Power Analysis Attack on Trivium Stream Cipher

CO 421- Final Year Project I

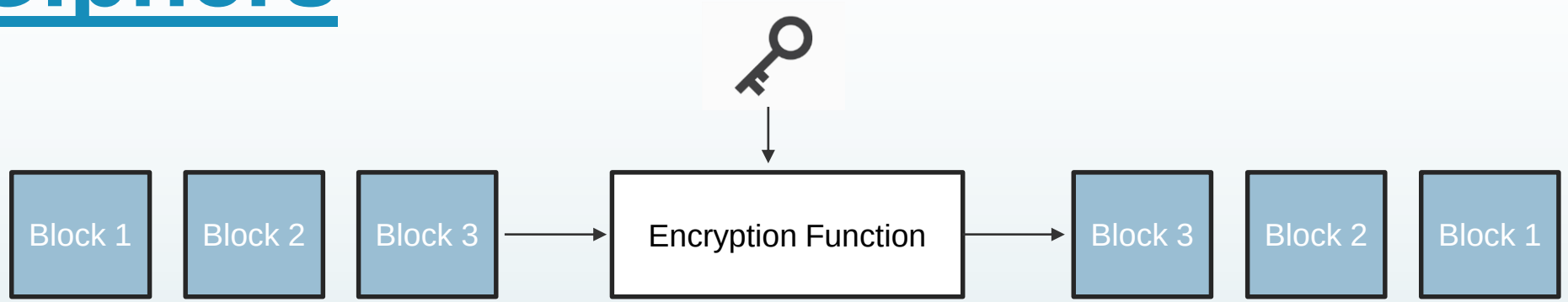
GROUP 18

De Silva M.D.R.A.M. (E/13/058)

Navaratna N.M.I. (E/13/237)

Kumarasiri G.M.D. (E/13/200)

