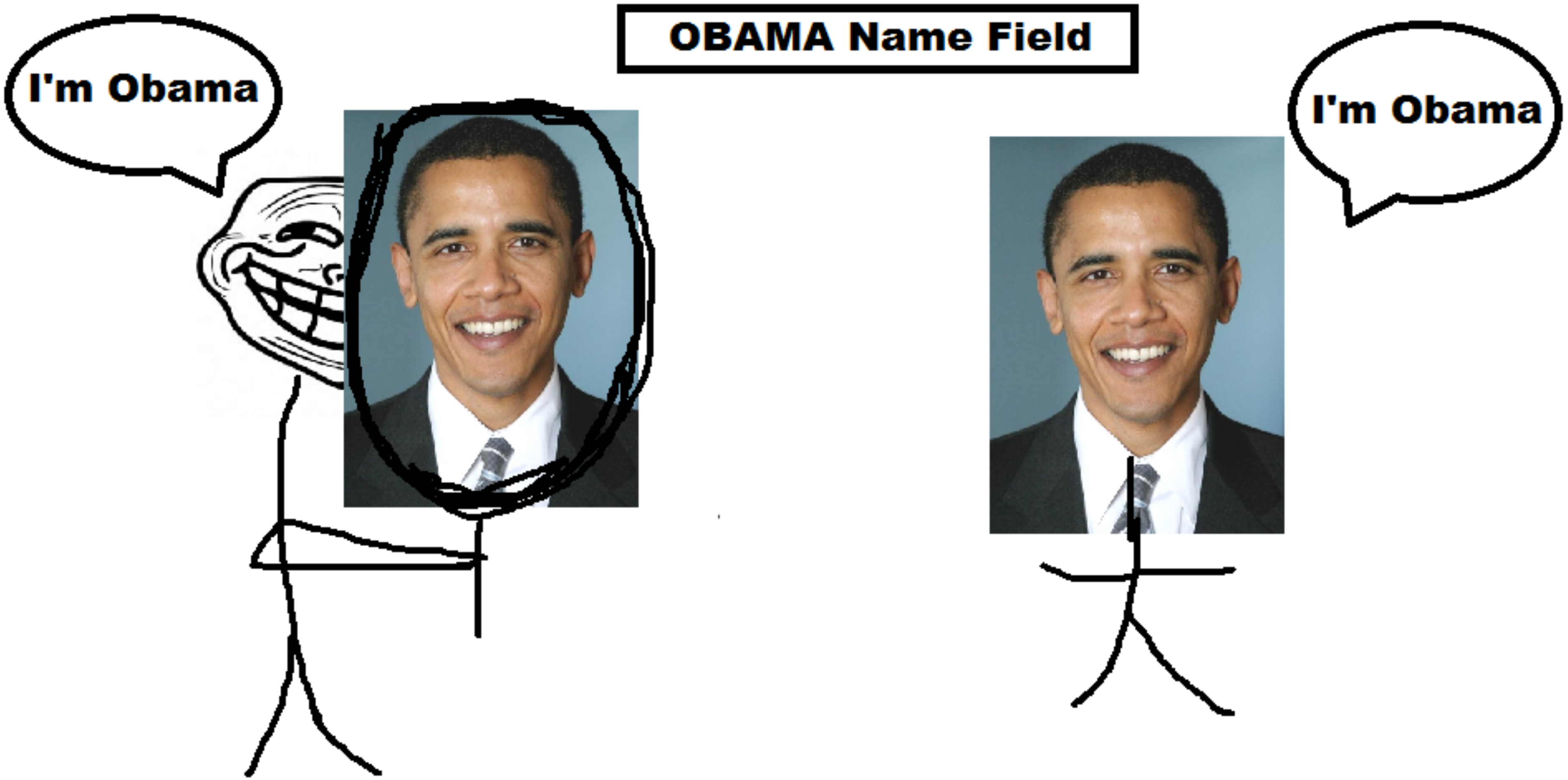
