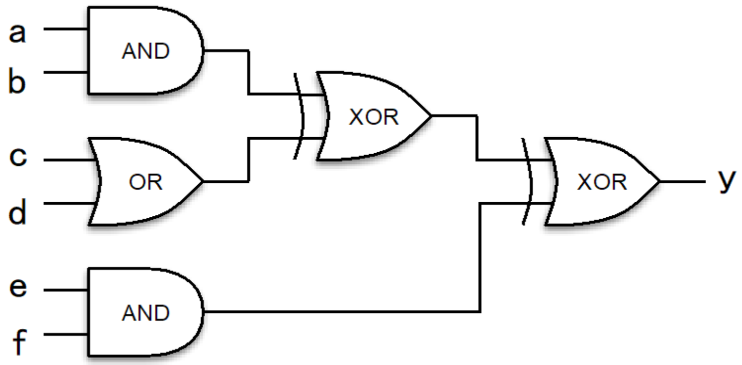
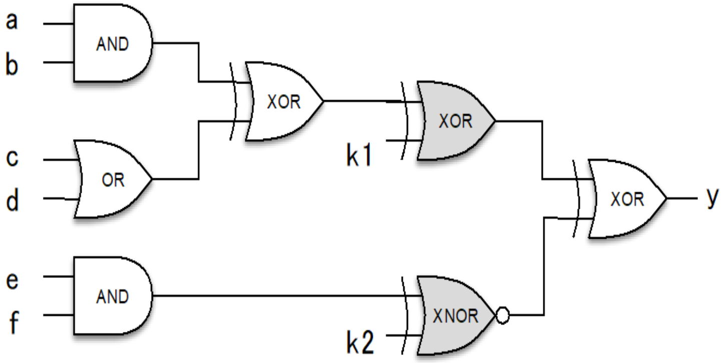
A Comparative Investigation of Approximate Attacks on Logic Encryptions

Hai Zhou (joint w/ Shen and Rezaei)
EECS, Northwestern University

Logic Encryption

- Central technique for hardware security
- Many years’ research



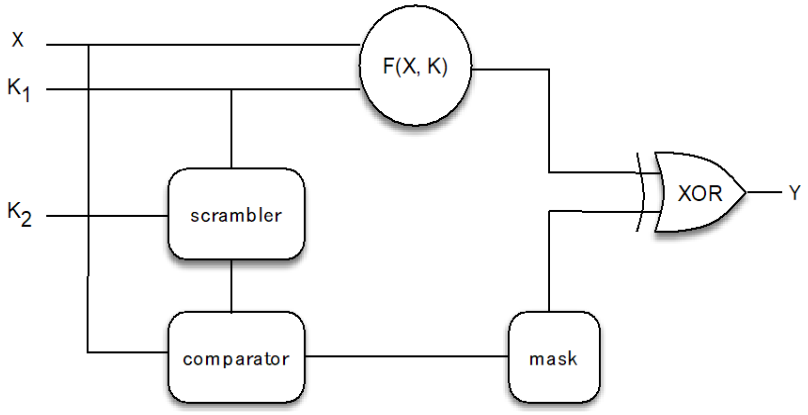


SAT-based Attack

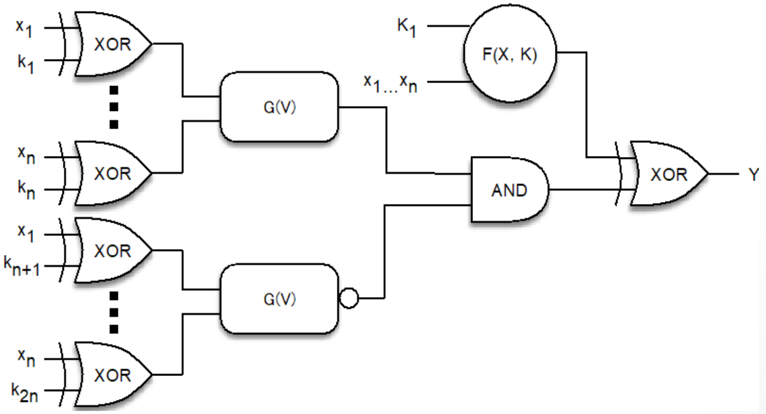
- Corrupted all existing logic encryption algorithms up to 2015
- **Idea**: use SAT solver to iteratively find DIPs and their correct outputs to prune out wrong keys
- Only need a small number of DIPs to exclude all wrong keys.

SAT-proof techniques

- Enhancing methods such as SARLock and Anti-SAT
- **Idea**: make the number of iterations exponential.