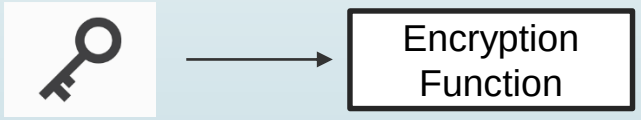
Ciphers



Plaintext

Block Cipher

Ciphertext



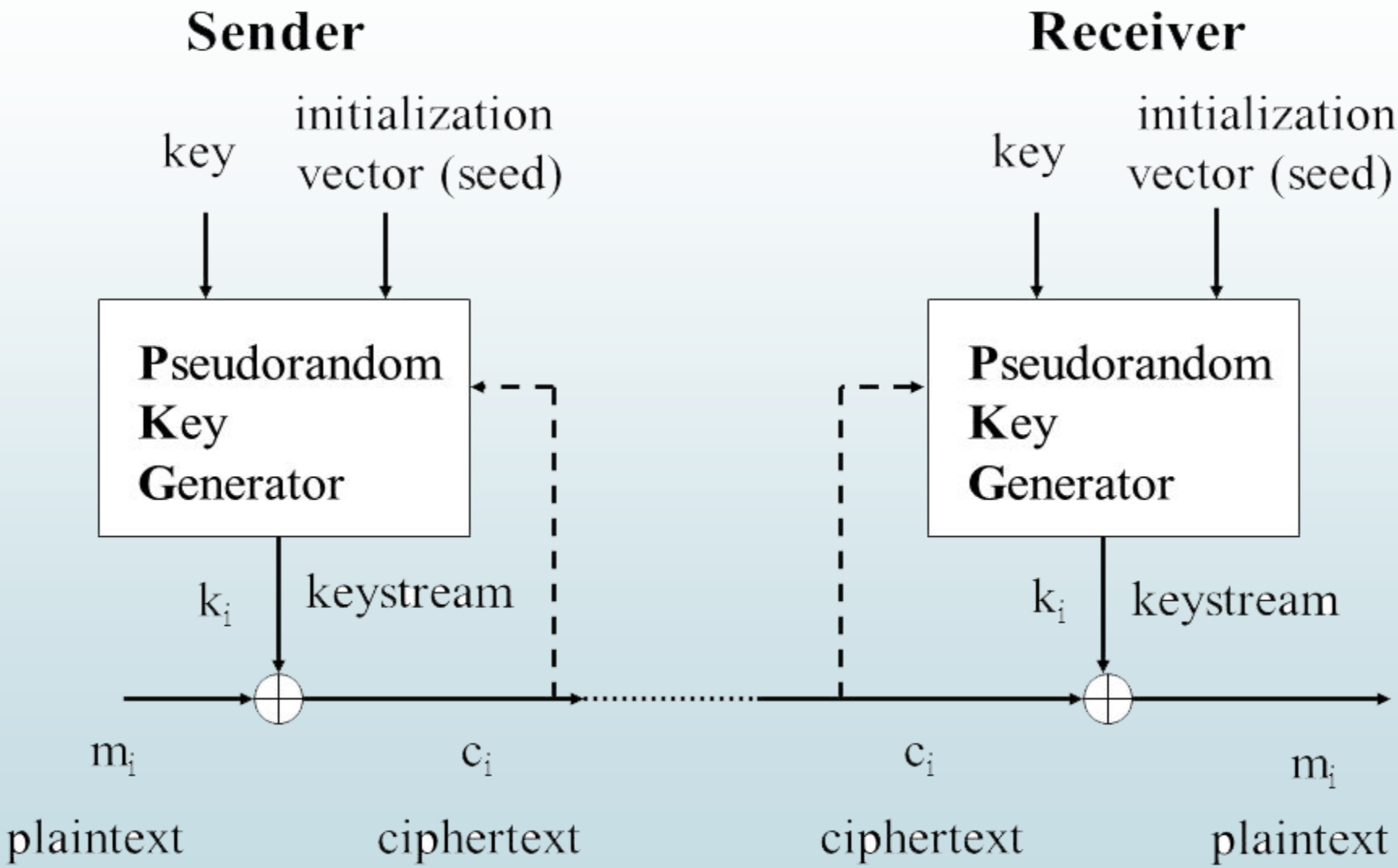
Bit stream 00101110011100011

Plaintext 110011100011

Ciphertext 100110011010

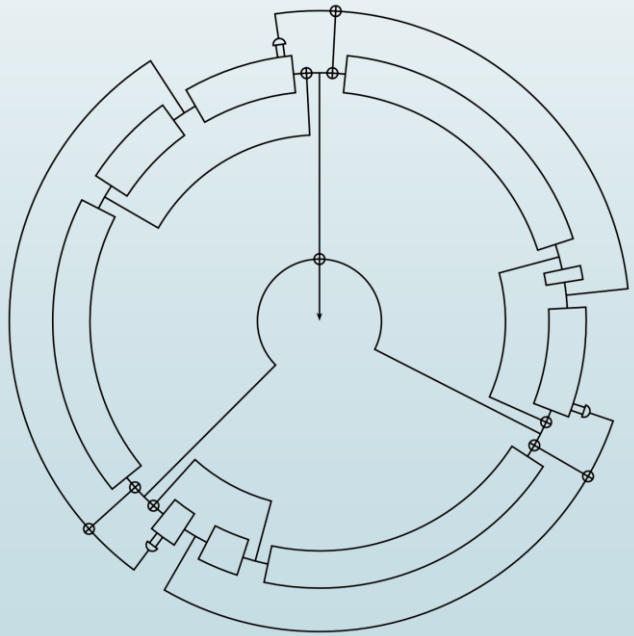
Stream Cipher

Typical Stream Cipher



Trivium Stream Cipher

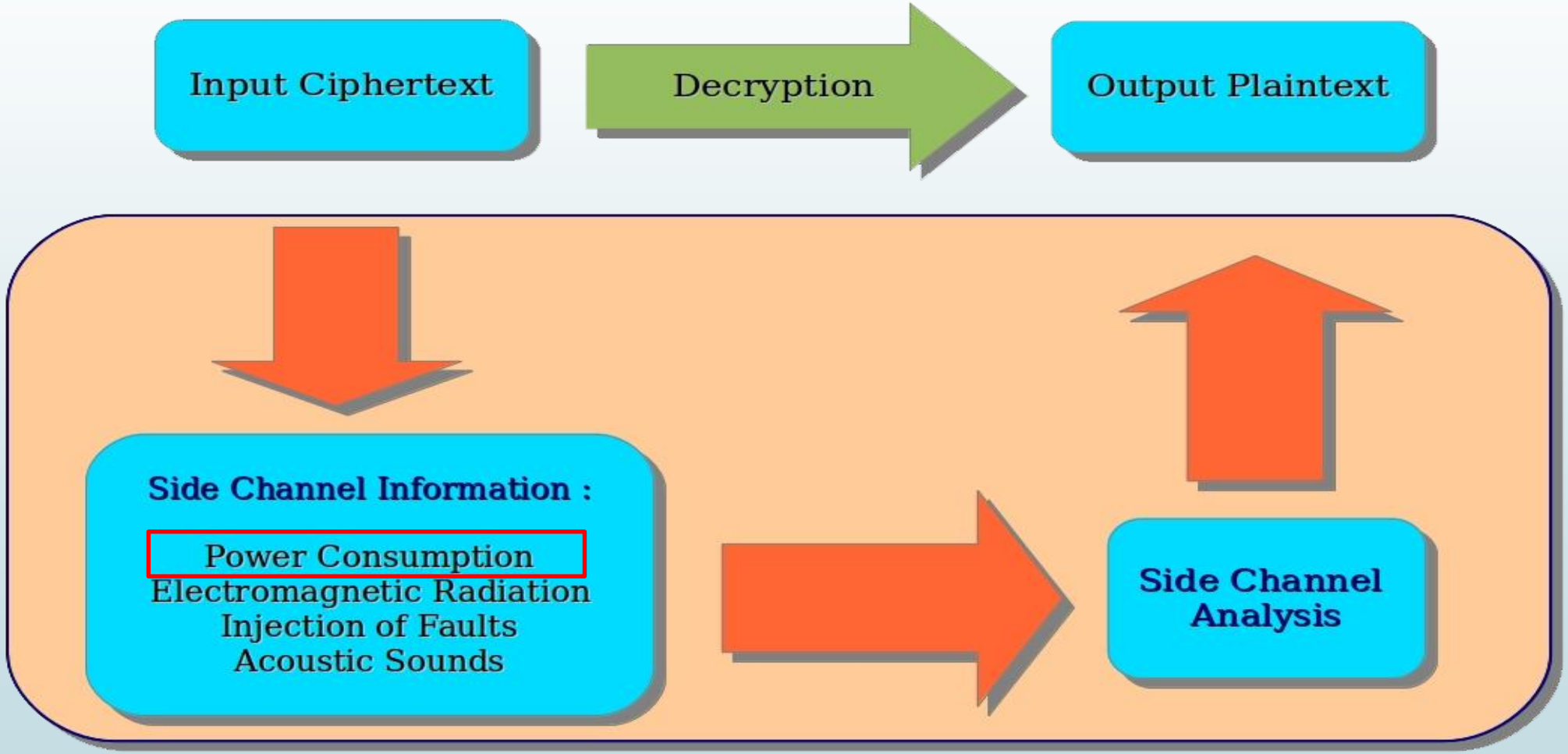
- Christophe De Canniere and Bart Preneel
- Hardware oriented synchronous stream cipher
- Key length – 80 bits
- IV length – 80 bits
- State – 288 bits
- Stream – 2^{64} bits



Trivium Security

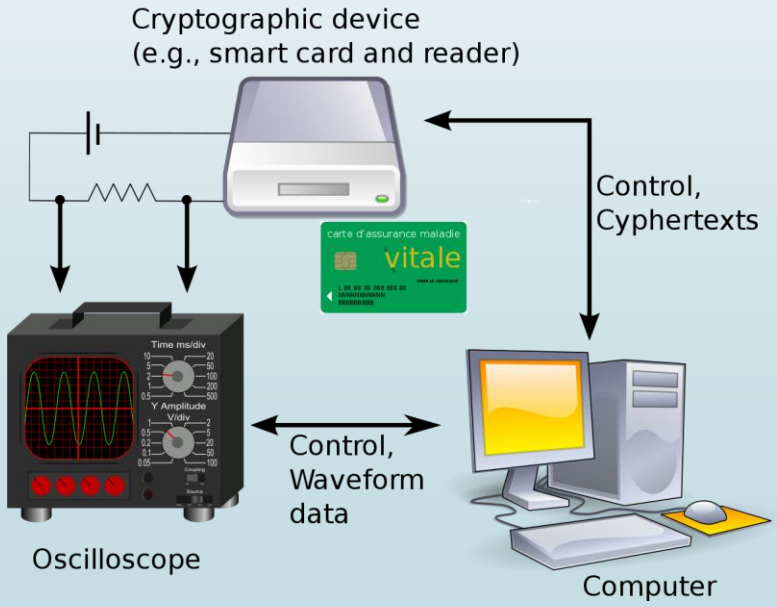
- Evolves non-linearly
- Trivium is secure from both
  - Correlation between linear combination of keystream bits and internal state
  - Correlation between keystream bits

