DPA Attacks vs. unknown input: a 1st-order Attack on Counter Modes

Rump Session Talk

CHES 2006

Josh Jaffe
Cryptography Research, Inc.
www.cryptography.com

575 Market St., 21st Floor, San Francisco, CA 94105

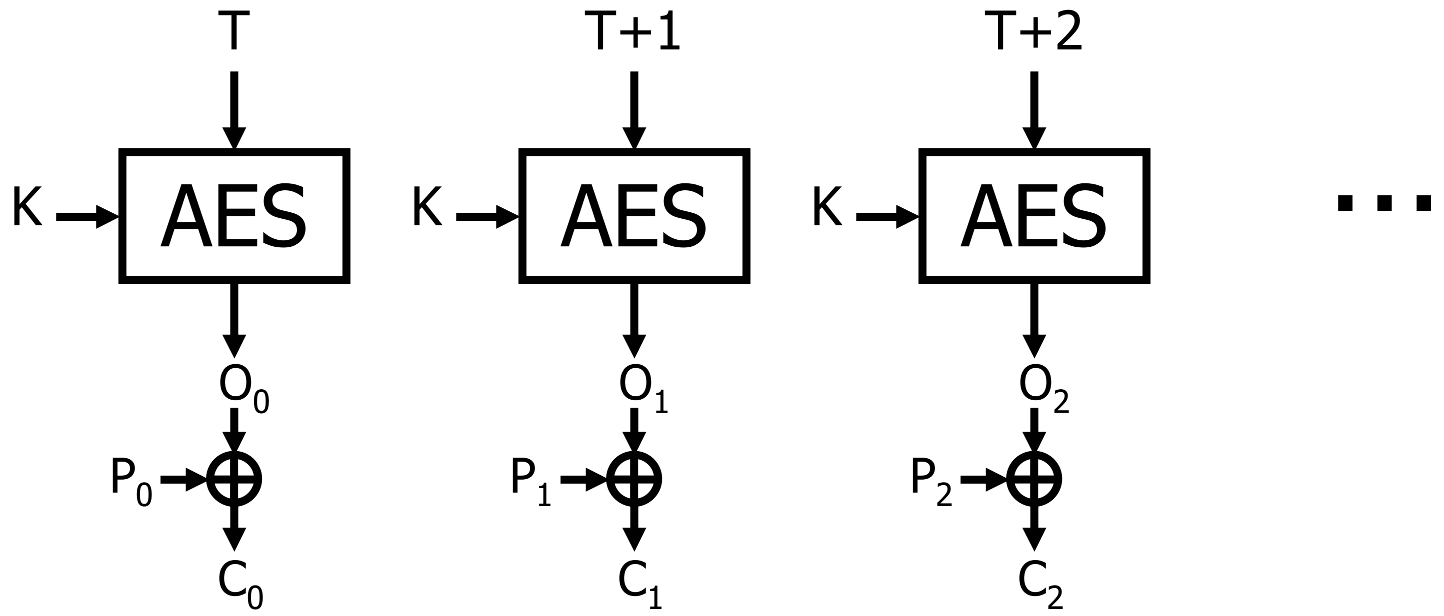
CRYPTOGRAPHY RESEARCH

Context

- This talk outlines a first-order DPA attack on ciphers used in **counter mode**, that works even if the initial counter value and cipher output are unknown.

- *When input/output are not known, high-order attacks are traditionally used.*
- *But this is undesirable (if it can be avoided) because high-order attacks need more traces…*

Review of Counter Mode



- Example construction, w/ AES
- O_i = enc(K, T+i)
- $C_i = O_i \oplus P_i$
- *Assume T, Oi, K are unknown*

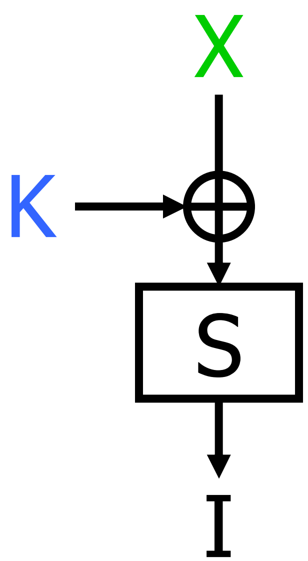
DPA Attack example

- Target: AES in counter mode with unknown input.
  - Galois counter mode, len(IV)!=96
  - Note: This attack works in general, but AES has some structural elements that are particularly helpful.
- Step 0: Collect Measurements
  - Monitor encryptions of 2^{17*} sequential (T+i).
    - * attack could use fewer messages too. E.g. 2^8.
    - Also, I'm skipping over the fact that you could use an SPA or 1st-order DPA attack to find T value with low byte(s) equal to 00. The attack I actually implemented assumed I had.
  - Record power measurements covering at least the first four rounds of each encryption.

CRYPTOGRAPHY RESEARCH

DPA attack (review)

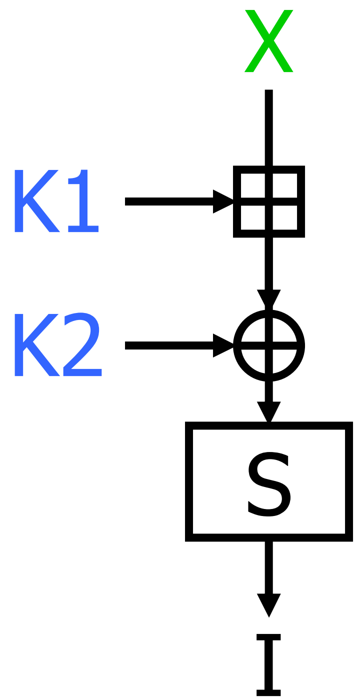
- Context for typical DPA attack (e.g. on AES):
  - Known variable X
  - Secret constant K
  - Intermediate derivative I



- Typical DPA Attack:
  - For many X, measure power P.
  - For each K, predict I_K.
  - Calculate $\Delta P/\Delta I_K$.
  - Test: $\text{abs}(\Delta P/\Delta I_K) >> \text{noise}$?
  - "Yes" indicates that a value correlated to I_K is present; suggests that K is correct.

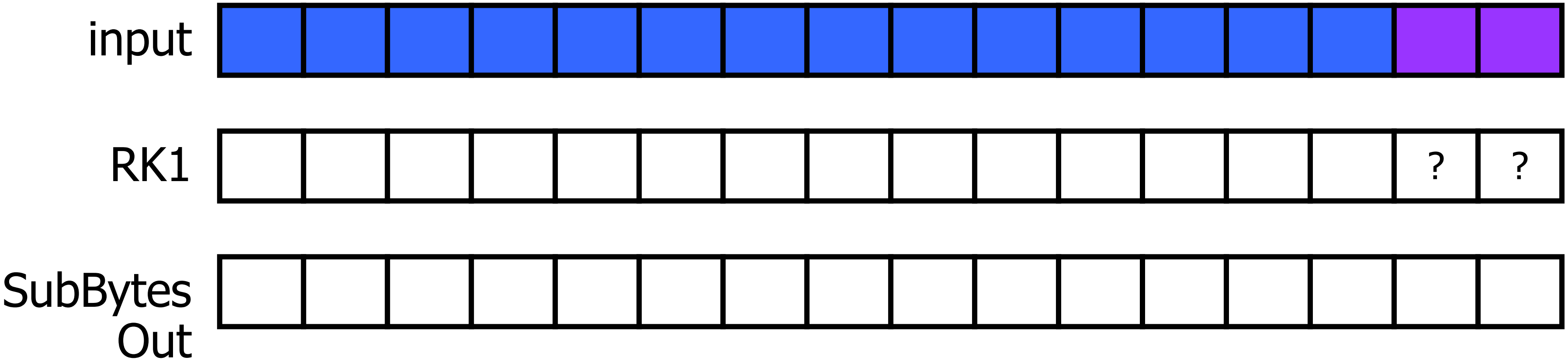
Analysis (round 1)

