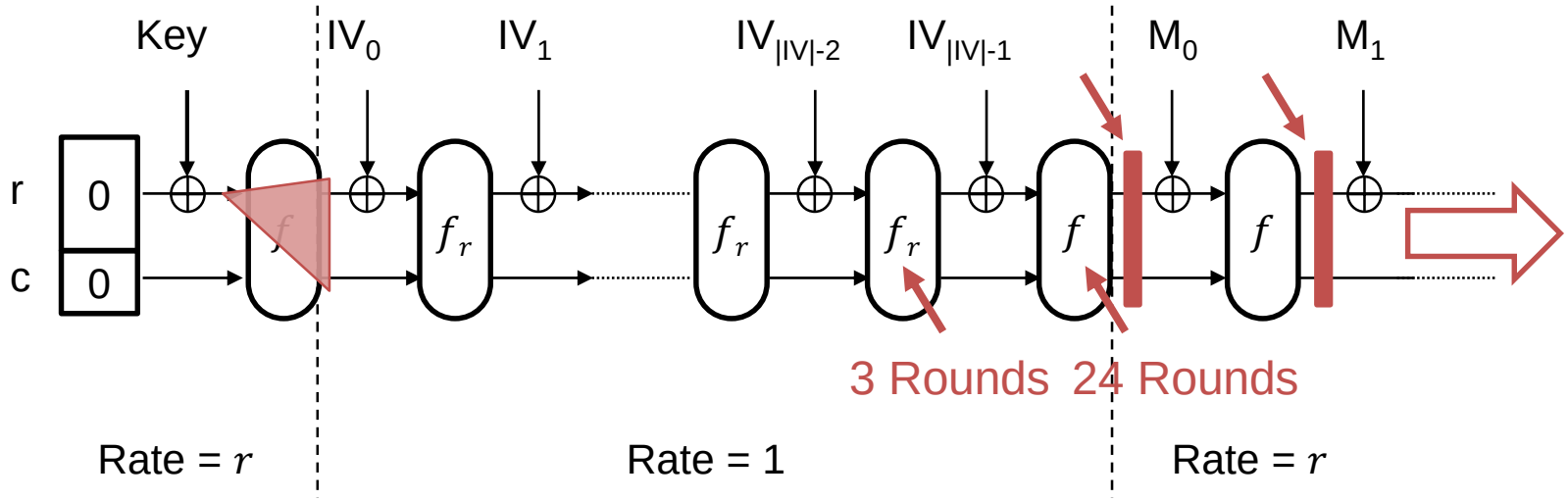
1. The Key goes t… …he secret
2. Mandate the us… …nce
3. Squeeze the ra…
4. Use Round-Re… …xcept the last bit.
5. Then, proceed …





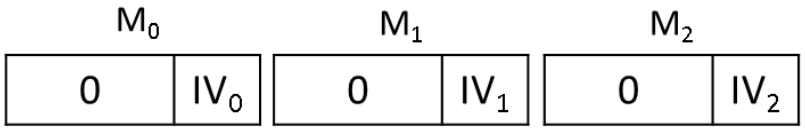
SHA-3 Countermeasure, WHY?

1. The Key goes to a separate input. Increase size of the secret
2. Mandate the use of IV in all keyed applications. Nonce
3. Squeeze the rate to “one bit” during IV.
4. Use Round-Reduced version of Keccak during IV, except the last bit.
5. Then, proceed normally.