Side Channel Attacks



Power Analysis Attacks

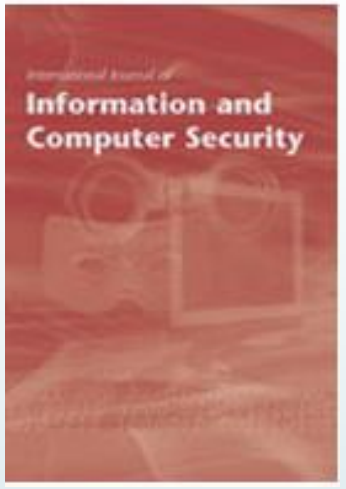
- Observe and study power consumption of the cryptographic system
- Collect power traces and do statistical analysis to get the secret key



Related Work

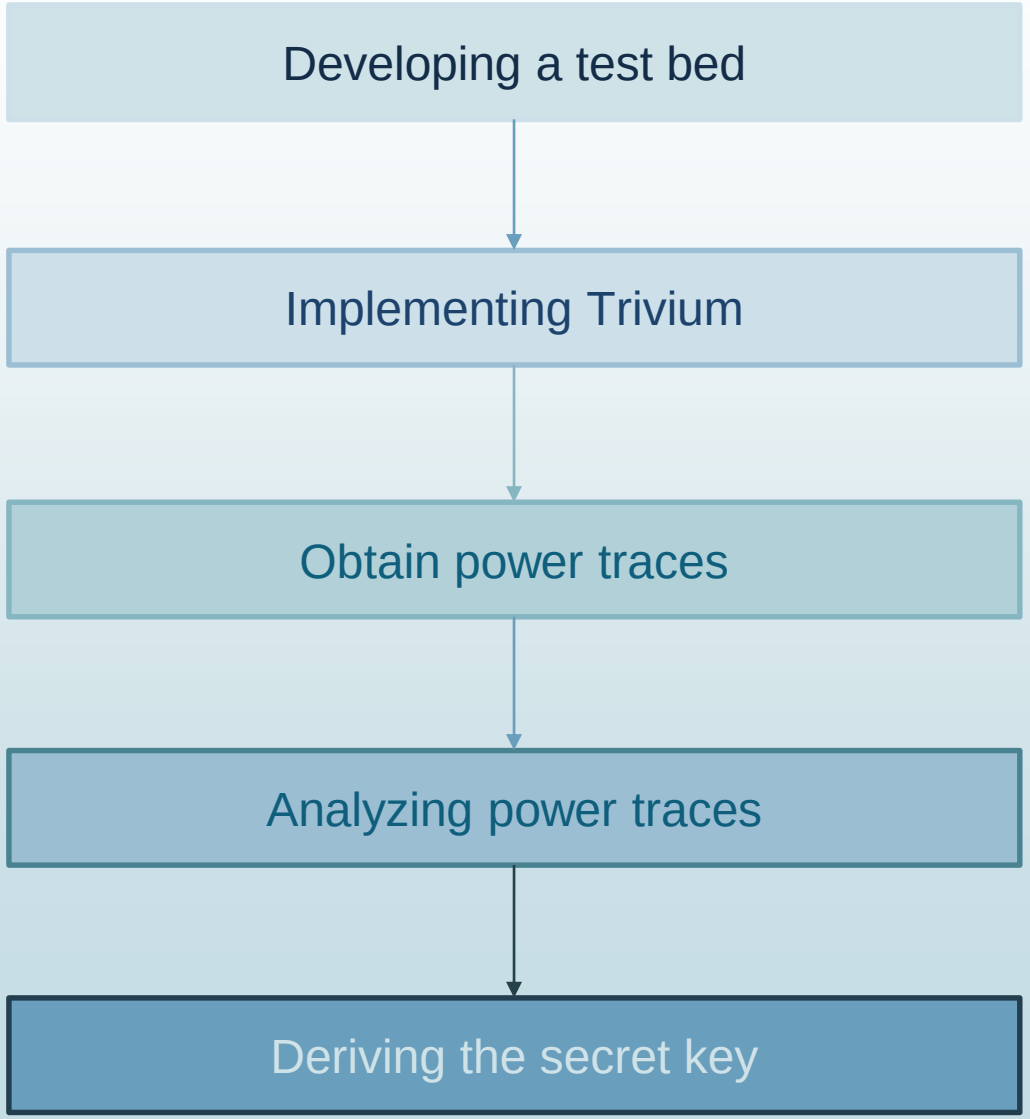
Work	Introduced by	Issues
Correlation power analysis of trivium	Yanyan Jia, Yupu Hu, Fenghe Wang, Hongxian Wang	Implemented using a hardware module and not tested using a software module.
Differential power analysis of stream ciphers	Fischer W, Gammel BM, Kuiffer O	High computational complexity than correlation power analysis method
Power analysis based side channel attack	Hasindu Gamaarachchi, Harsha Ganegoda	Focused on block cipher