SARLock



Anti-SAT

Approximate Attack

- Approximate attack generates an approximate key instead of correct key.
- Characteristics of approximate key:
  - The error rate is exponentially small (only one or few inputs).
  - Approx attack = Exact attack + Stealthy Trojan insertion

Approximate Attack

Correct Key vs. Approximate key

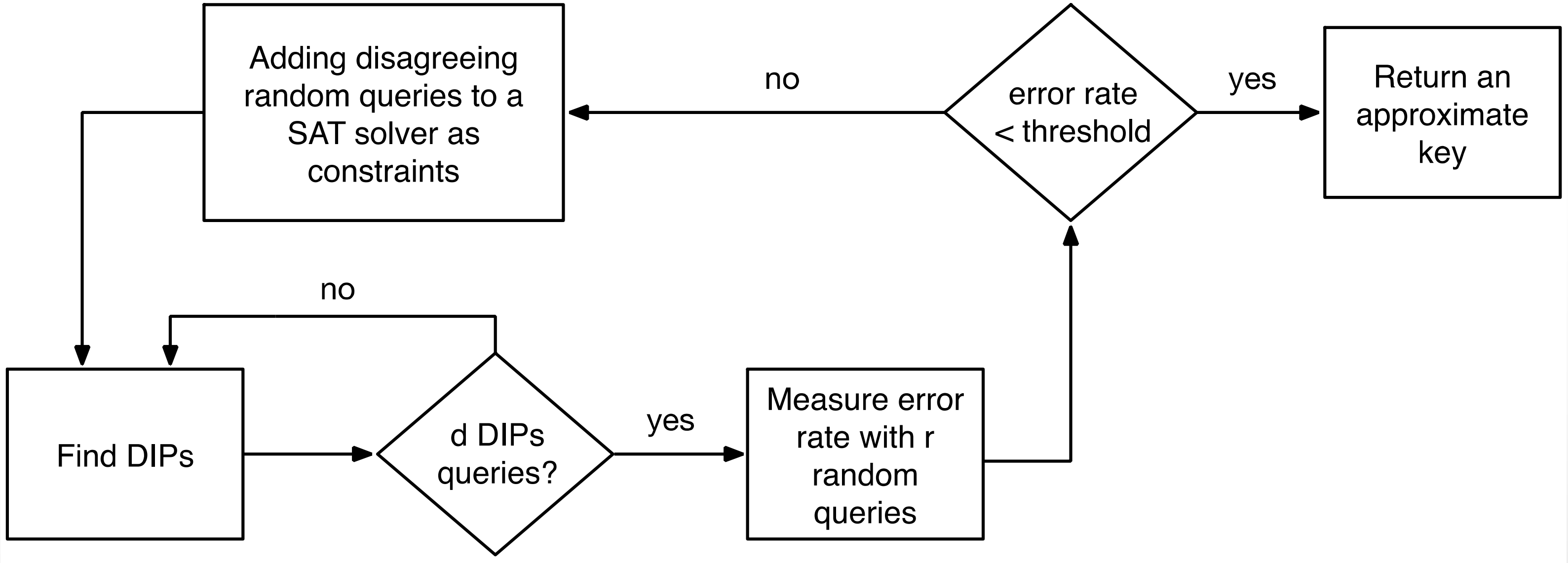
- Correct key: economic loss
- Approximate key: economic loss + threats!

Approximate Attacks

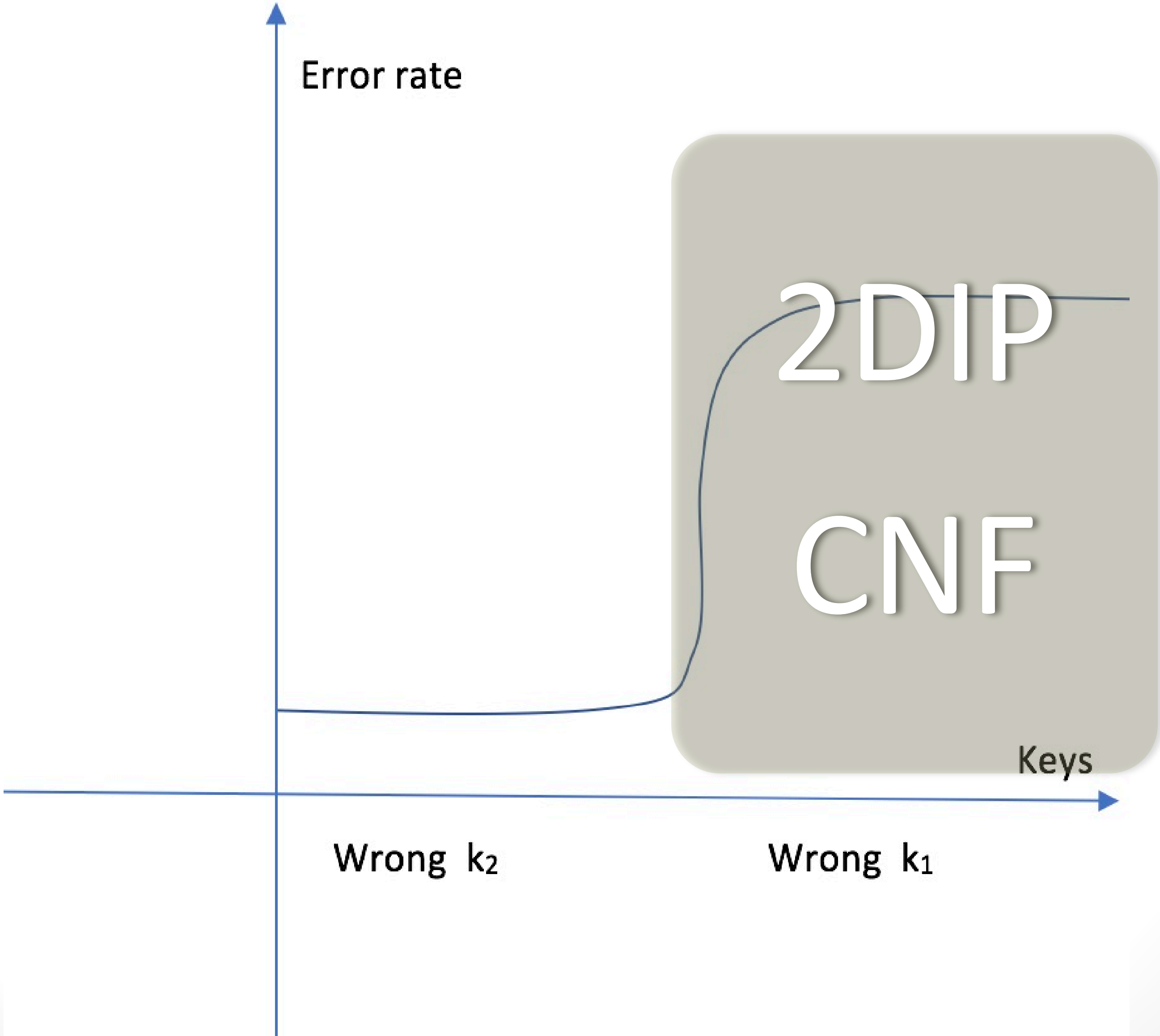
- Double DIP (Shen & Zhou 17)
    - **Goal**: find a correct traditional logic encryption key
    - **Key Idea**: instead of finding a DIP, find 2DIP (doubly differentiating input pattern) in each iteration
    - **Result**: guarantee a correct traditional key

