



Establishing Pairwise Keys in Distributed Sensor Networks

D. Liu & P. Ning
CCS 2003



Motivation

- Sensors in a Sensor Network (SN) need to establish secure communication channel
- Public Key cryptography is unwieldy
- Need to establish a shared key between every pair
- Should deal with compromise of some nodes



Problems

- Storage
- Computation
- Communication overhead (discovery)
- Resiliency



First Cut

- Polynomial Based key predistribution
- Every sensor has an ID
- A Key distribution server selects a bivariate t -degree symmetric polynomial over a field F_q

$$f(x, y) = \sum_{i,j=0}^t a_{ij} x^i y^j$$

such that $f(x, y) = f(y, x)$



Key setup

- For each sensor, the server gives it a polynomial share based on its ID

$$Key(y) = \sum_{i,j=0}^t a_{ij} ID^i y^j$$

- A shared key is computed by replacing y with the ID of the peer



Analysis

- Each sensor needs to store a t -degree polynomial
 - $(t+1) \log q$ storage space
- Proof states that it is t -collusion resistant
- No Communication overhead
- Key-Share distribution is required



Limitation

- Secure against at most t compromises
- For a large network, adequate security requires a large value of t
- Storage requirement on sensors is prohibitive
- Single polynomial is not good enough



Polynomial pool

- Randomly generate a pool of polynomials and distribute these polynomials
- Setup : Polynomial distribution
- Direct Key establishment
 - Predistribution
 - Real-time discovery
- Path Establishment
 - Predistribution
 - Real-time discovery (Similar to routing)



Polynomial Distribution

- Server selects a Random subset F of polynomials for each sensor and assigns the corresponding share.
 - s' polynomials out of total s polynomials
- A direct key is established using challenge response.
 - A common key is found out
- A Path key is established using other neighbors in Key-domain
 - Neighbors take part only in key setup



Analysis

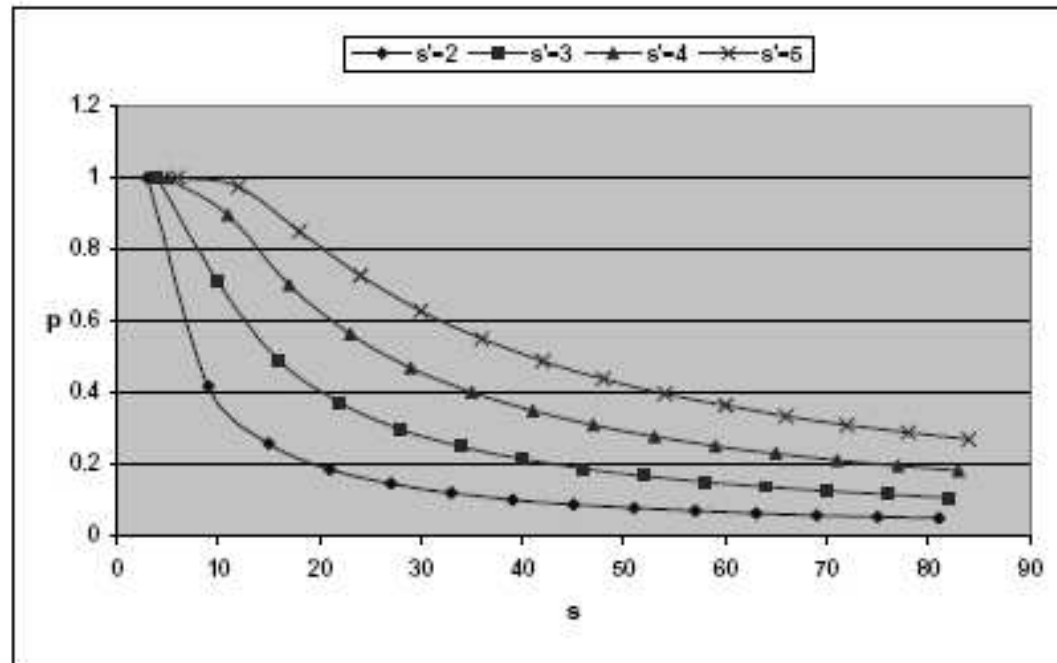
- Probability of two nodes establishing a direct key

$$p = 1 - \prod_{i=0}^{s'-1} \frac{s - s' - i}{s - i}$$

- Probability of two nodes establishing a direct key or a path key through d nodes