Review Paper

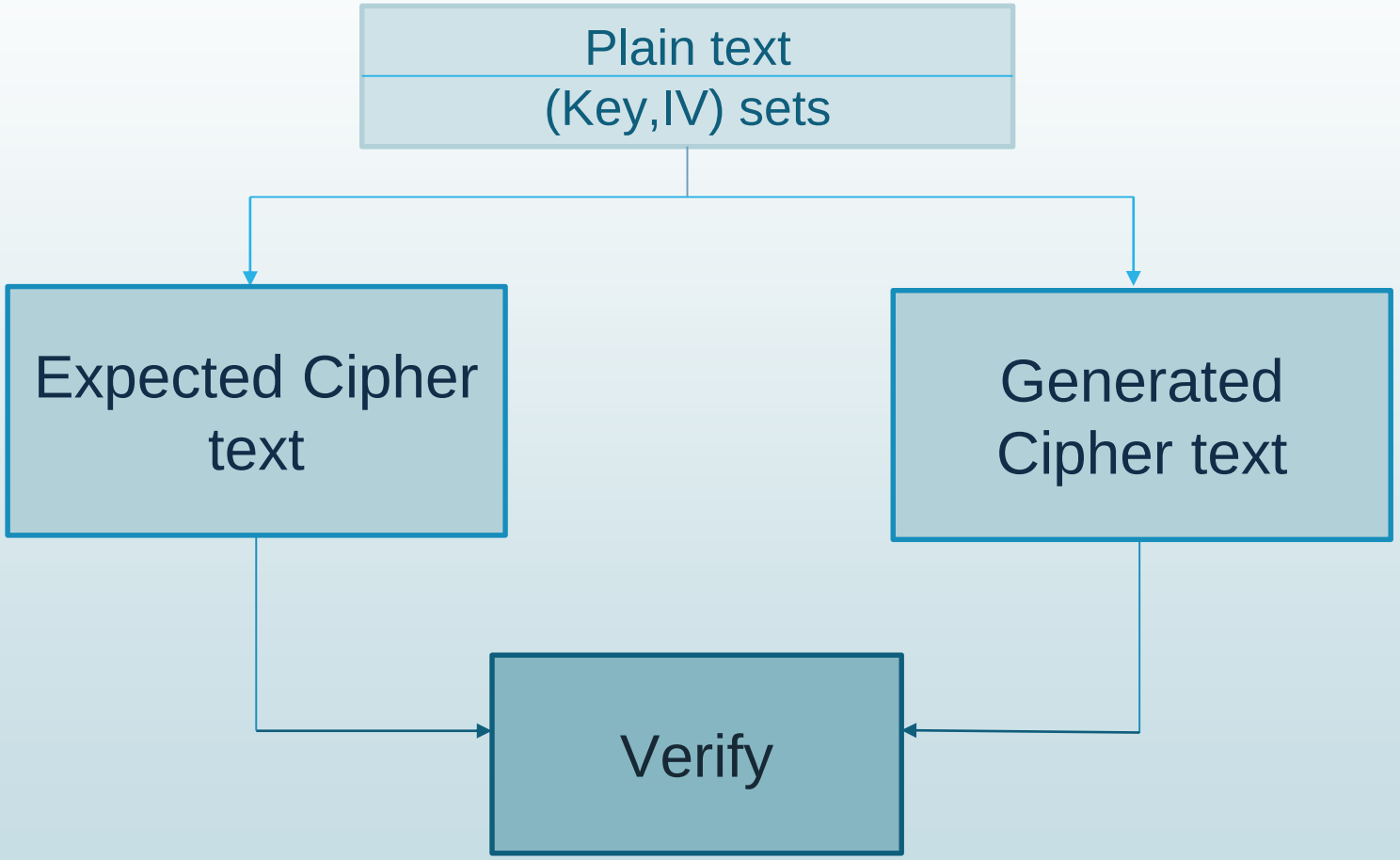


- Scope expanded – “On Power Analysis Against Hardware Stream Ciphers”
- Review paper submitted to the **International Journal of Information and Computer Security** (**IJICS**)
- **IJICS** is indexed in Scopus(Elsevier), Compendex [formerly EI] (Elsevier), ACM Digital Library etc.
- Currently the revised version is submitted after initial review.

Overall Design



Verification of Trivium implementation on PIC



Milestones

Milestone	Allocated Time Period	Actual Time Spent	Completion
Litterateur review and differentiating the project tasks from the previous attempts	3 Weeks	5 weeks	100%
Implementing Trivium stream cipher on a microcontroller and testing it's functionality	4 weeks	4 weeks	100%
Getting power traces and analyzing them to derive the secret key	4 weeks	2 weeks	40%

Plan for this semester

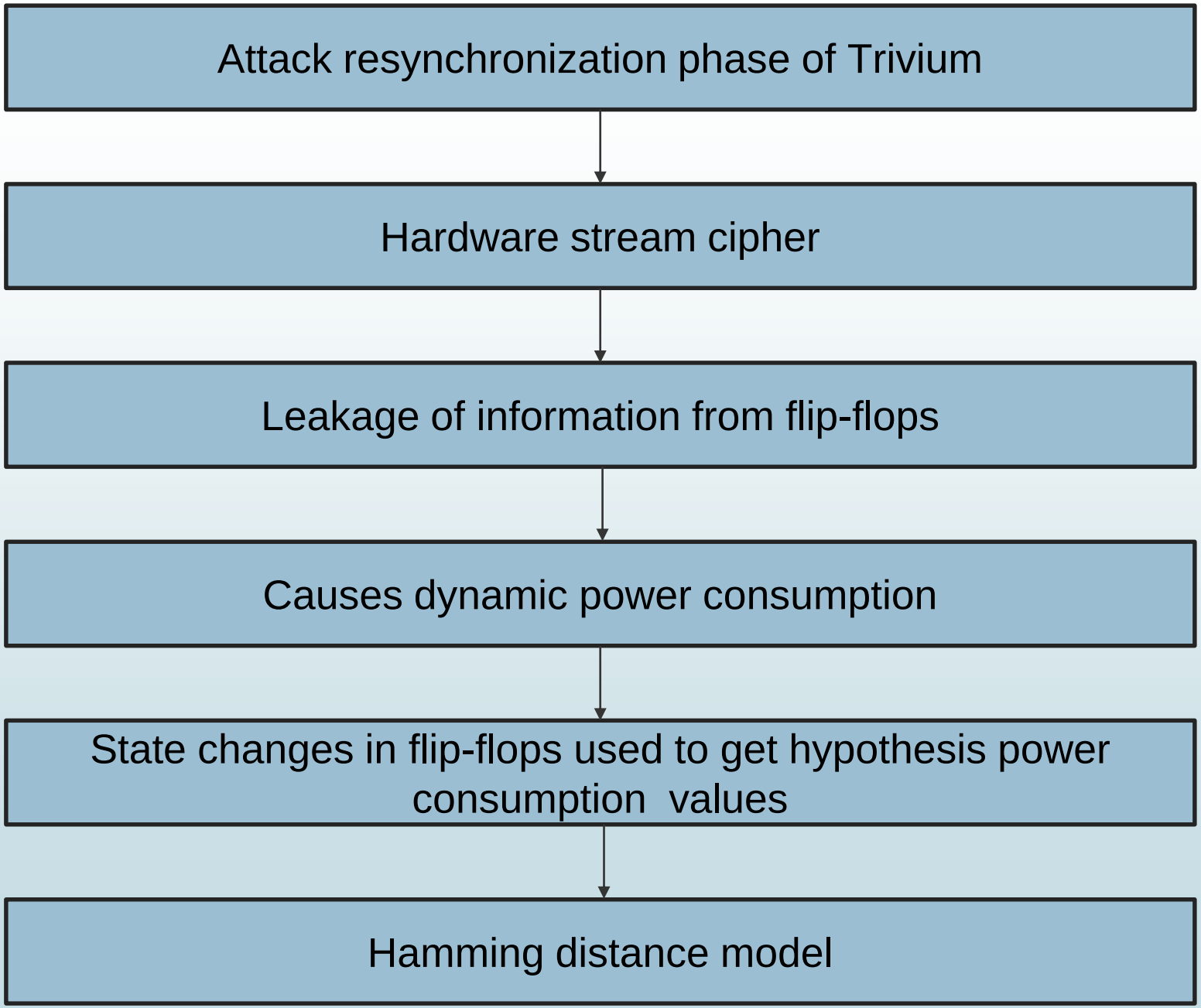
Milestone	Allocated Time Period
Getting power traces and analyzing them to derive the secret key	9 weeks
Developing countermeasures	5 weeks