Approximate Attack

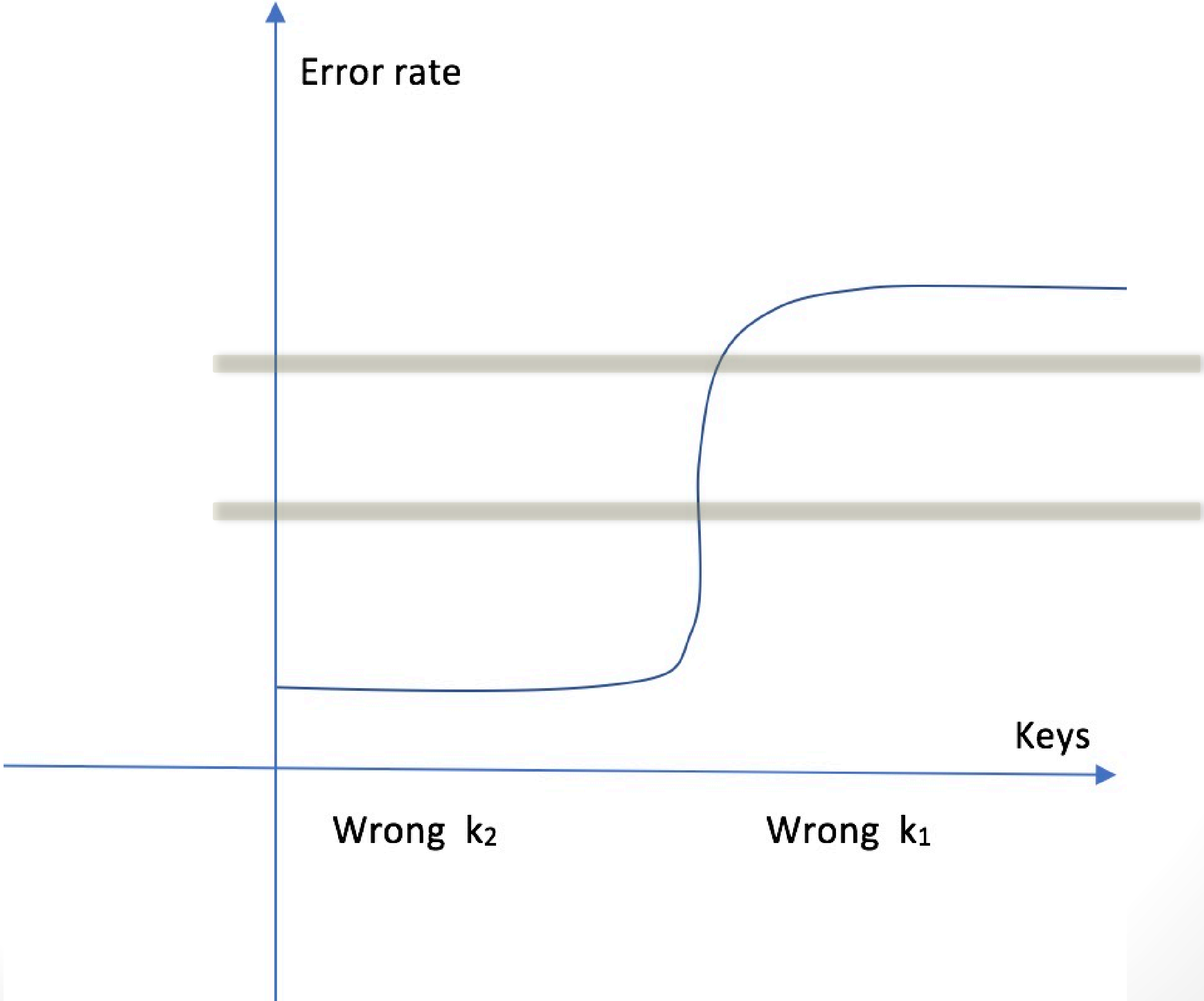
- AppSAT (Shamsi et al 17)
  - Combination of SAT-based attack and random sampling
  - Find a key that estimated error rate is below a threshold