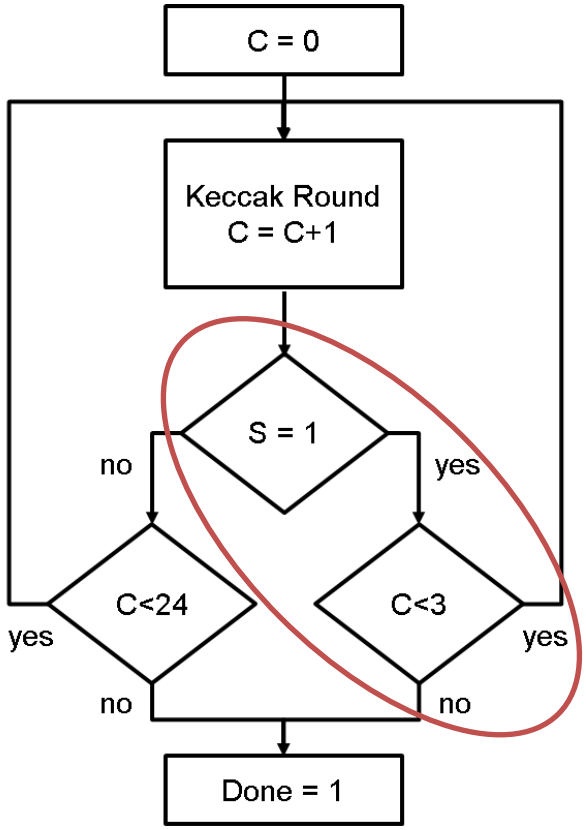
SHA-3 Countermeasure

- Implementation
  - Rate Reduction



  - Round-Reduced Keccak

    Using 2 Gates at 3.7 GE
    (Synopsys Design Compiler at 130nm technology)



SHA-3 Countermeasure

- Implementation:
  - Performance

    $(|IV| - 1) * 3$ extra rounds

    $(|IV| - 1)/8$ extra Keccak runs

  - Trading SCA-protection for performance New!

    Use s bits of IV per step

    s is an SCA-security parameter in $[1: |IV|]$

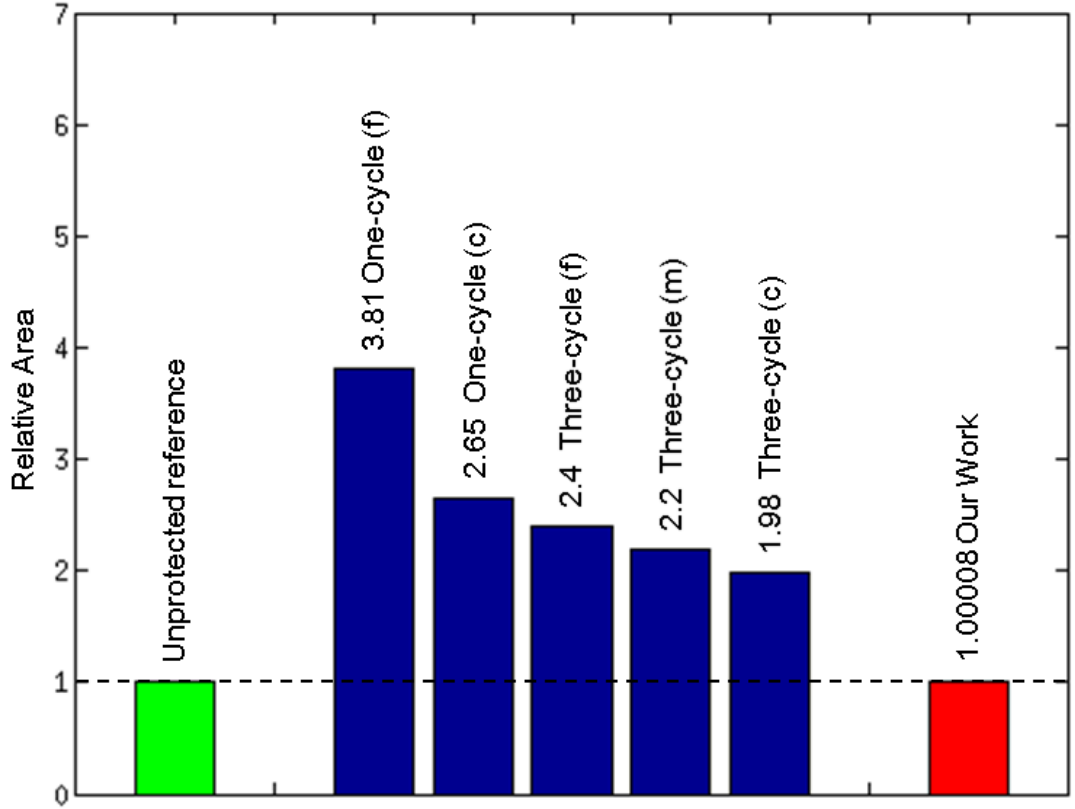
    New performance: $(|IV| - 1)/8s$ extra Keccak runs

SHA-3 Countermeasure

- Comparison :