WBS for CPA

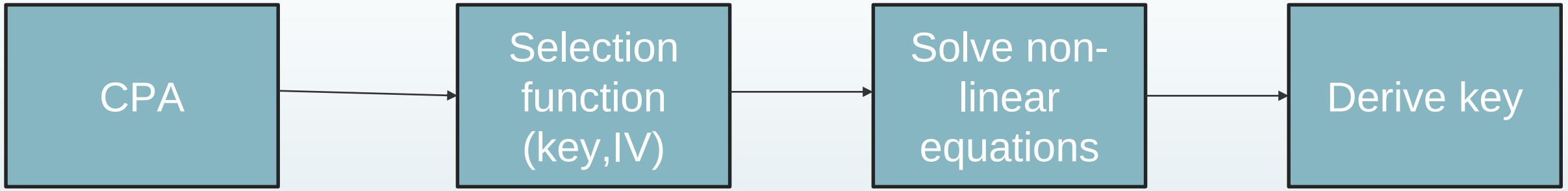
Task	Allocated Time Period
Setup the oscilloscope to attack trivium	1 week
Automate oscilloscope using Matlab	2 week
Obtain power traces	1 week
Implement CPA method	3 week
Analyze power traces and derive secret key	2 week

Design Specifications of Trivium

$$(a_1, \ldots, a_{93}) = (0, \ldots, 0, k_{80}, \ldots k_1)$$
$$(b_1, \ldots, b_{84}) = (0,0,0,0, IV_{80}, \ldots IV_1)$$
$$(c_1, \ldots, c_{111}) = (1,1,1,0, \ldots, 0)$$
$$a_{i+93} = a_{i+24} \oplus c_i \oplus c_{i+1}c_{i+2} \oplus c_{i+45}$$
$$b_{i+84} = b_{i+6} \oplus a_i \oplus a_{i+1}a_{i+2} \oplus a_{i+27}$$
$$c_{i+111} = c_{i+24} \oplus b_i \oplus b_{i+1}b_{i+2} \oplus b_{i+15}$$
$$z_i = a_i \oplus b_i \oplus c_i \oplus a_{i+27} \oplus b_{i+15} \oplus c_{i+45}$$

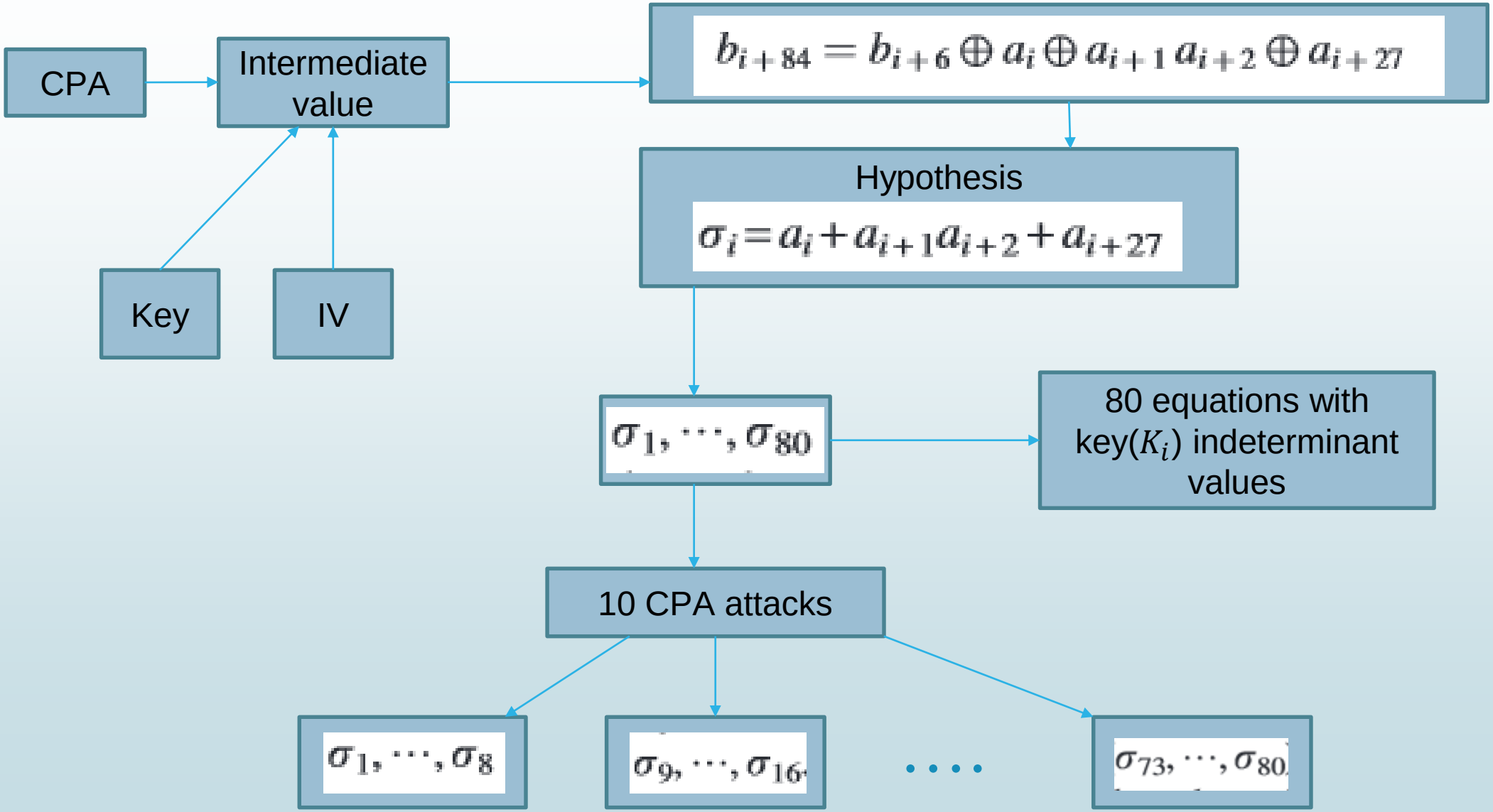

Attack resynchronization phase of Trivium
Hardware stream cipher
Leakage of information from flip-flops
Causes dynamic power consumption
State changes in flip-flops used to get hypothesis power consumption values
Hamming distance model