Adding disagreeing random queries to a SAT solver as constraints
no
error rate < threshold
yes
Return an approximate key
no
Find DIPs
d DIPs queries?
yes
Measure error rate with r random queries

How do they work?—2DIP

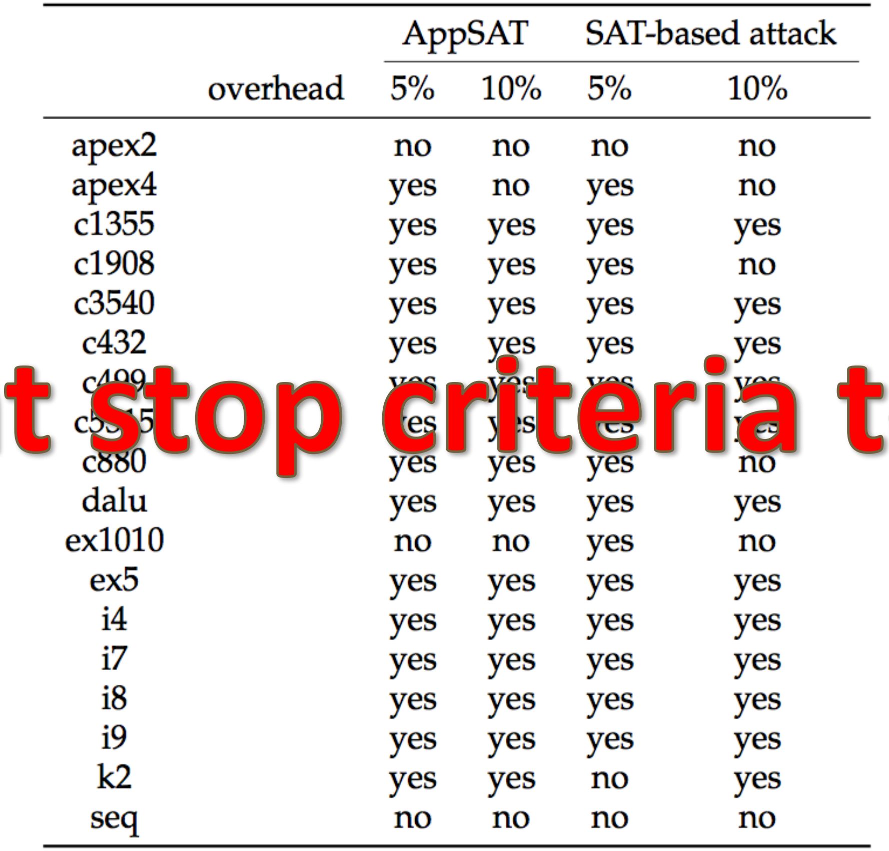


How do they work?—AppSAT



AppSAT is close to SAT

- Same #iterations of SAT will get same result



	AppSAT		SAT-based attack	
overhead	5%	10%	5%	10%
apex2	no	no	no	no
apex4	yes	no	yes	no
c1355	yes	yes	yes	yes
c1908	yes	yes	yes	no
c3540	yes	yes	yes	yes
c432	yes	yes	yes	yes
c499	yes	yes	yes	yes
c5315	yes	yes	yes	yes
c880	yes	yes	yes	no
dalu	yes	yes	yes	yes
ex1010	no	no	yes	no
ex5	yes	yes	yes	yes
i4	yes	yes	yes	yes
i7	yes	yes	yes	yes
i8	yes	yes	yes	yes
i9	yes	yes	yes	yes
k2	yes	yes	no	yes
seq	no	no	no	no

What stop criteria to use?