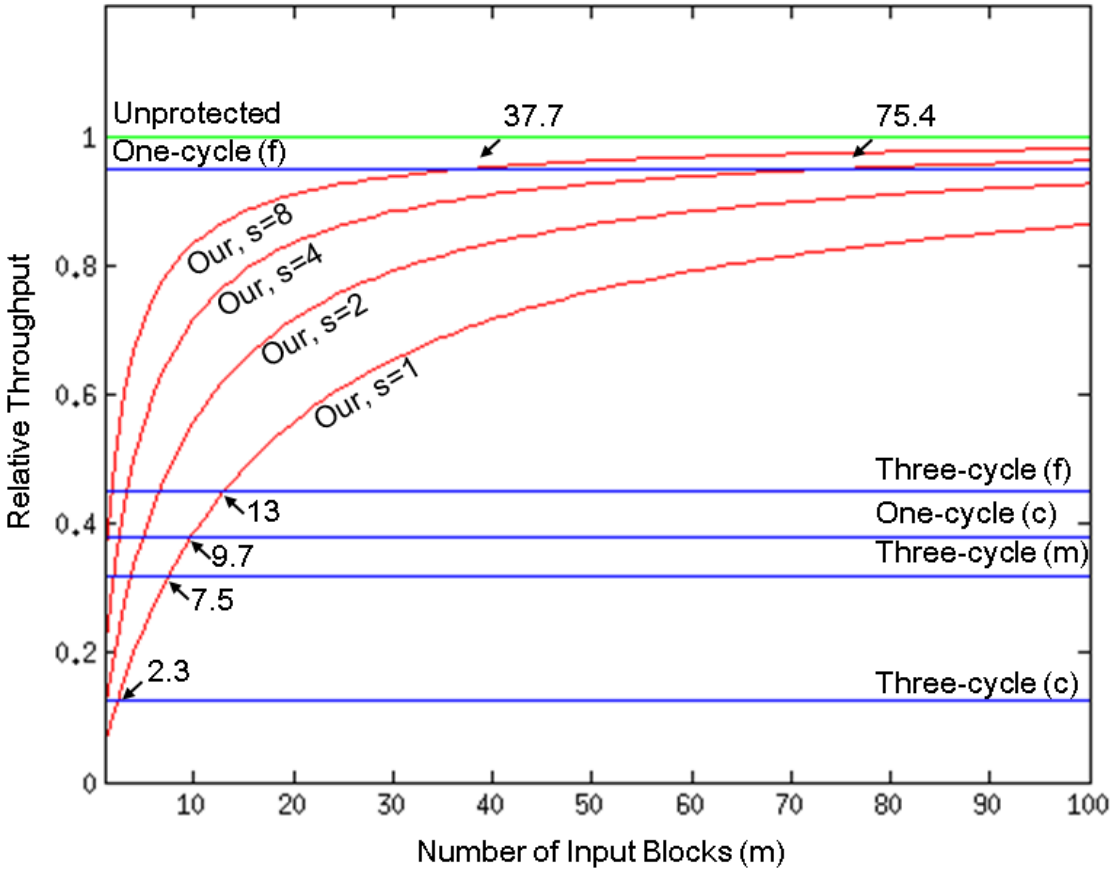
  against masked implementations of Keccak designers

  – Area



SHA-3 Countermeasure

- Comparison:
  - Performance



SHA-3 Countermeasure

- Comparison:
  - Flexibility

    SCA-protection can be tuned by software.

  - Compatibility

    Can be applied to already built modules

  - Portability

    Protect the *Sponge* with any permutation function

Outline

- Side-Channel Analysis
- SHA-3, the story and its applications
- SHA-3 Countermeasure
- New Design Concept

New Design Concept

Practical Methods

Masking & Hiding

- Fix current implementations
- Efficient solutions
- Practically verifiable

But

- Limited scope of protection
- Turns SCA harder
- Occasionally, breaks-down

Theoretical Methods

Leakage Resilient Cryptography

- Design new primitives
- Provable prevention against all differential attacks

But

- No solution to current primitives
- Very high implementation cost

Our Solution

New Design Concept

Practical Methods

Masking & Hiding

- Fix current implementations
- Efficient solutions
- Practically verifiable

But

- Lir
- Tu
- Oc

Theoretical Methods

Leakage Resilient Cryptography

- Design new primitives
- Provable prevention against all differential attacks

But

- itives
- cost

Our Solution

- Use LRC as a tool to protect current primitives
- Achieve a fine granularity of SCA-protection vs performance

Yes, our countermeasure belongs to LRC

Conclusion

- Proposed a countermeasure for the applications of SHA-3 at almost-zero area overhead.
- The performance overhead can be trivialized at long message lengths.
- The countermeasure is flexible, compatible and portable.
- Use theoretical ideas through practical experience.

Thank You

Questions?