CPA on Trivium

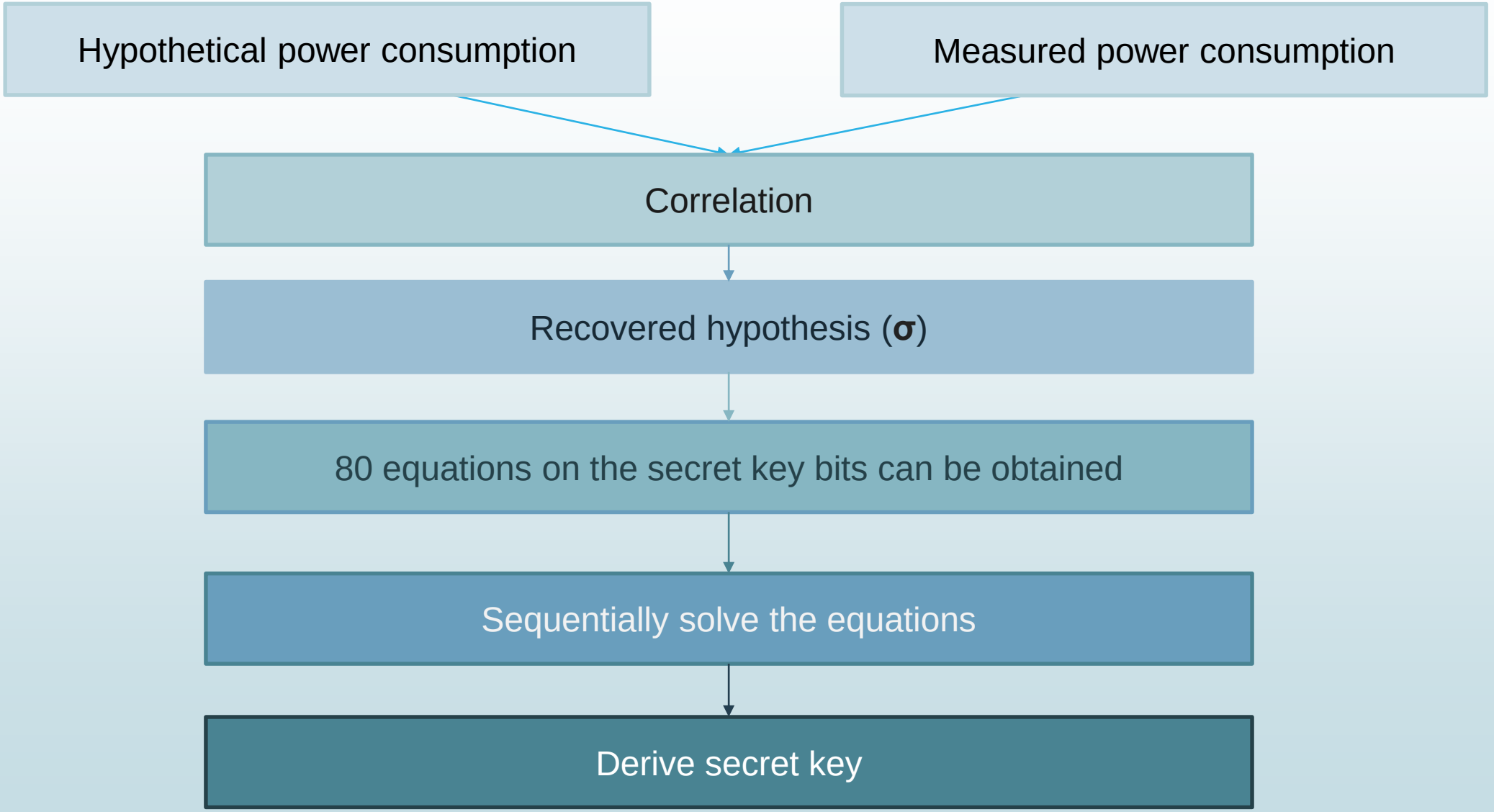


CPA for each 8 rounds of initialization process

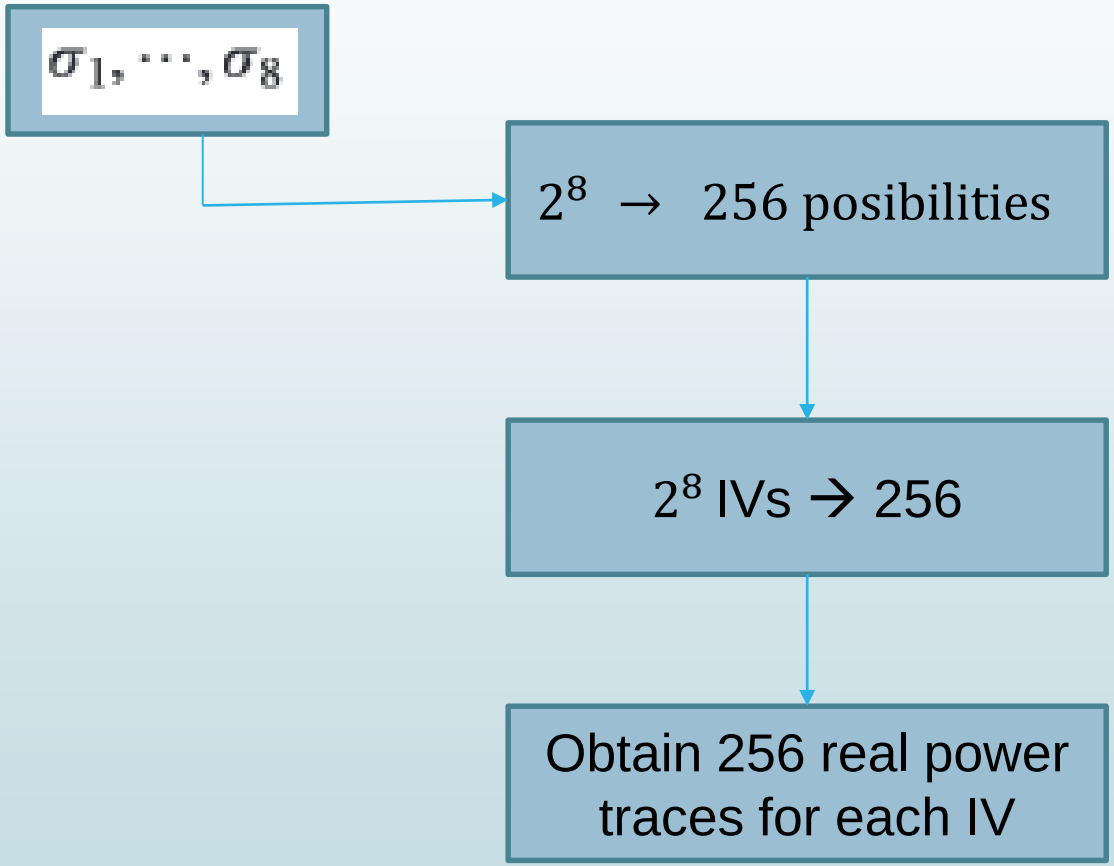
CPA on Trivium



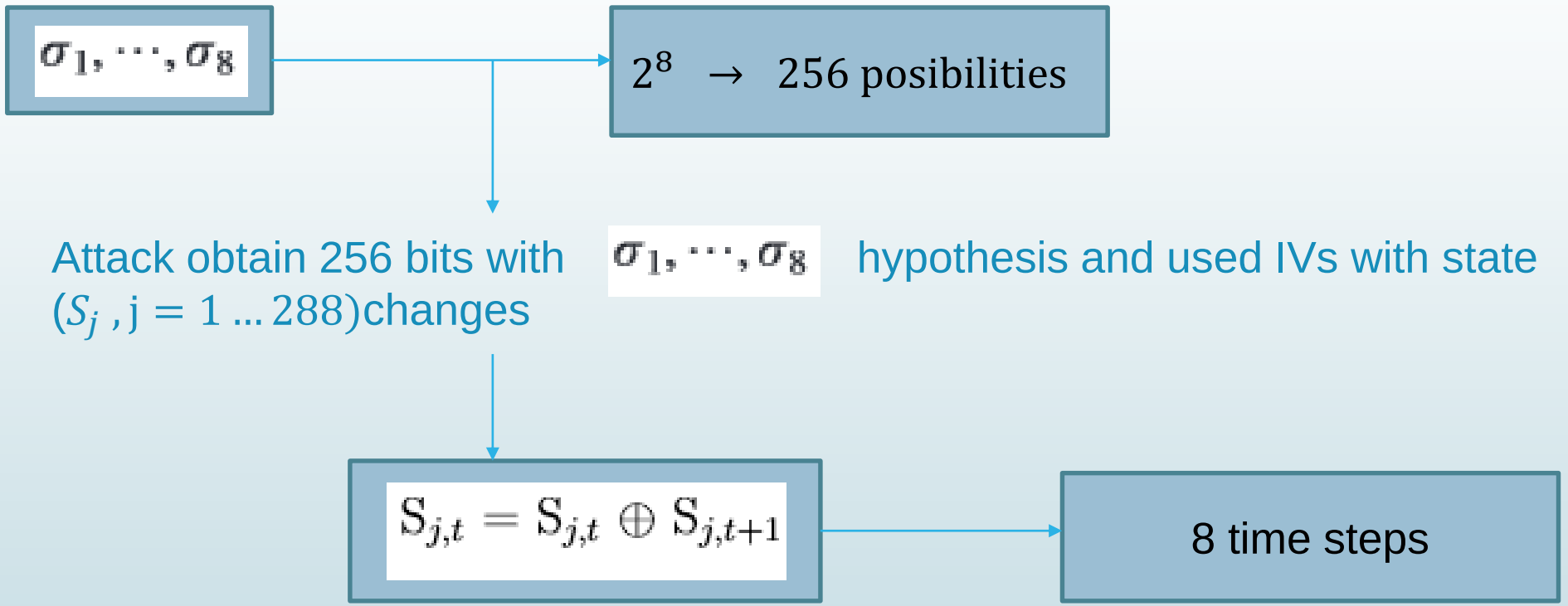
CPA on Trivium



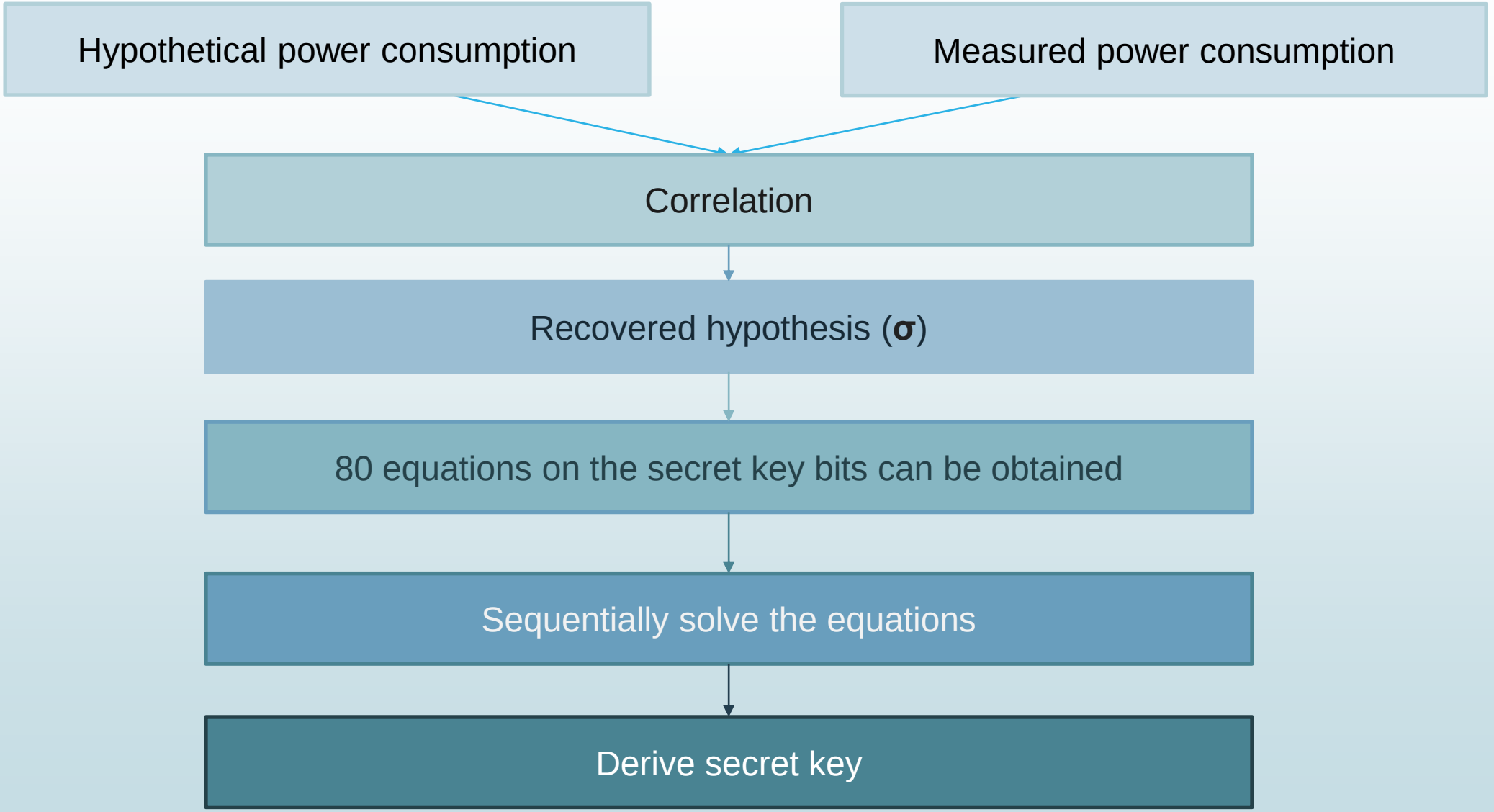
1 CPA attack : real power traces



Hypothetical power traces



CPA on Trivium



Thank You

WBS for CPA

$$\mathrm{b}_{i+84} = \mathrm{b}_{i+6} \oplus \mathrm{a}_i \oplus \mathrm{a}_{i+1}a_{i+2} \oplus \mathrm{a}_{i+27}$$

$$\sigma_i = \mathrm{a}_i \oplus \mathrm{a}_{i+1}a_{i+2} \oplus \mathrm{a}_{i+27}$$

$$\mathrm{S}_{j,t} = \mathrm{S}_{j,t} \oplus \mathrm{S}_{j,t+1}$$

- Stream cipher and block ciphers. stream used in https wep dvd encryption(css encrp) gsm encryption Bluetooth

- Encryption has long been used by militaries and governments to facilitate secret communication. Encryption is now used in protecting information within many kinds of civilian systems, such as computers, networks (such as for Internet e-commerce), mobile telephones, wireless microphones, wireless intercom systems, Bluetooth devices, and bank automatic teller machines. Encryption is also used in digital rights management to restrict the use of copyrighted material and in software copy protection to protect against reverse engineering and software piracy.