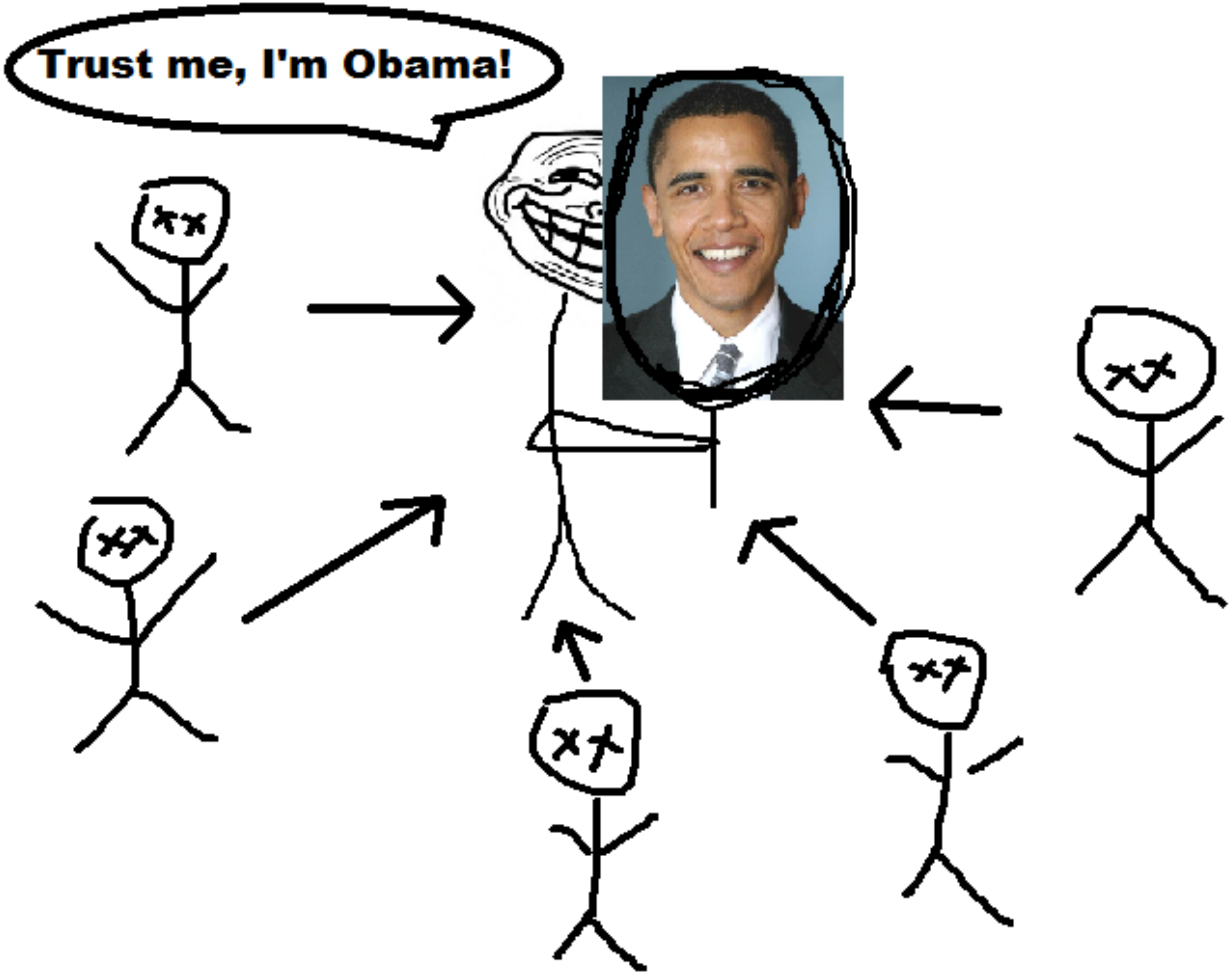
Part 2

