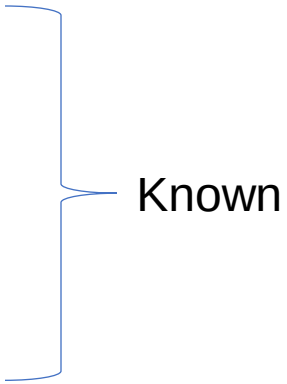
Simple Power Analysis (SPA)

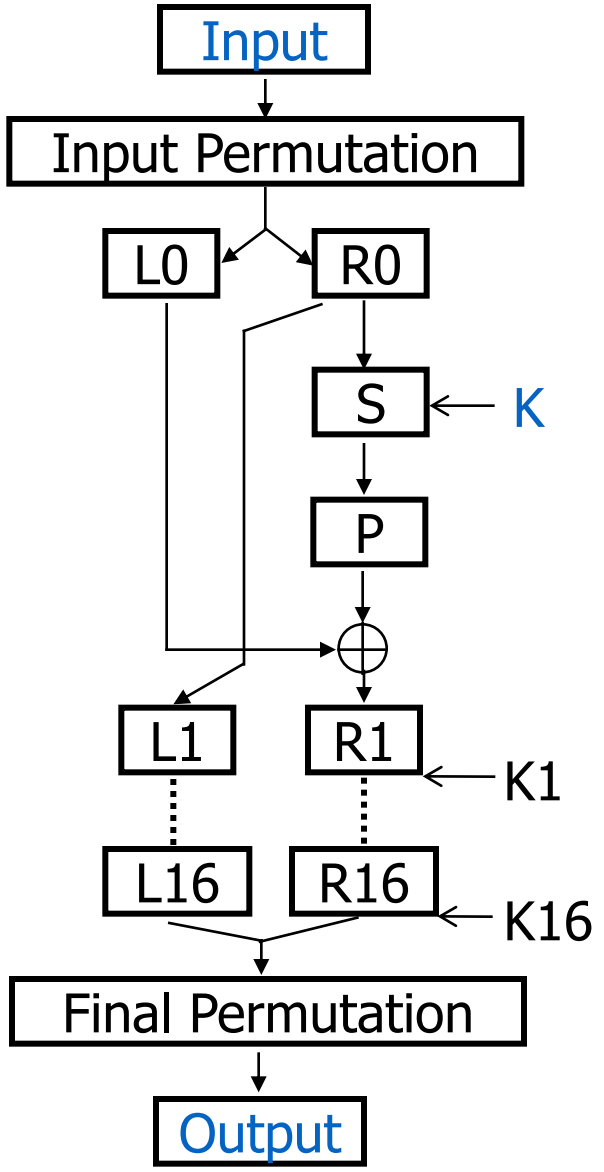
- Originally proposed by Paul Kocher, 1996
- Monitor the device's power consumption to deduce information about data and operation
- Example: SPA on DES – smart cards
  - The internal structure is shown on the next slide
- Summary of DES – a block cipher
  - a product cipher
  - 16 rounds iterations
    - substitutions (for confusion)
    - permutations (for diffusion)
  - Each round has a *round key*
    - Generated from the user-supplied key