- In counter mode the input is T + i, where T is unknown.
- We want a known input X for the DPA attack.
- Solution: Let i be our known 'X'.
  - T is secret, so lets rename it 'K1'.
- We can now perform a DPA attack on this construction:
  - I = S[(X + K1) ⊕ K2]
  - known X
  - guess K1, K2 and predict I



CRYPTOGRAPHY RESEARCH

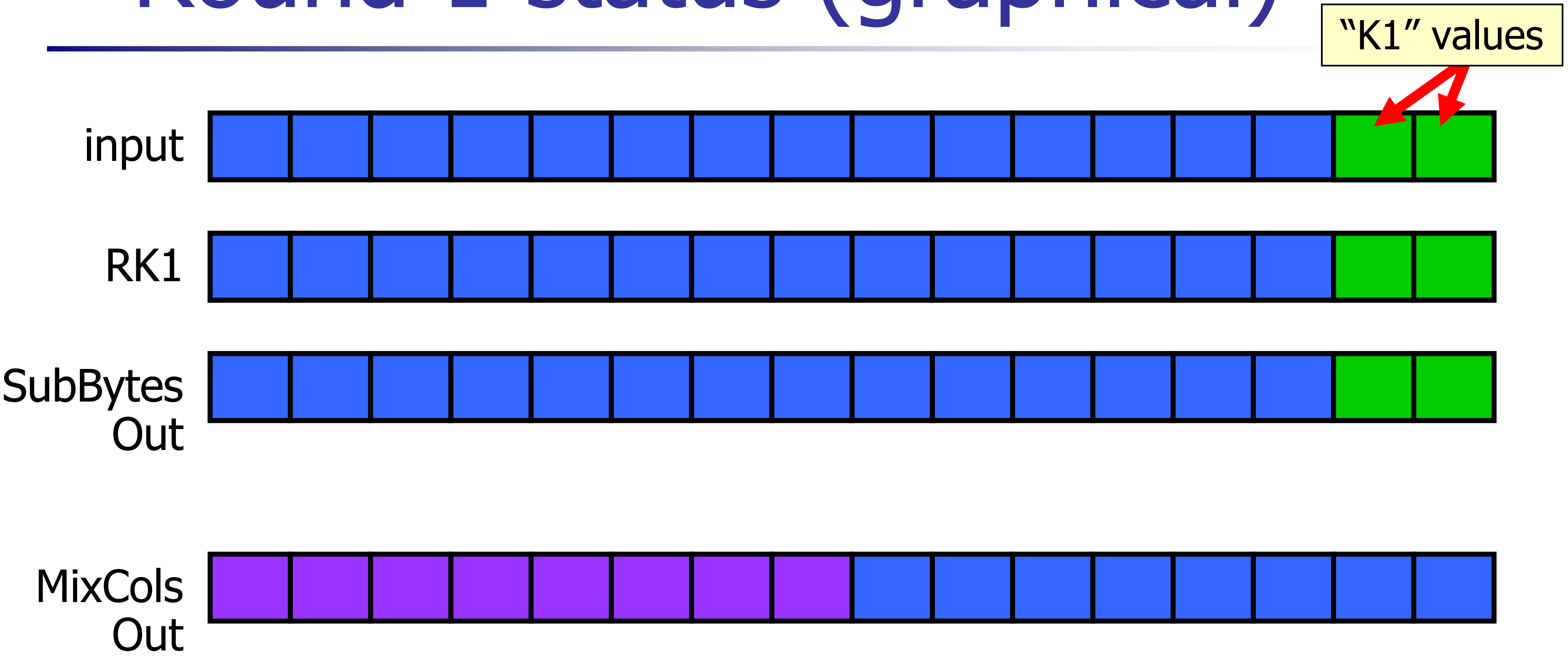
Round 1 status (graphical)



Legend

- Secret Constant (blue)
- Known value (green)
- An unknown variable equal to the XOR or SUM of a known variable with a secret constant. (purple)

Round 1 status (graphical)



Legend

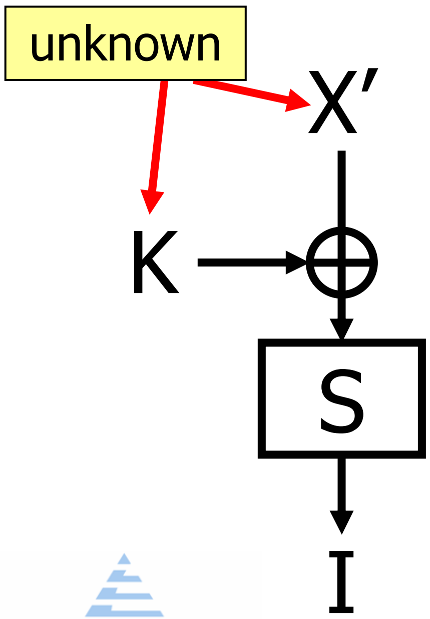
- Secret Constant (blue)
- Known value (green)
- An unknown variable equal to the XOR or SUM of a known variable with a secret constant. (purple)

DPA attack, round 2

- Round 1 attack yields bytes 15 and 16 of the round key and corresponding "K1" values.
  - Correct S-output bytes 15 and 16.
  - SKIP OVER THE REST OF ROUND 1.
    - Assume all unknown constant bytes of input and RK1 are ZERO.
    - Result: a constant error XORed onto MixCols out!
- In Round 2, input block is then:
  - [8 masked bytes] || [8 unknown, constant bytes]
  - Masked bytes X' can be expressed as the XOR of a known value X and an unknown constant C.
  - $X' = X \oplus C$.

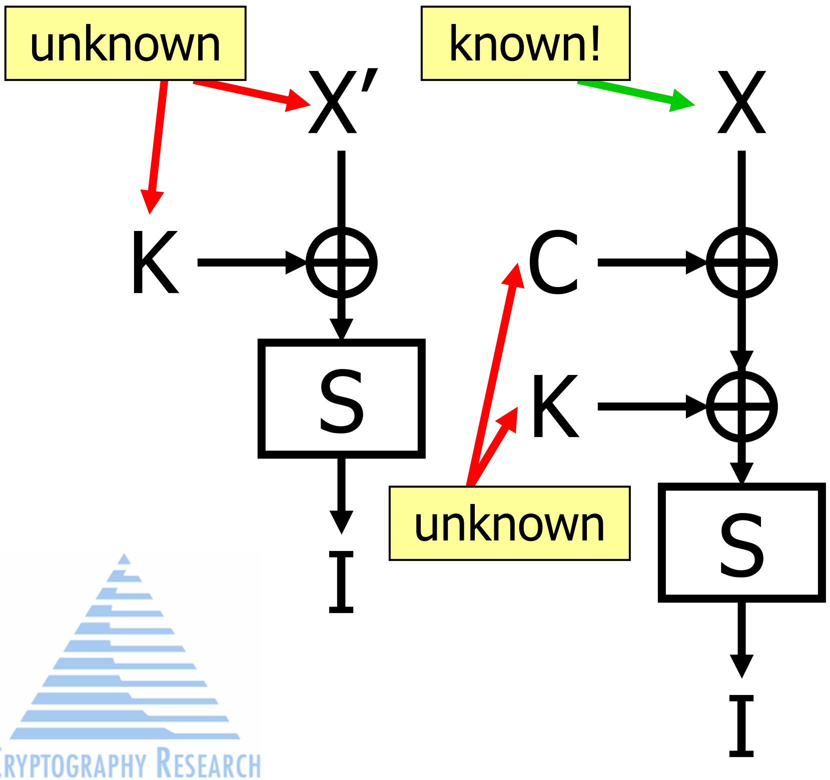
DPA attack in round 2…

- The unknown C's can be pushed into K's!
  - $I = S[X' \oplus K] = S[(X \oplus C) \oplus K] = S[X \oplus (C \oplus K)] = S[X \oplus K']$