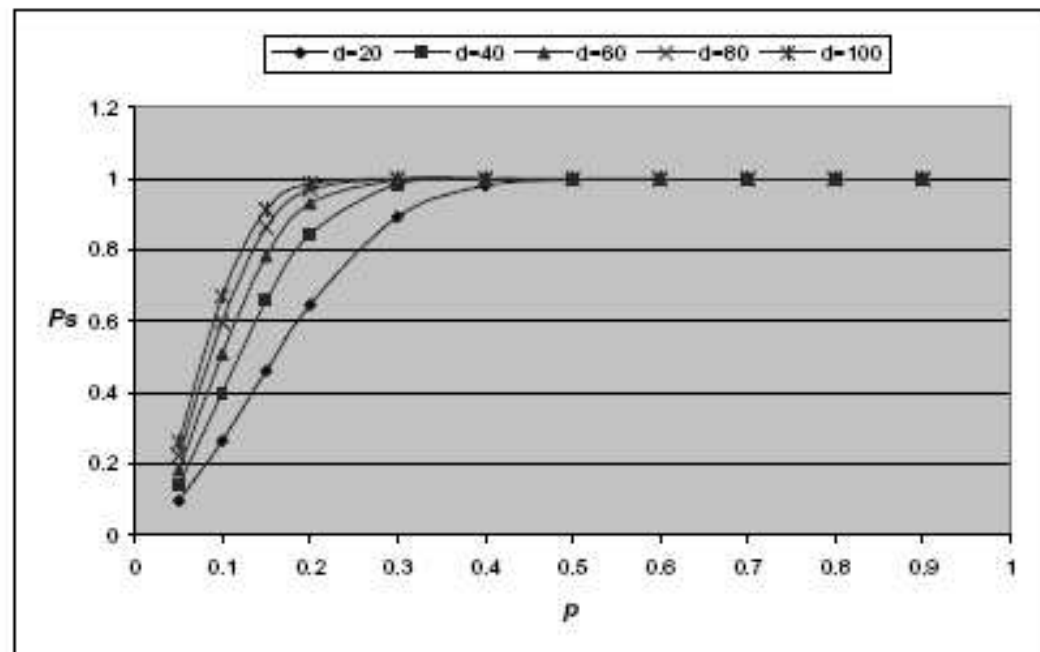
$$P_s = 1 - (1 - p)(1 - p^2)^d$$

Direct key establishment probability



(a) The probability p that two sensors share a polynomial v.s. the size s of the polynomial pool

Key Establishment probability



(b) The probability P_s of establishing a pairwise key v.s. the probability p that two sensors share a polynomial



Polynomial Compromise

- Compromised sensors N_c
- Probability of a polynomial used i times

$$P(i) = \frac{N_c!}{(N_c - i)!i!} \left(\frac{s'}{s}\right)^i \left(1 - \frac{s'}{s}\right)^{N_c - i}$$