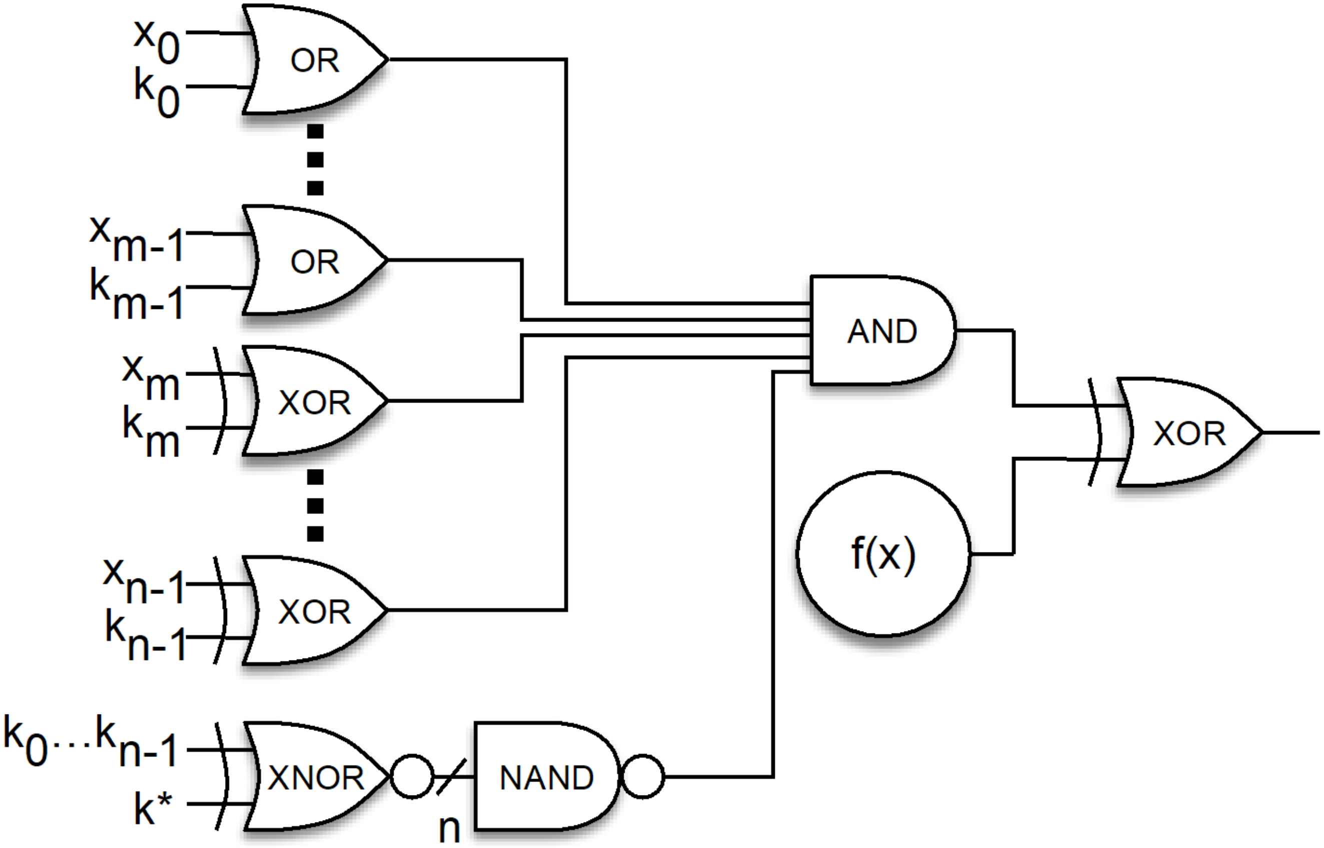
Challenges

- How are Approx Attacks performing in general?
  - SARLock (or Anti-SAT) + traditional is special
- Hard to measure performance of approx attacks
  - Computing error rate is expensive!
  - Sampling for error rate is NOT reliable!

Scientific Benchmarks

- Ideal Properties of benchmarks:
  - Different keys have different error rates
  - Error rate is known for each key
  - Error rate is adjustable
  - Benchmarks are hard to SAT-based attack

Error-Controllable Encryption



Error-Controllable Encryption

- **Theorem. 1** *The ECE scientific benchmarks will have different error rate ranging from* 2^{-n} *to* 2^{m-n} *for a wrong key.*
    - *Lower and upper bound of error rate happens when* $l = 0$ *and* $l = \mathrm{m}$, *respectively.*

Error-Controlable Encryption

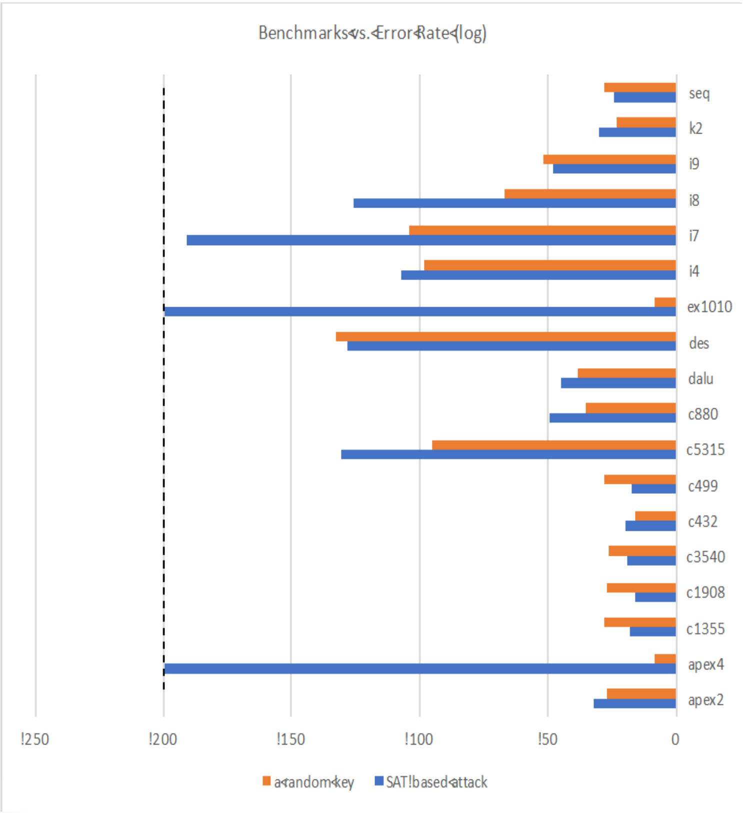
- **Theorem. 2** *The minimal number of iterations for the SAT-based attack is* 2^{n-m}.
  - *Only keys with* $k_i = x_i$ *for all* $i \in m \dots n-1$ *are possible to be pruned in each iteration.*
  - *For bits* $x_m \dots x_{n-1}$, *there exists* 2^{n-m} *combinations.*