- Standards and cryptographic software and hardware to perform encryption are widely available, but successfully using encryption to ensure security is a challenging problem. A single slip-up in system design or execution can allow successful attacks. Sometimes an adversary can obtain unencrypted information without directly undoing the encryption.

Trivium

- IV and key used to initialize the state
- Iterate state
  - extract values of 15 specific state bits and use them to update 3 bits of the state and to compute 1 bit of the key stream zi.
  - state bits then rotated and process repeats

Trivium Key Stream Generation

for i = 1 to N do

$t1 \leftarrow s66 \oplus s93$

$t2 \leftarrow s162 \oplus s177$

$t3 \leftarrow s243 \oplus s288$

$zi \leftarrow t1 \oplus t2 \oplus t3$

$t1 \leftarrow t1 \oplus s91 \wedge s92 \oplus s171$

$t2 \leftarrow t2 \oplus s175 \wedge s176 \oplus s264$

$t3 \leftarrow t3 \oplus s286 \wedge s287 \oplus s69$

$(s1; s2; : : : ; s93) \leftarrow (t3; s1; : : : ; s92)$

$(s94; s95; : : : ; s177) \leftarrow (t1; s94; : : : ; s176)$

$(s178; s279; : : : ; s288) \leftarrow (t2; s178; : : : ; s287)$

Trivium Initialization

- load 80-bit key and 80-bit IV into 288-bit initial state
- set all remaining bits to 0, except for s286, s287, and s288, which are set to 1
- state is rotated over 4 full cycles of the for look, but no bits are output (for i = 1 to 4∗288)

Trivium

- state bit is not used for at least 64 iterations after it has been modified
- up to 64 iterations can be computed at once, provided that 3 AND gates and 11 XOR gates in the original scheme are duplicated a corresponding number of times