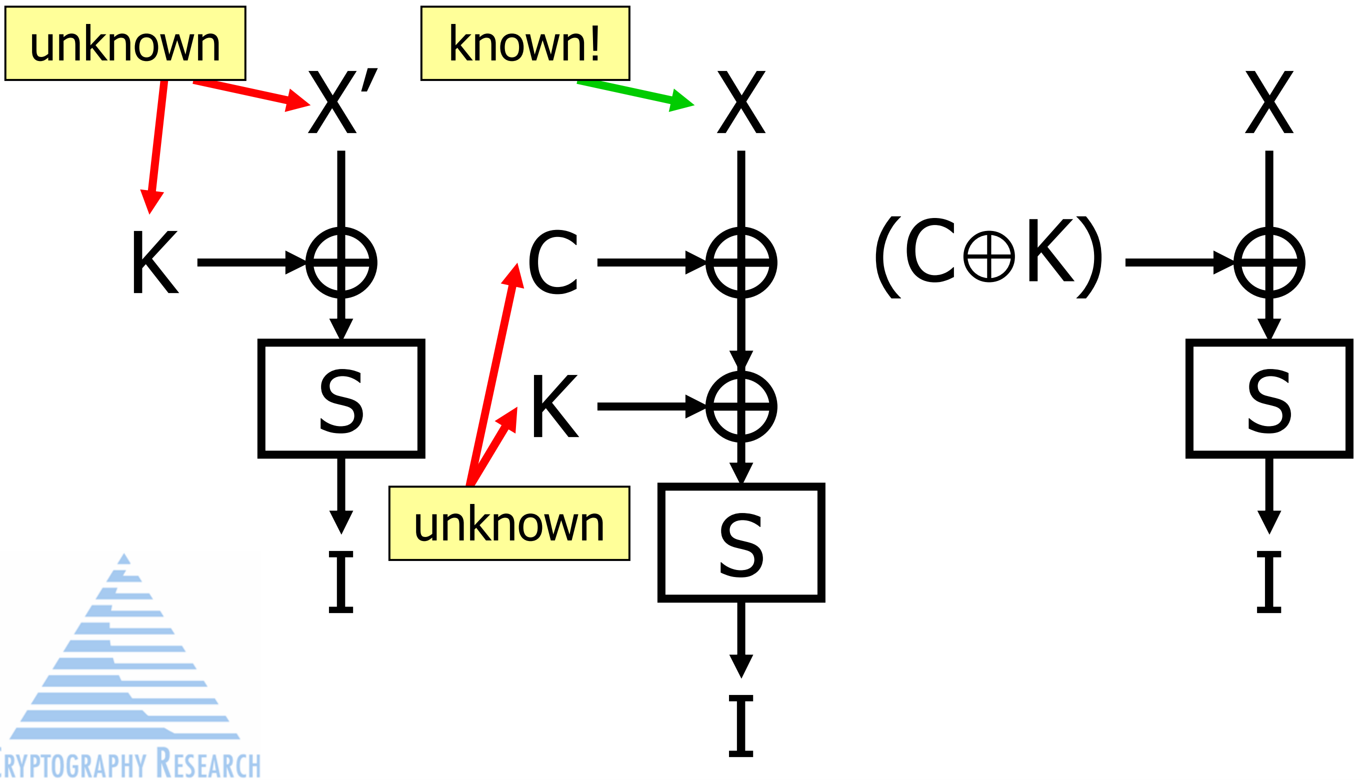
DPA attack in round 2…

- The unknown C's can be pushed into K's!
  - $I = S[X' \oplus K] = S[(X \oplus C) \oplus K] = S[X \oplus (C \oplus K)] = S[X \oplus K']$



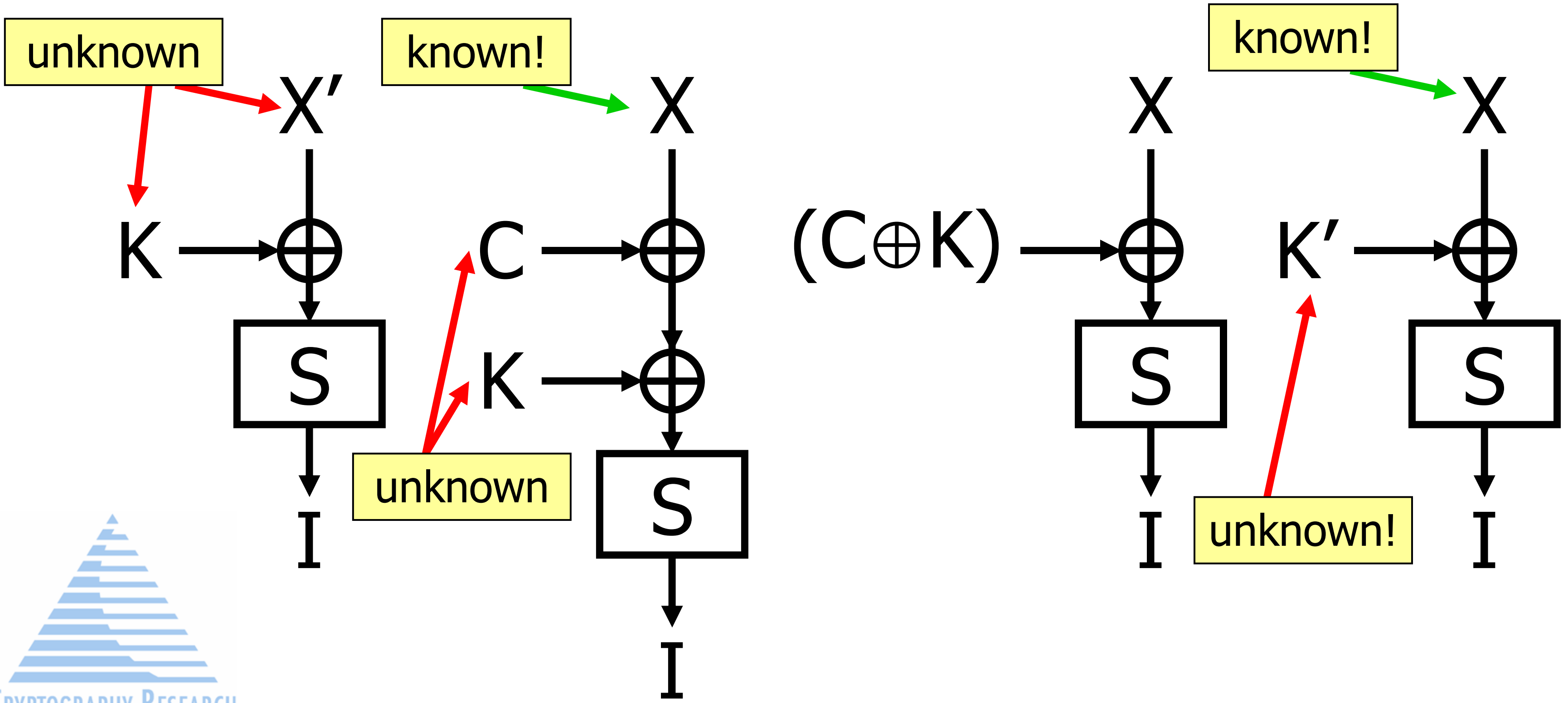
DPA attack in round 2…

- The unknown C's can be pushed into K's!
  - $I = S[X' \oplus K] = S[(X \oplus C) \oplus K] = S[X \oplus (C \oplus K)] = S[X \oplus K']$