- Probability of a polynomial compromise

$$P_c = 1 - \sum_{i=0}^t P(i)$$

- This is the fraction of compromised links



Security and Overhead

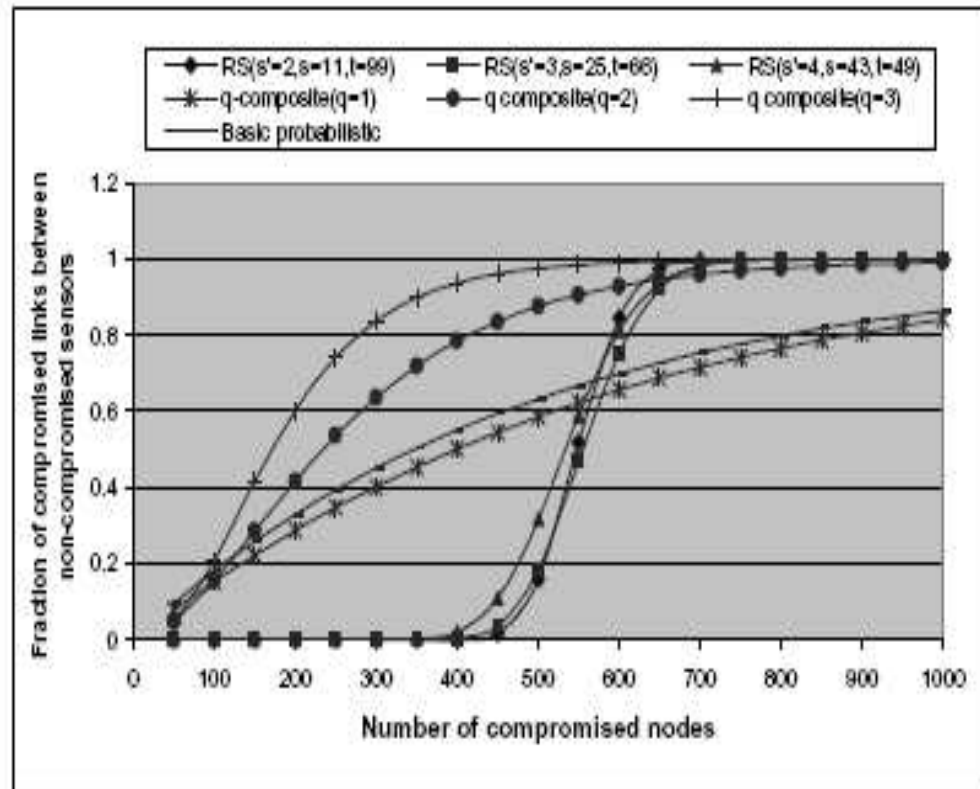
- Don't use a polynomial more than $t+1$ times
- Limitation on number of sensors $\frac{(t+1)*s}{s'}$
- Storage overhead $s'(t+1) \log q$
- Communication overhead: a list of s' IDs



Comparison with Other Schemes

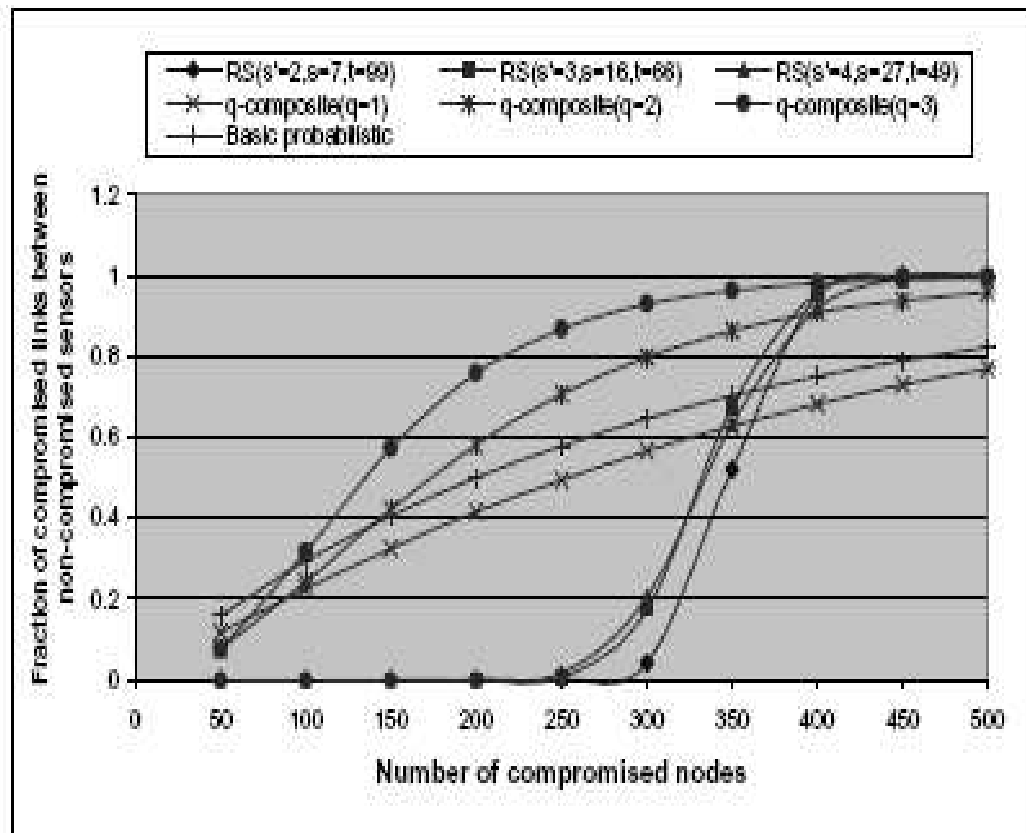
- Probabilistic
 - Pick random keys instead of shares
- Q-composite
 - Must share q common keys
- Random pairwise
 - Disallows the re-use of keys