PGP Problem 1: Impersonation



Problem 2: *Anyone* can sign *Any* key



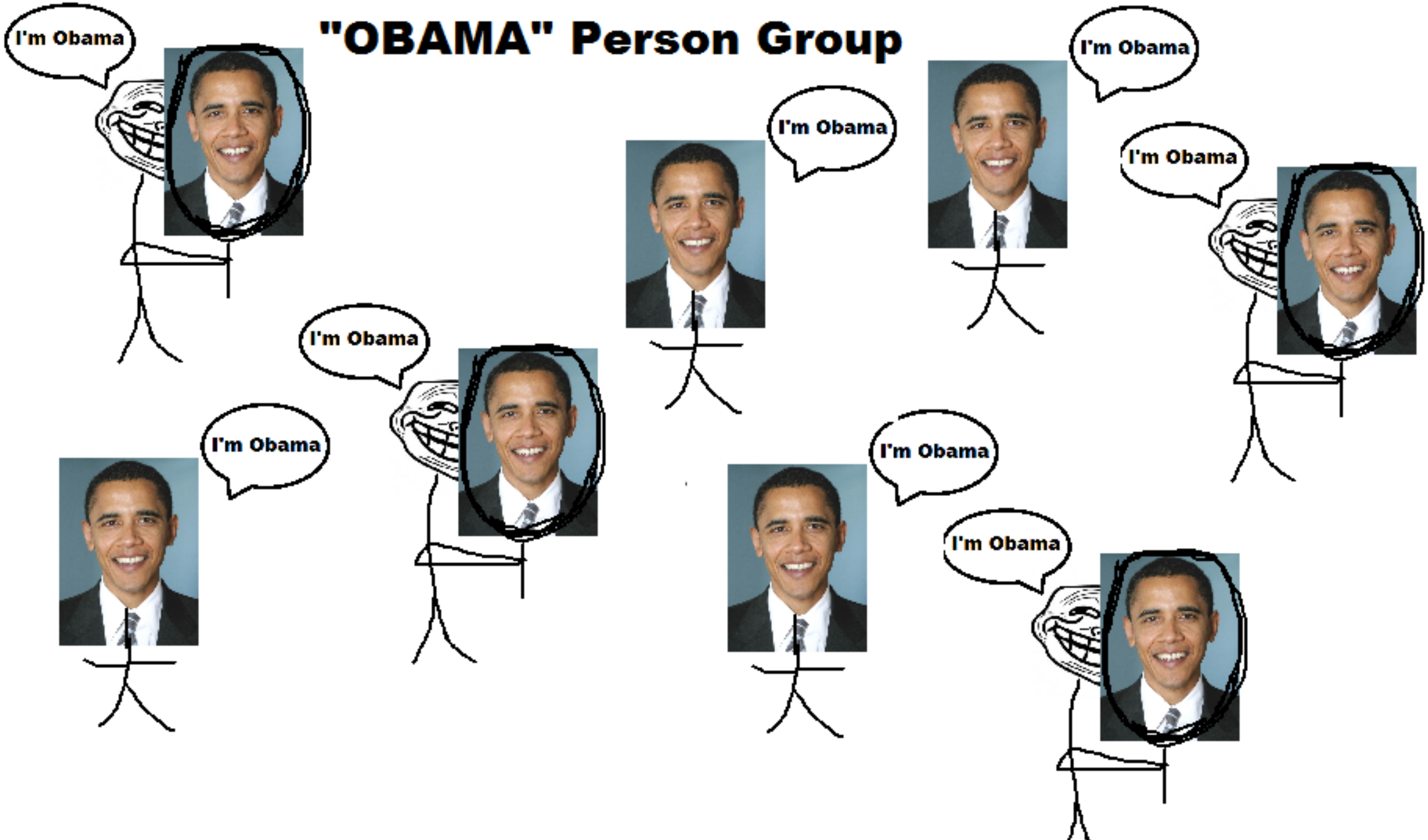
Who can we trust?