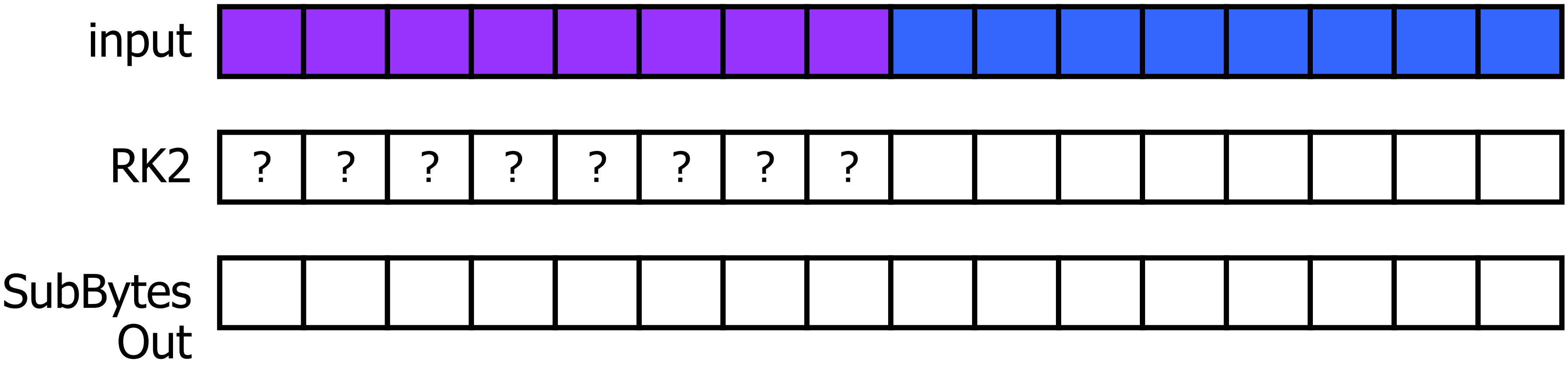


DPA attack in round 2…

- The unknown C's can be pushed into K's!
  - $I = S[X' \oplus K] = S[(X \oplus C) \oplus K] = S[X \oplus (C \oplus K)] = S[X \oplus K']$



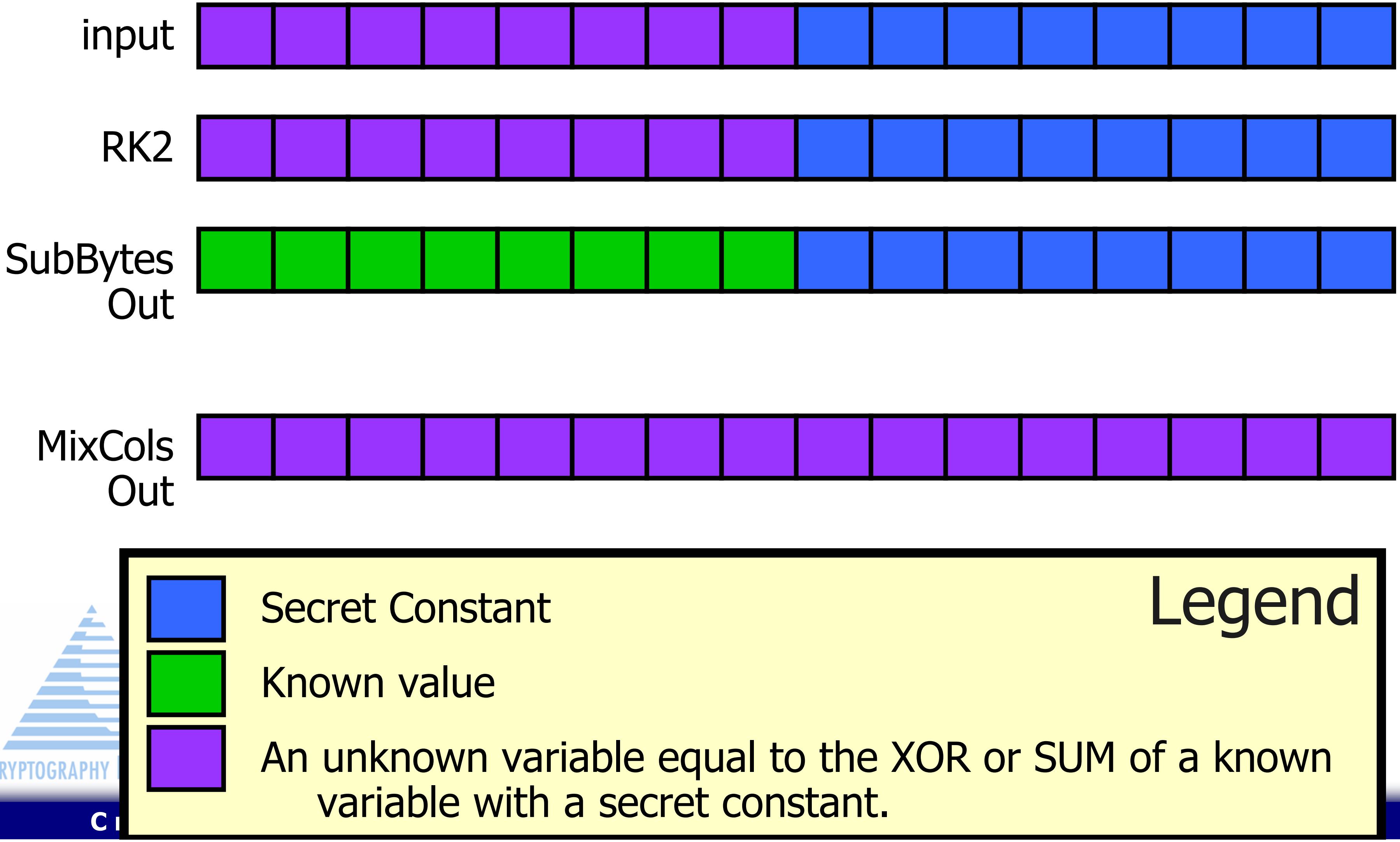
Round 2 status (graphical)



Legend

- Secret Constant
- Known value
- An unknown variable equal to the XOR or SUM of a known variable with a secret constant.