Link Compromise



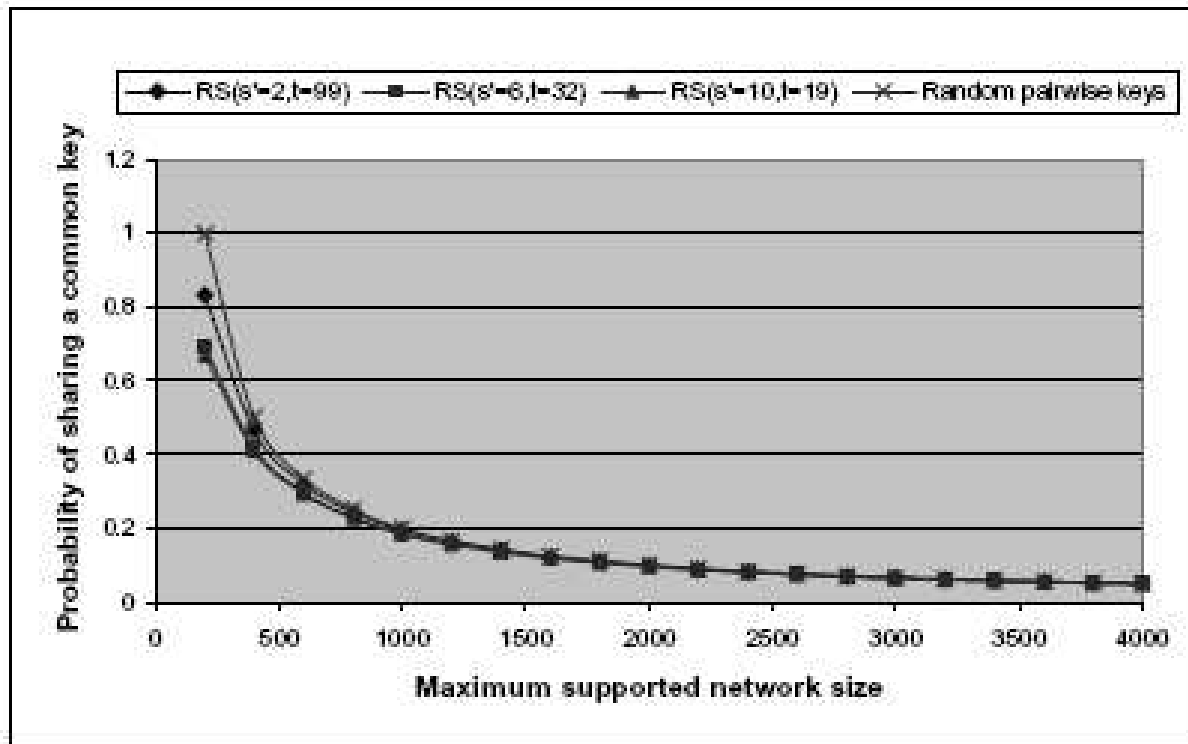
(a) $p=0.33$

Link Compromise

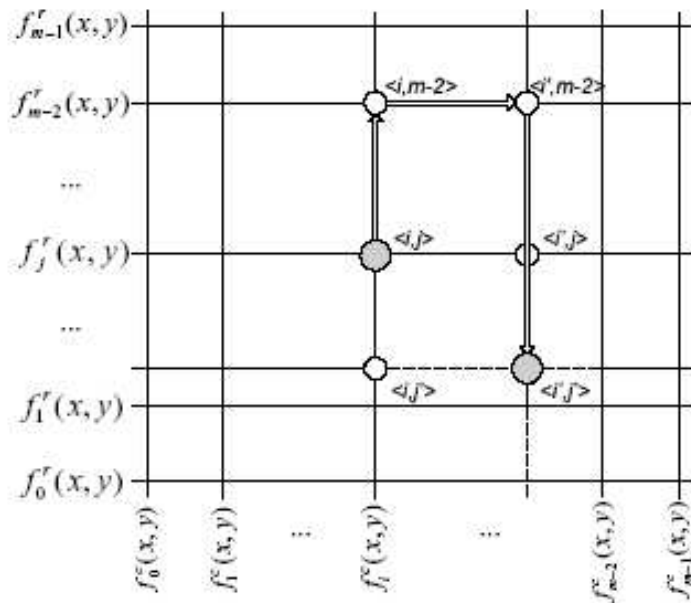


(b) $p=0.5$

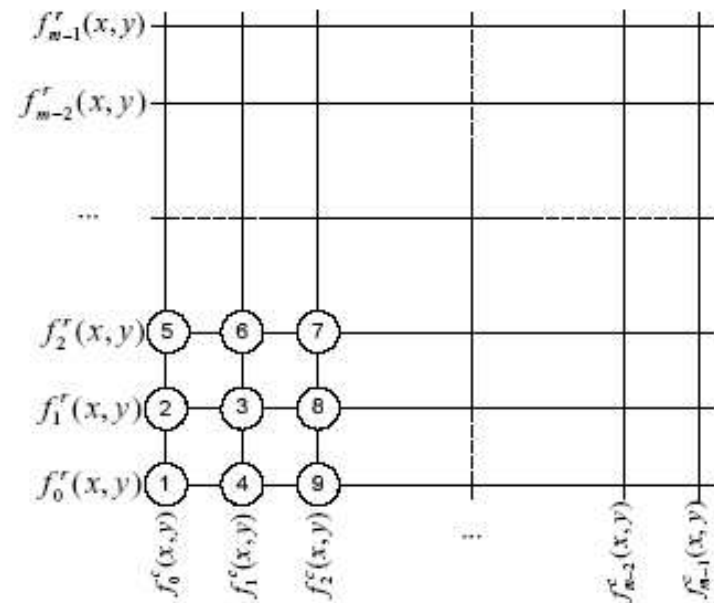
Random Pairwise key establishment



Grid Based Key predistribution



(a) The grid



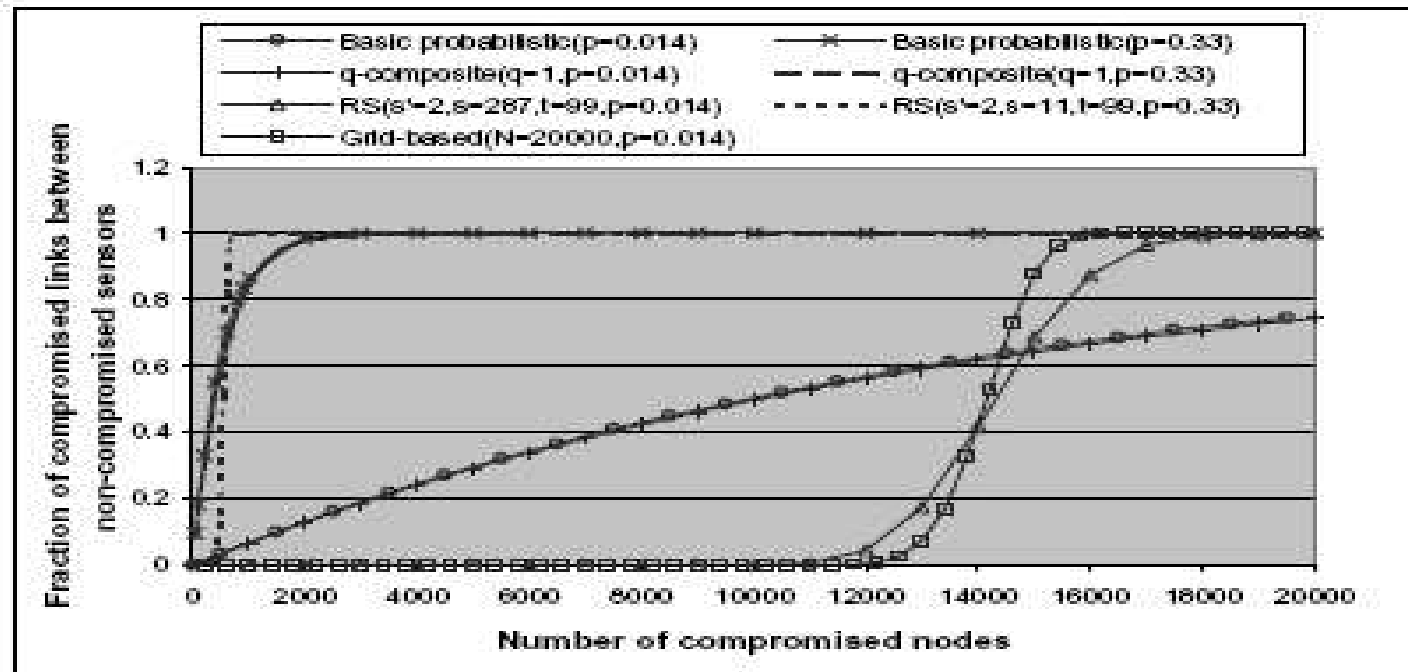
(b) An example order of node assignment



Key establishment

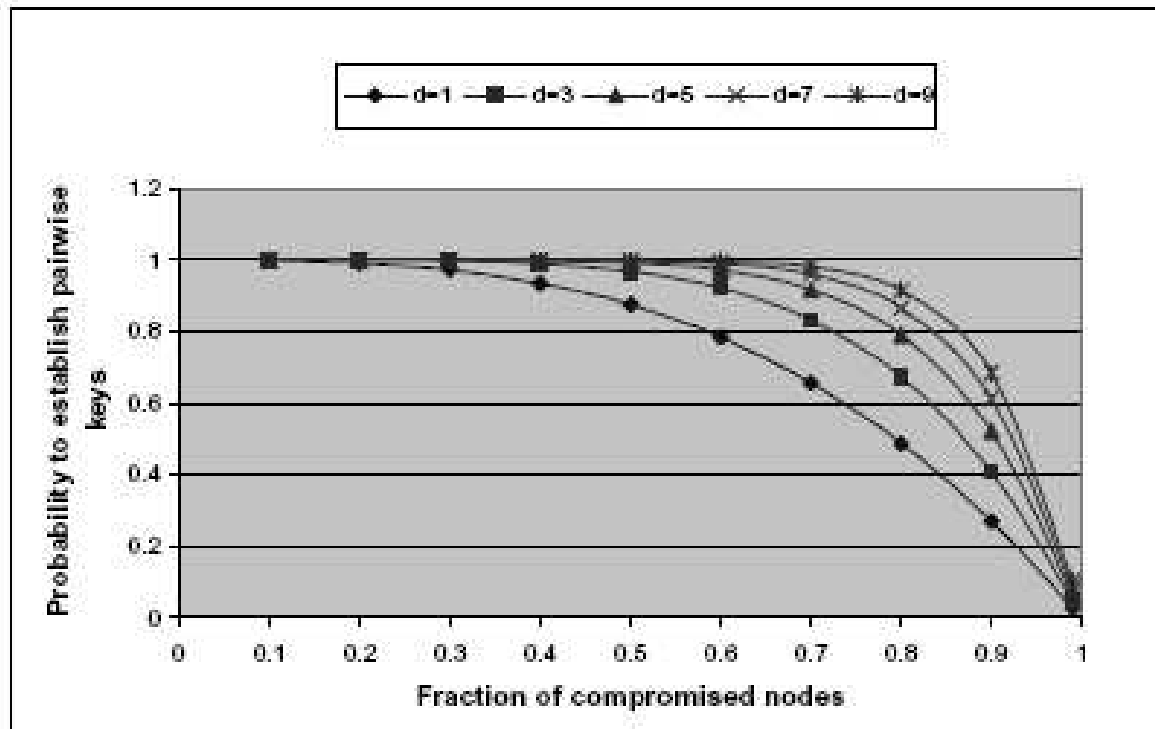
- Simple between neighbors in ID domain
- For remote nodes a *random key* is generated and that key is propagated through non-compromised neighbors
- Detecting compromise is not discussed
- Post compromise scenario is not discussed

Grid based scheme performance



(a) Fraction of compromised links between non-compromised sensors v.s. number of compromised sensor nodes. Assume each sensor has available storage equivalent to 200 keys.

Key establishment after compromise



(b) Probability to establish a pairwise key v.s. the fraction of compromised nodes



Opinion

- A simple enhancement over existing schemes
- Discussion about compromise is weak
- Distinction between ID domain neighbors and physical neighbors is blurred
- Not sure .. But the comparison does not seem to be entirely fair in terms of parameters
- The storage requirement would be high