Error-Controllable Encryption

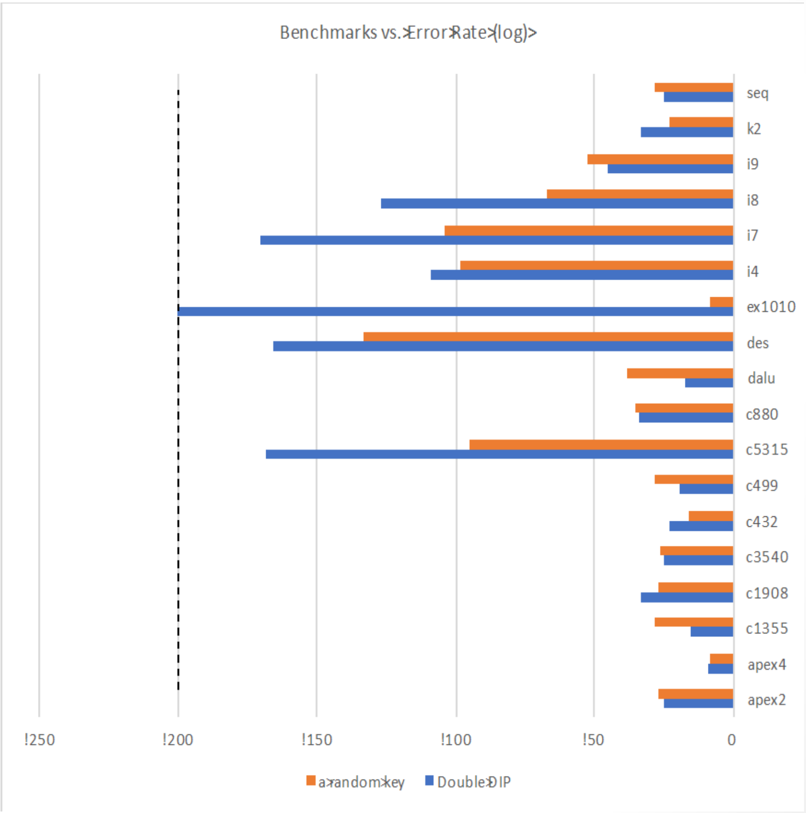
- ***Adjustable**: choose different m.*
- ***Trade off**: error rate and iteration numbers.*
- ***Randomness:** can be further obfuscated by randomly selecting the correct key, inserting inverters after key bits, etc.*
- ***Exponential** number of iterations for SAT-based attack to decrypt.*