Estimated Gate Counts

1-bit to 64-bit hardware implementations

Components	1-bit	8-bit	16-bit	32-bit	64-bit
Flip-ops	288	288	288	288	288
AND gates	3	24	48	96	192
XOR gates	11	88	176	352	704
NAND gate count	3488	3712	3968	4480	5504

Software

- Stream generation: 12 cycles/byte
- Key setup: 55 cycles
- IV setup: 2050 cycles
- on Intel XeonTM CPU 1.5 GHz

Trivium Security

- Linear correlations between key stream bits and internal state bits are easy to find because zi is simply defined to be equal to s66⊕s93 ⊕ s162 ⊕ s177 ⊕ s243 ⊕ s288.
- But, as opposed to LFSR based ciphers, Trivium's state evolves in a nonlinear way
  - not clear how an attacker should combine these equations in order to efficiently recover the state
  - Estimate: follow linear trails through the cipher and approximate the outputs of all encountered AND gates by 0. However, the positions of the taps in Trivium have been chosen in such a way that any trail of this specific type is forced to approximate at least 72 AND gate outputs
  - If assume that the correlation of linear combination is completely explained by a specific trail considered, then it would have a correlation coefficient of 2^{-72}
- Detecting such a correlation would require at least 2^{144} bits of key stream
- Other more complicated types of linear trails with larger correlations might exist, estimate that no correlations will