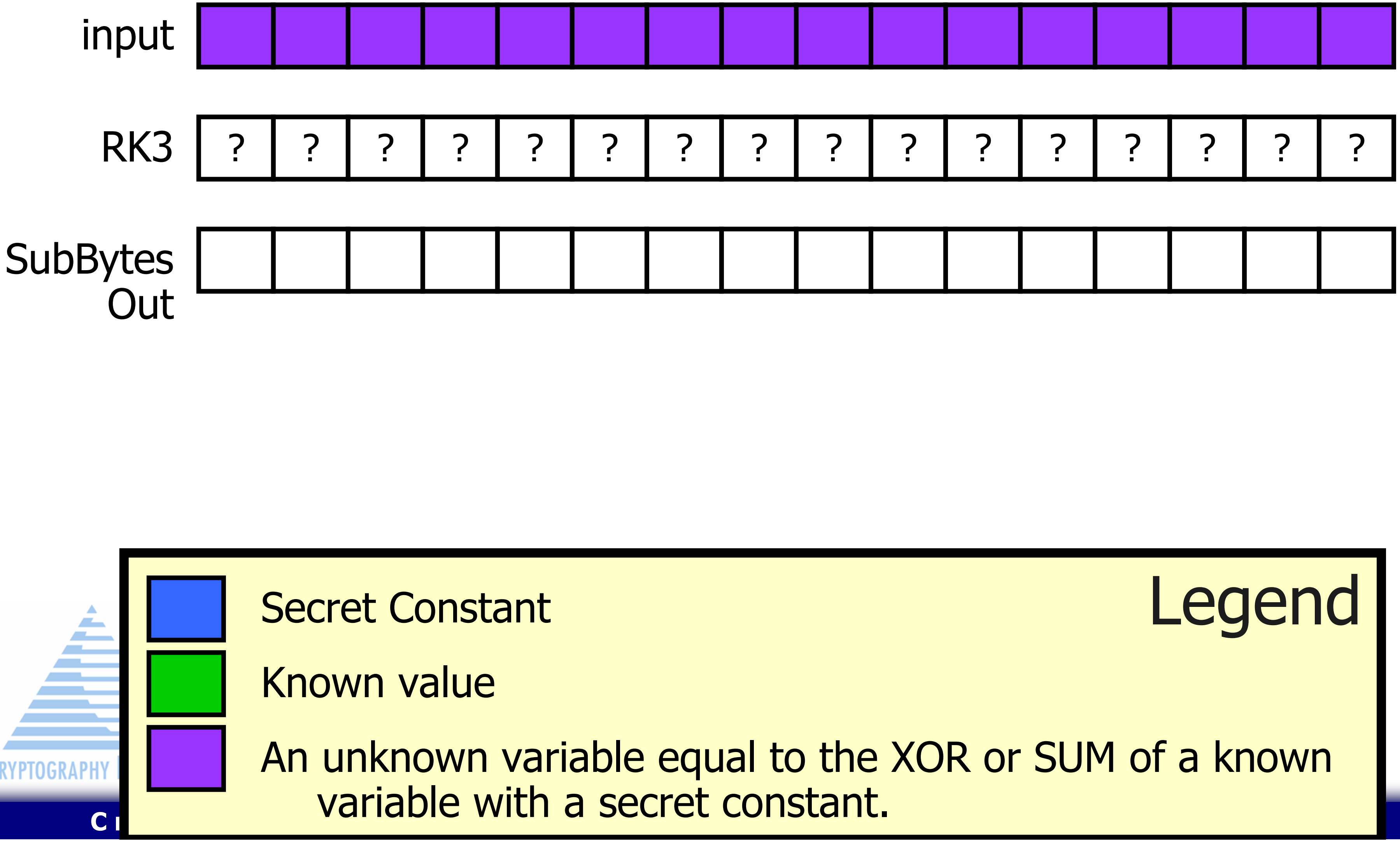
Round 2 status (graphical)



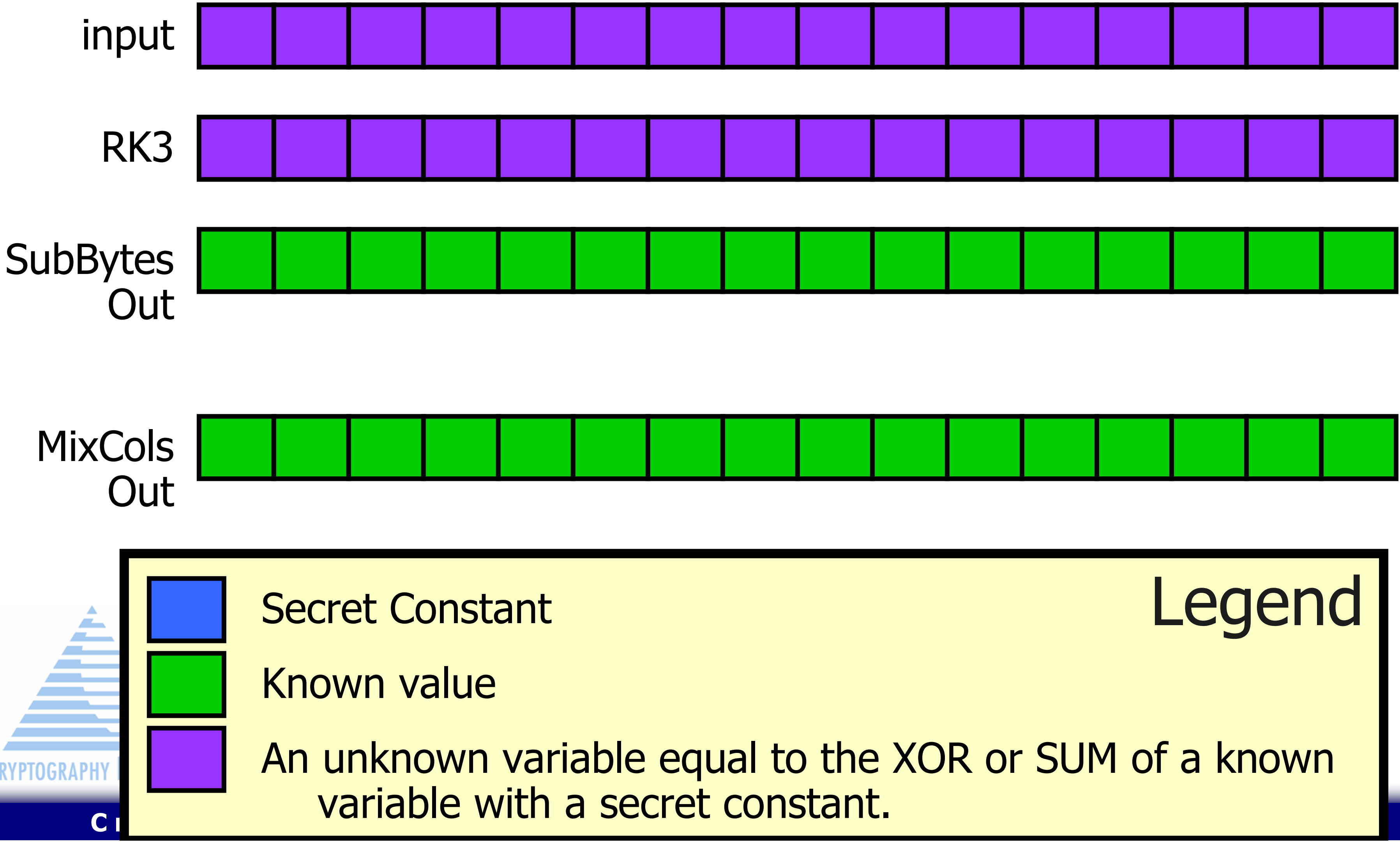
The DPA Attack, round 3

- Round 3 input block is:
  - [16 masked bytes]
  - Masked bytes X′ can be expressed as the XOR of a known value X and an unknown constant C.
  - $X' = X \oplus C$.
- As in round 2:
  - DPA attack finds $rk3' = rk3 \oplus C$.
  - S output is correct…
- But now we have ALL S-out bytes correct.
- There is no error in the MixCols step… <u>we have the **correct** input to round 4</u>.