Evaluation

- *Compare error rates of returned key and a random key on ECE*



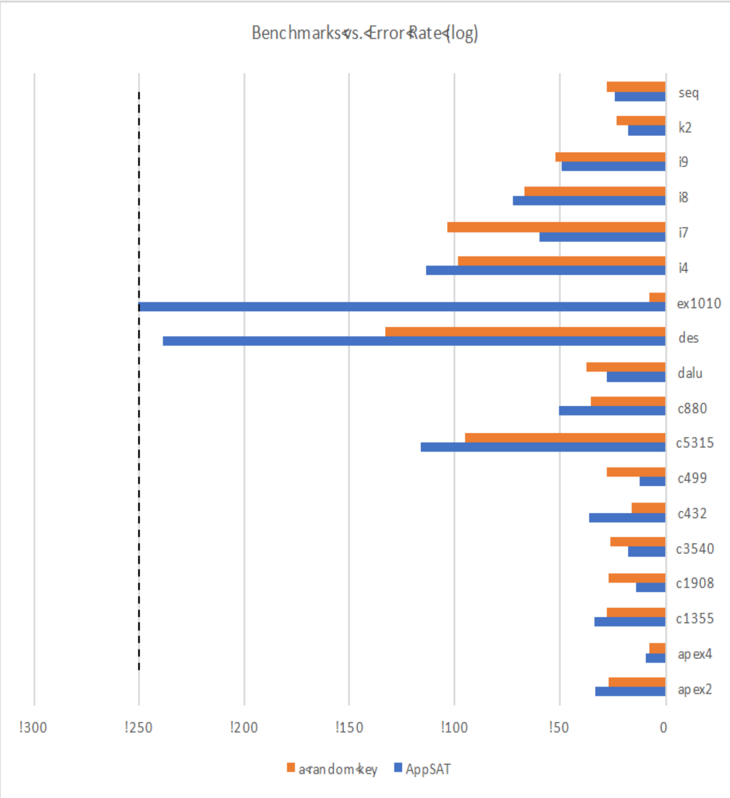
SAT-based Attack



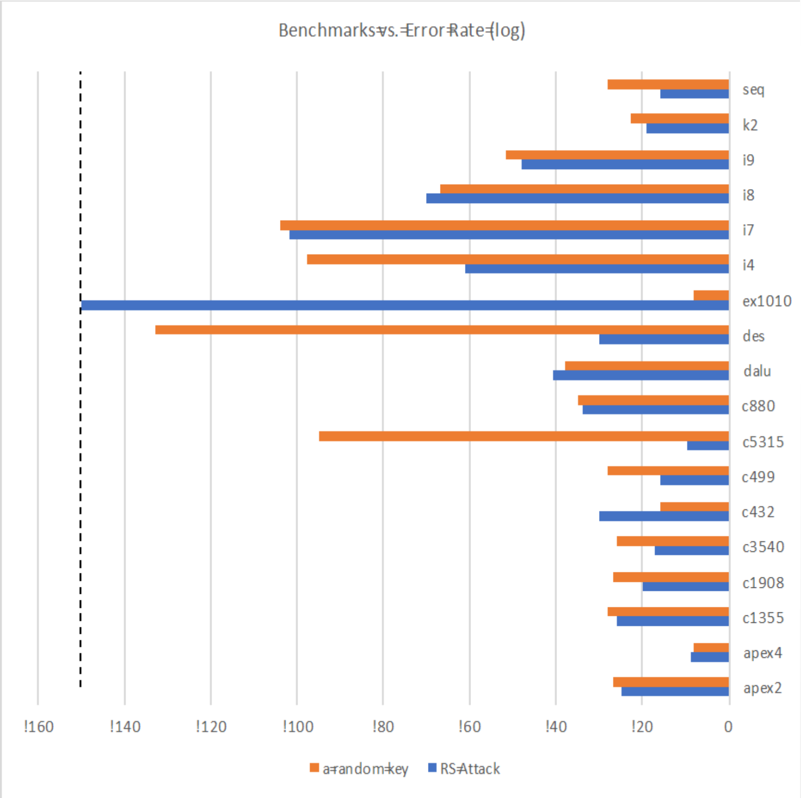
Double DIP

Evaluation

- *Compare error rates of returned key and a random key on ECE*



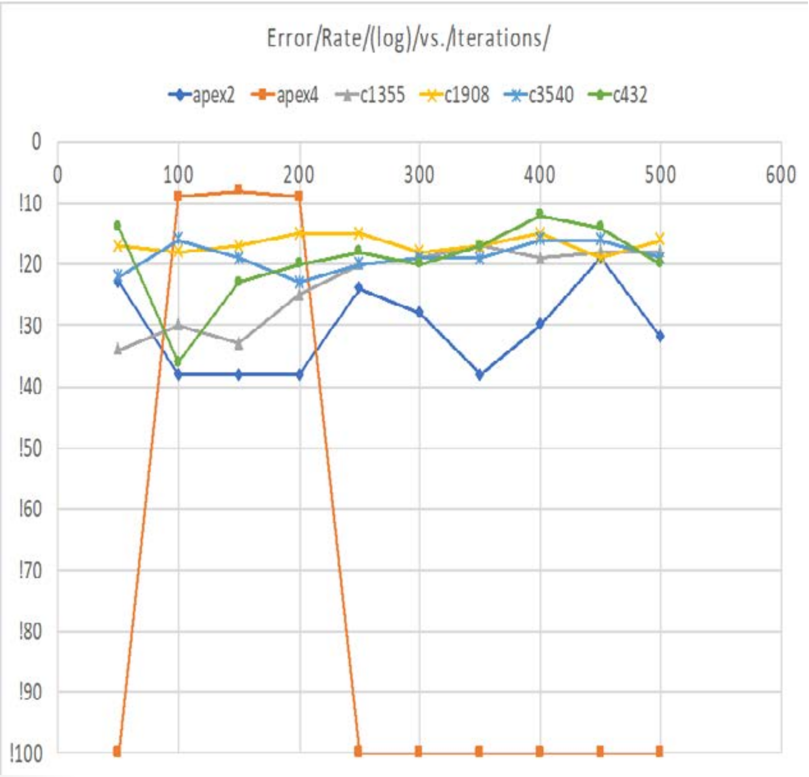
AppSAT



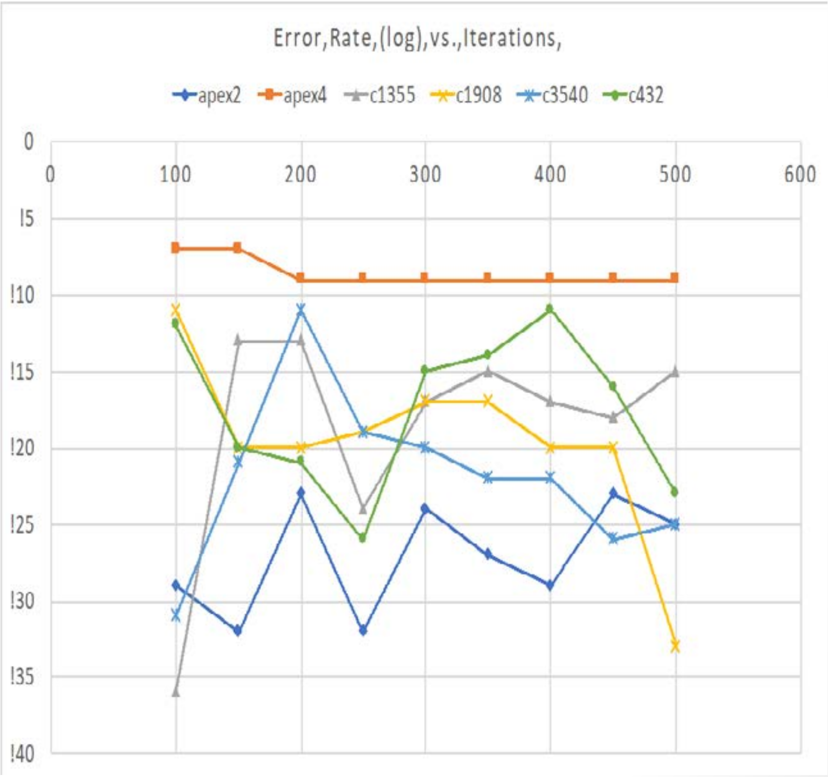
RS Attack

Evaluation

- *Error rates of returned key is at different iterations*



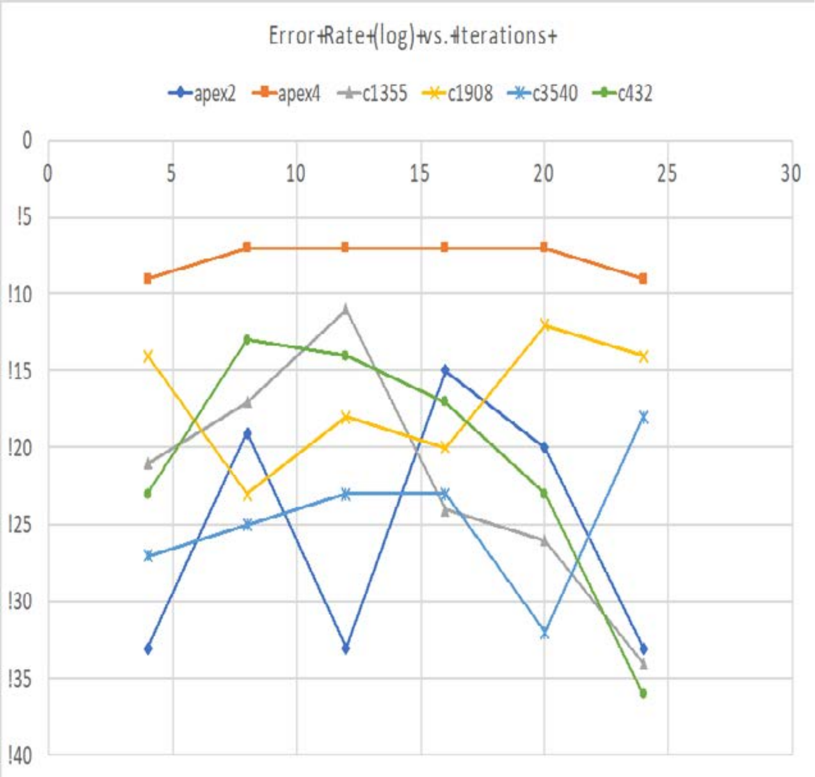