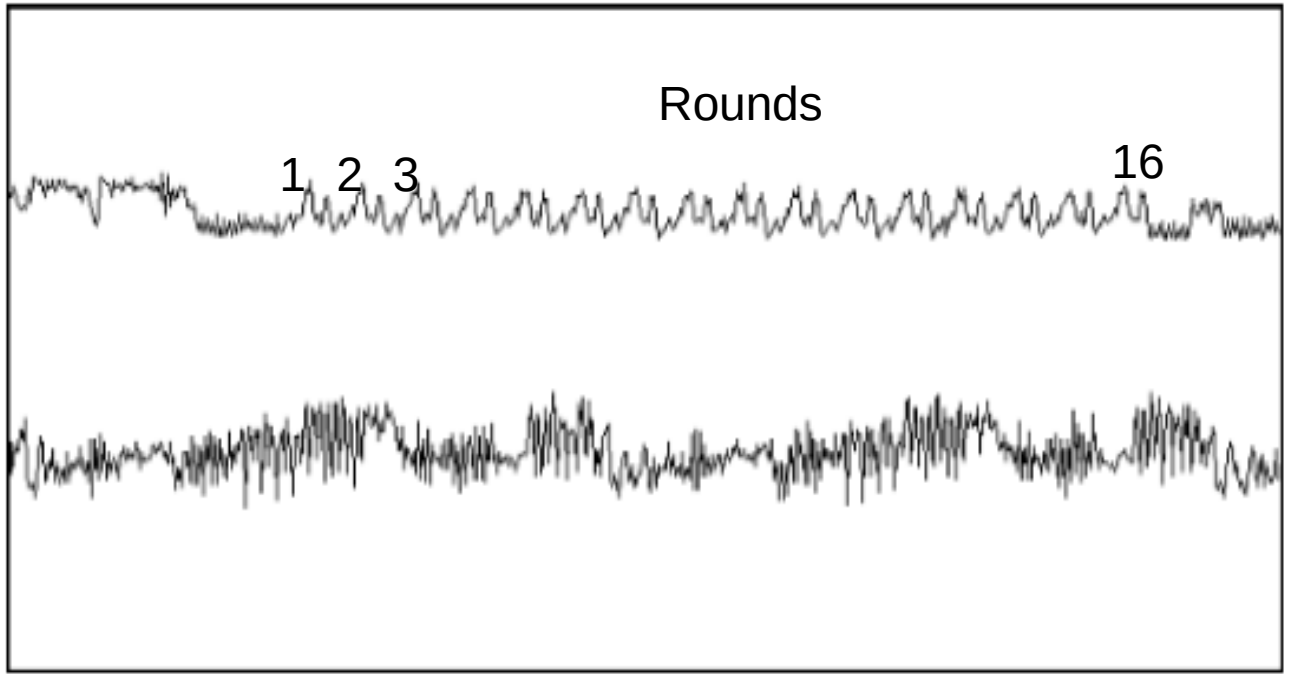
DES Basic Structure

- Input: 64 bits (a block)
- Li/Ri– left/right half (32 bits) of the input block for iteration i– subject to substitution S and permutation P
- K - user-supplied key
- Ki - round key:
  - 56 bits used +8 unused

    (unused for encryption but often used for error checking)
- Output: 64 bits (a block)
- Note: Ri becomes L(i+1)
- All basic op's are simple logical ops
  - Left shift / XOR



SPA on DES (cont'd)



- The upper trace – entire encryption, including the initial phase, 16 DES rounds, and the final permutation
- The lower trace – detailed view of the second and third rounds
- **The power trace can reveal the instruction sequence**

SPA

- SPA can be used to break cryptographic implementations (execution path, instruction, key change, etc.)
  - **DES key schedule:** Involves rotating 28-bit key registers
  - **DES permutation:** involves conditional branching
  - The DES structure and 16 rounds are known
  - Instruction flow depends on data → power signature
  - **Comparison:** Involves string and memory comparison operations performing a conditional branch when a mismatch is found
- SPA Countermeasure:
  - Avoid procedures that use secret intermediates or keys for conditional branching operation

SPA for other encryption techniques

- AES is another private encryption technique that includes a data mixing step.
- RSA is a public key encryption technique that involves modulo exponents.
- Example: Modular exponentiation in DES is often implemented by square and multiply algorithm
- Then, the power trace of the exponentiation can directly yields the corresponding value
- All programs involving conditional branching based on the key values are at risk!

square and multiply algorithm

```
exp1(M, e, N)
{  R = M
   for (i = n-2 down to 0)
   {  R = R^2 mod N
      if (ith bit of e is a 1)
         R = R·M mod N }
   return R }
```