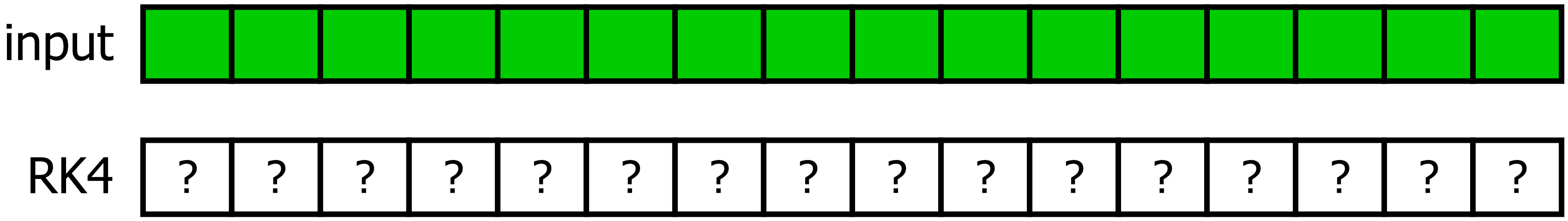
Round 3 status (graphical)



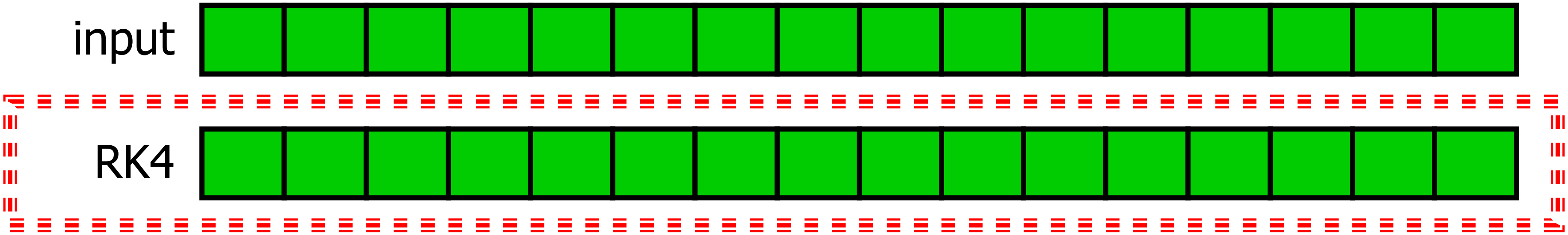
Round 3 status (graphical)



Round 4 status (graphical)



Round 4 status (graphical)



- AES-128: **DONE.**
- Find the master key by running the key schedule backwards.
- *In AES-192 & AES-256, iterate the attack one more round, then get the master key from rk4 and rk5.*

Conclusions

In counter mode, DPA attack is efficient in even when counter is not known! *High-Order attack is not needed.*

- Cipher A in counter mode can be thought of as cipher A’ with “known input” i.
  - enc_A(key = k; input = T+i) is equivalent to enc_A’(key = {k, T}; input = i).
- If you don’t know a constant, you can sometimes ignore it and clean up later – or fold it into another constant.

CRYPTOGRAPHY RESEARCH

Bonus Topic: Attacking RSA-CRT

- A simple DPA attack on RSA-CRT involves attacking a modular multiplication.
  - If X is the input, RSA-CRT manipulates X mod P and X mod Q.
  - GCD(X – (X mod P), P*Q) = P.
  - Attack goal: find X mod P for some X.
- Attack method:
  - Submit X, X+1, X+2, …
  - RSA-CRT uses these (X+i mod P) $\equiv$ (X mod P) + i
  - DPA attack by predicting mult. intermediates

CRYPTOGRAPHY RESEARCH

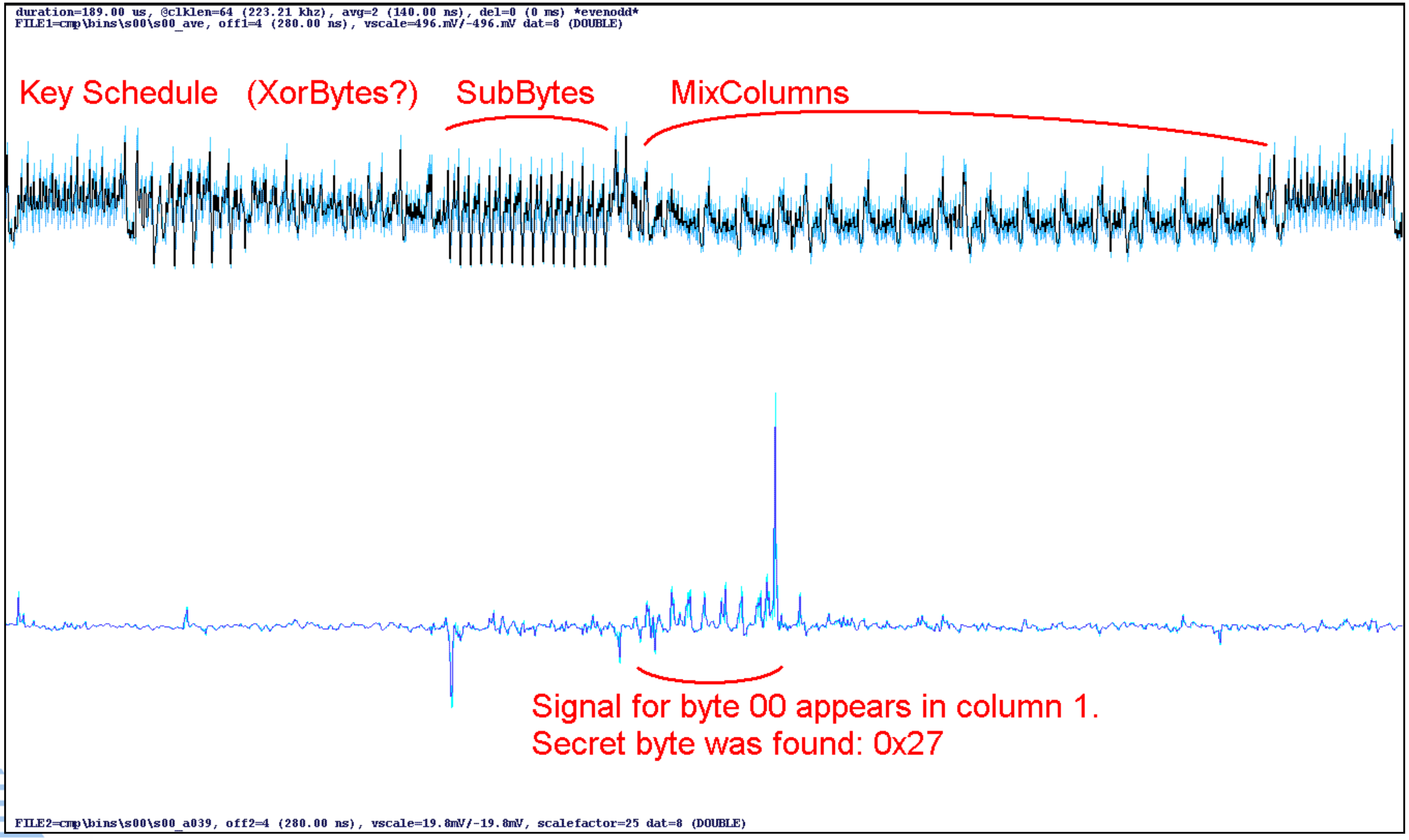
Bonus Topic: Attacking ctr output

- Example: counter mode is being used to encrypt a constant plaintext.
  - $C_i = O_i \oplus P_i$
  - Assume C_i is known.
  - Assume you can repeatedly encrypt the plaintext with different initial counters.
- Attack method:
  - Request repeated encryptions of P_0.
  - DPA attack vs. the cipher output, using C_0.
    - $C_0 = O_0 \oplus P_0 = O_0'$.
    - Use C_0 as the approximation of O_0, and roll P_0 into the key.