SAT-based Attack



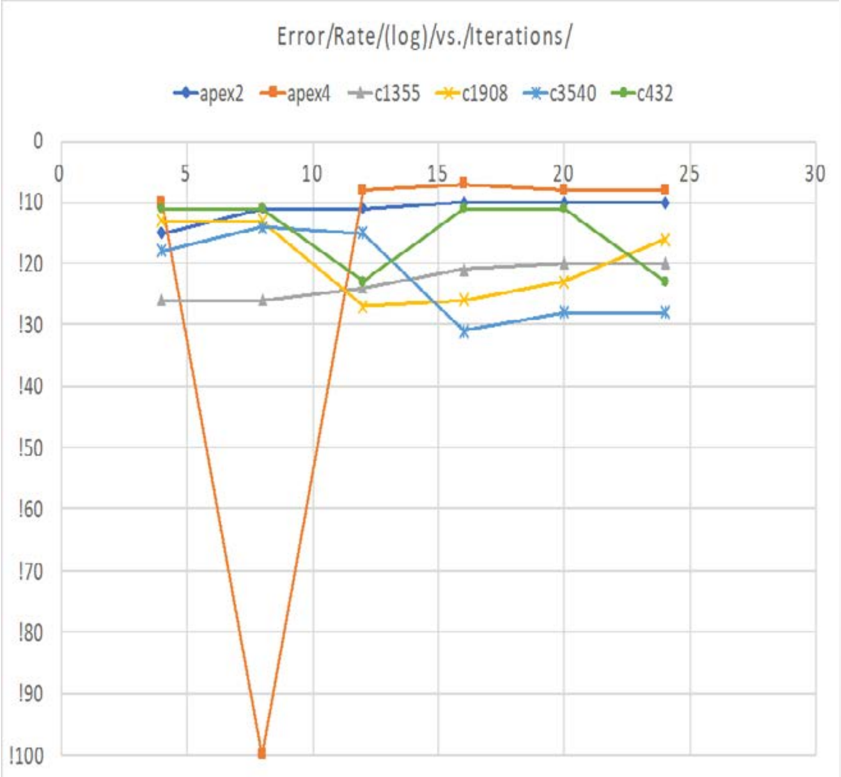
Double DIP

Evaluation

- *Error rates of returned key is at different iterations*



AppSAT



RS Attack

Conclusion

- Approx attacks are good at hybrid encryptions w/ big gaps of error rates
- They are not effective on homogenous encryptions
  - Not different from random key guessing on ECE benchmarks
  - Error rates not decreasing with more iterations
- More investigations are needed on approx attacks

Thank You!
Q&A