Real attack results…

slides lifted from another deck

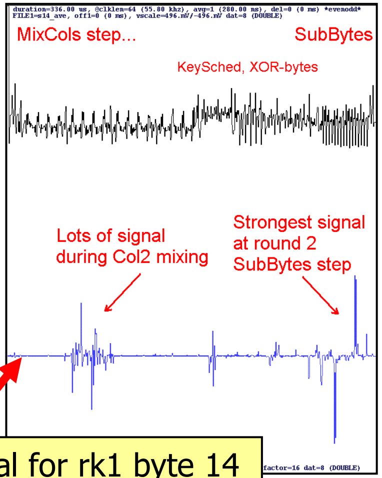
(background: AES overview trace)



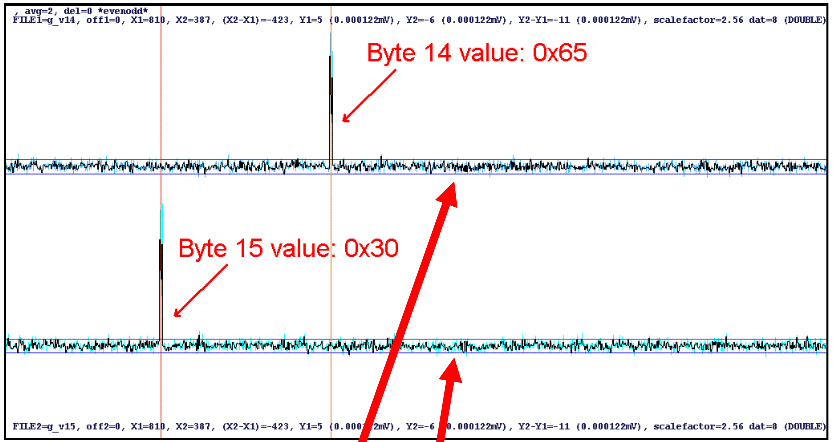
Real attack results:

- Initialized counter and key w/ unknown random values.
- Set low order 16-bits of counter to 0000h. (without loss of generality.)
- collected 65536 traces over encryptions of incrementing counter values;
  - 31.8GB compressed to 1.85GB.
- Analyzed traces…

Attack step 2. Get rk1'



DPA signal for rk1 byte 14

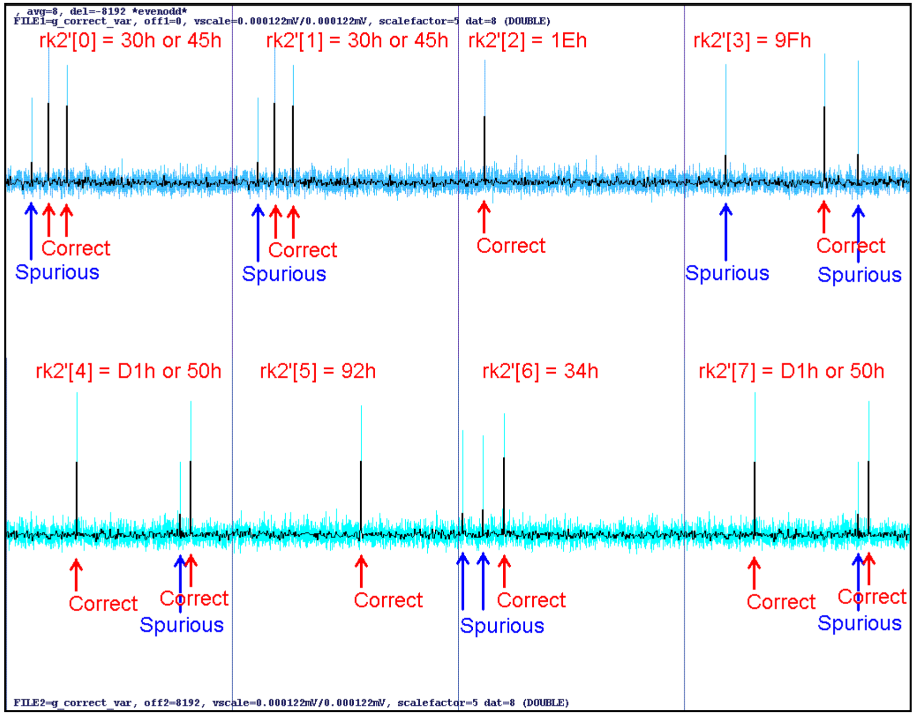


DPA results graph for bytes 14 & 15, in compact key search form.

- Message format was:
  - M = [14 secret, constant bytes] || [2 byte counter]
- And the approximation (setting unknown bytes of M to 0x00):
  - M' = 0x0000000000000000000000000000cccc
- I first determined the bytes of the key that lined up with the two varying bytes of the counter.
  - Byte 14 of rk1 is 0x65.
  - Byte 15 of rk1 is 0x30.
- I inserted 0x00 for unknown bytes in rk1 to get rk1':
  - rk1' = 0x00000000000000000000000000006530

Attack step 3. Go for rk2'

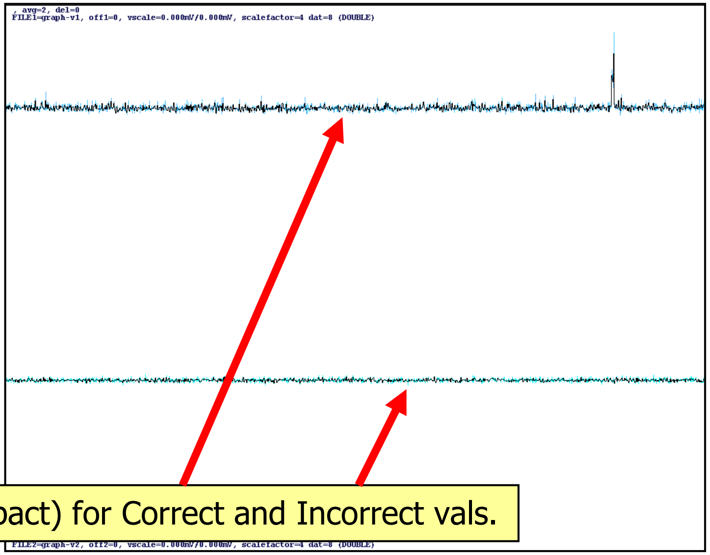
- Using rk1' and M' I calculated the input to round 2 (X').
- DPA attack using X' gave:
  - rk2'[0] = 0x30 or 0x45
  - rk2'[1] = 0x45 or 0x30
  - rk2'[2] = 0x1E
  - rk2'[3] = 0x9F
  - rk2'[4] = 0xD1 or 0x50
  - rk2'[5] = 0x92
  - rk2'[6] = 0x34
  - rk2'[7] = 0x50 or 0xD1
- Other bytes of rk2' are unknown (X' is constant)



Attack step 3a get rk2'

- Inputs to S[0] and S[1] are correlated. So are inputs to S[4] and S[7].
  - The input messages are correlated for these bytes; same byte XORed w/ different constants
  - BOTH values are correct… but for different rk2 bytes.
  - Next step is to determine which is which.
- Extra step: determine byte order (use DPA).

Key	Value for rk2'	test1?	test2?
1	45301E9FD19234500000000000000000	N	Y
2	45301E9F509234D10000000000000000	N	N
3	30451E9FD19234500000000000000000	Y	Y
4	30451E9F509234D10000000000000000	Y	N

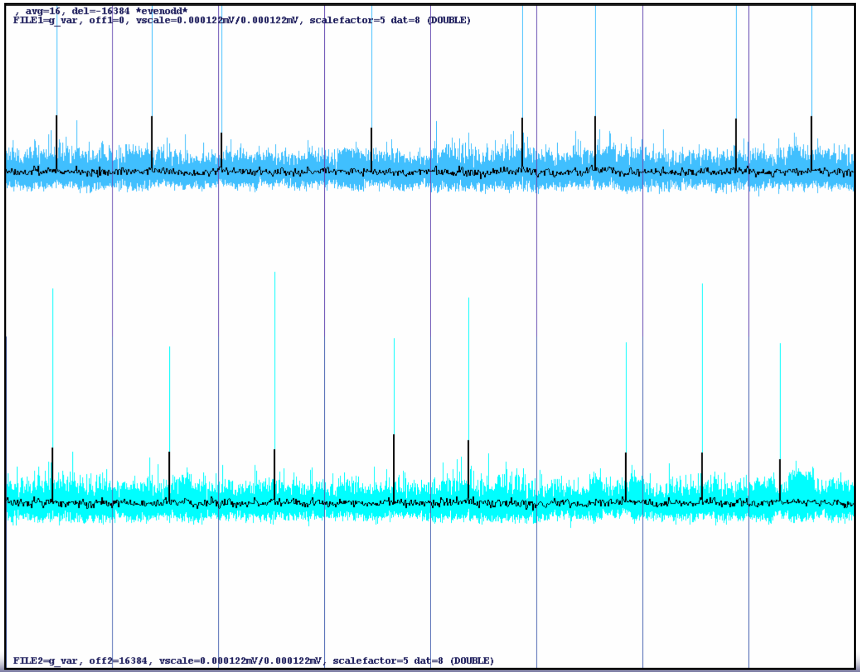


Verification step results (compact) for Correct and Incorrect vals.

CRYPTOGRAPHY RESEARCH

Attack step 4

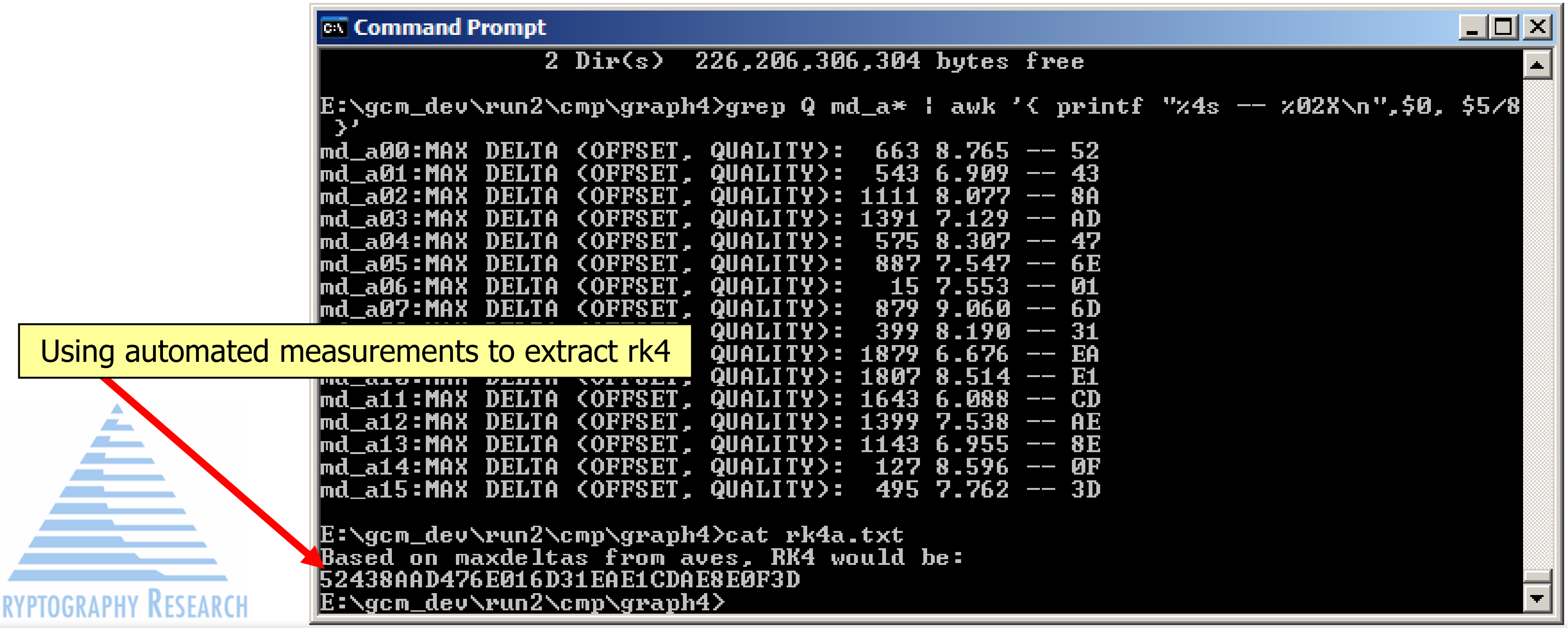
- Pad unknown bytes in rk2′ with zeros:
  - rk2′=0x30451E9FD19234500000000000000000
- Use rk2′ to predict input to round 3 (i3′), then attack rk3 using i3′.
- Result:
  - rk3′ = 0x7A610872DE8FE299708A89A85DD9914D



Cryptography Research

Attack step 5

- Given fully variable i3′ and rk3′, compute correct inputs to round 4 – and attack rk4.
- Result:
  - rk4 = 0x52438AAD476E016D31EAE1CDAE8E0F3D



Finishing the Attack

- Invert AES key schedule to find the base key…
  - `rk4 = 0x52438AAD476E016D31EAE1CDAE8E0F3D`
  - `rk3 = 0x156B0676152D8BC07684E0A09F64EEF0`
  - `rk2 = 0xF6C0556800468DB663A96B60E9E00E50`
  - `rk1 = 0xCC8D5116F686D8DE63EFE6D68A496530`
  - `KEY = 0xCC8D5116F686D8DE63EFE6D68A496530`
    - *Bonus step: find out the input counter value. For any message, take the* ***data*** *value in round 4 and run the rounds backwards to find the input.*

CRYPTOGRAPHY RESEARCH

Conclusions

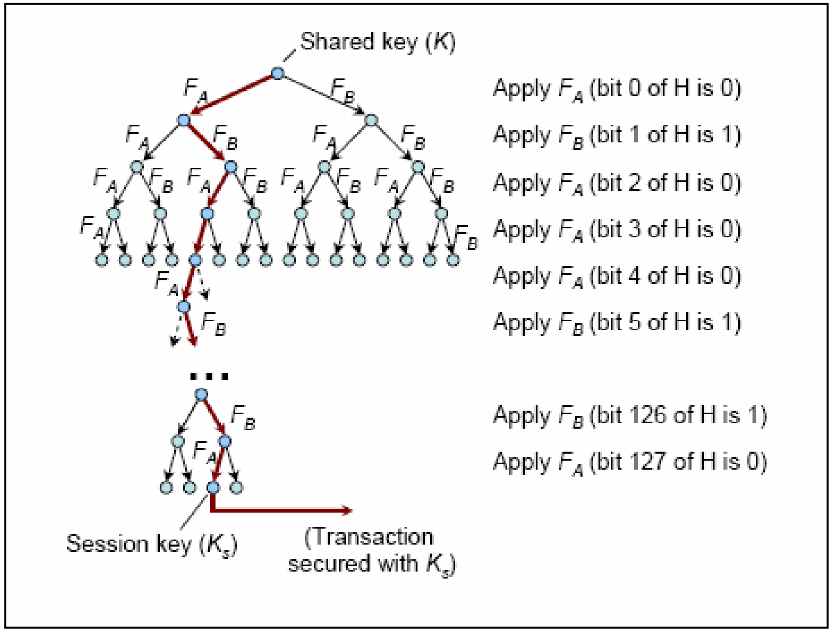
CONCLUSIONS (1/2)

- DPA Countermeasures will prevent all of these attacks
  - If the implementation is DPA-secure against chosen message attacks, then it will be secure when used in counter mode.
  - If it is NOT DPA-secure against chosen message attacks, then restricting AES input to a counter (i.e. using GCM) does not significantly increase the number of messages needed to extract the key.
    - This is true even when the initial counter value is not known.
    - Surprising result: High-Order attack is not required against AES/GCM, even if AES input is unknown.

CRYPTOGRAPHY RESEARCH

CONCLUSIONS (2/2)

- There are constructions that are secure even when the AES implementation is not DPA-secure against chosen message attacks.
- Example:
  - Kocher, "Design and Validation Strategies for Obtaining Assurance in Countermeasures to Power Analysis and Related Attacks", 2005 http://www.cryptography.com/resources/whitepapers/DPAValidation.pdf
  - tinyurl: http://tinyurl.com/k9fhe



END

(fin)