Our Approach Specifications

Graphical Representation

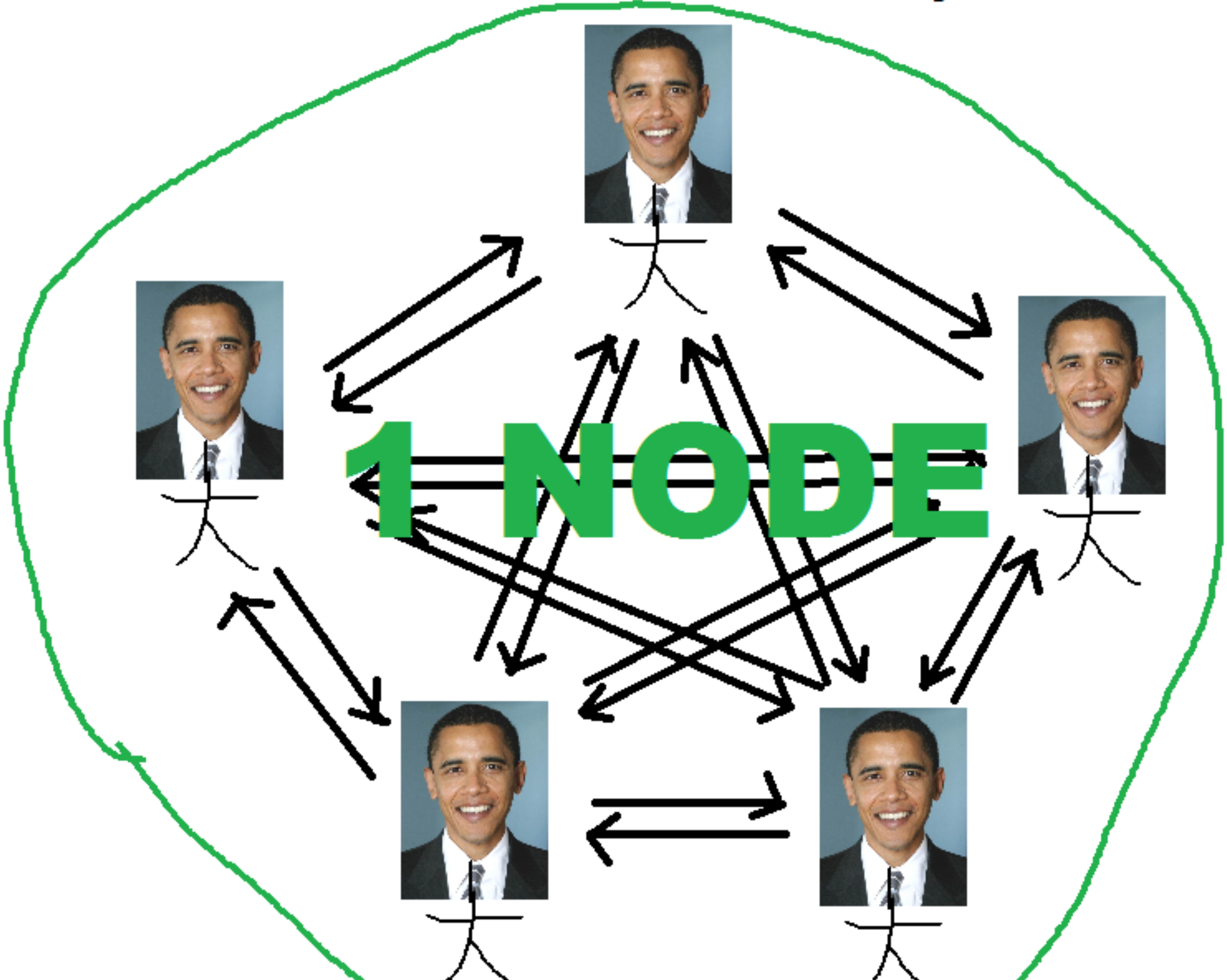
- Key = Node
- A -- signed --> B

Person Group:
All keys claiming to be XX



wolg each real person has 1 key

If the real Obama made 5 keys ...



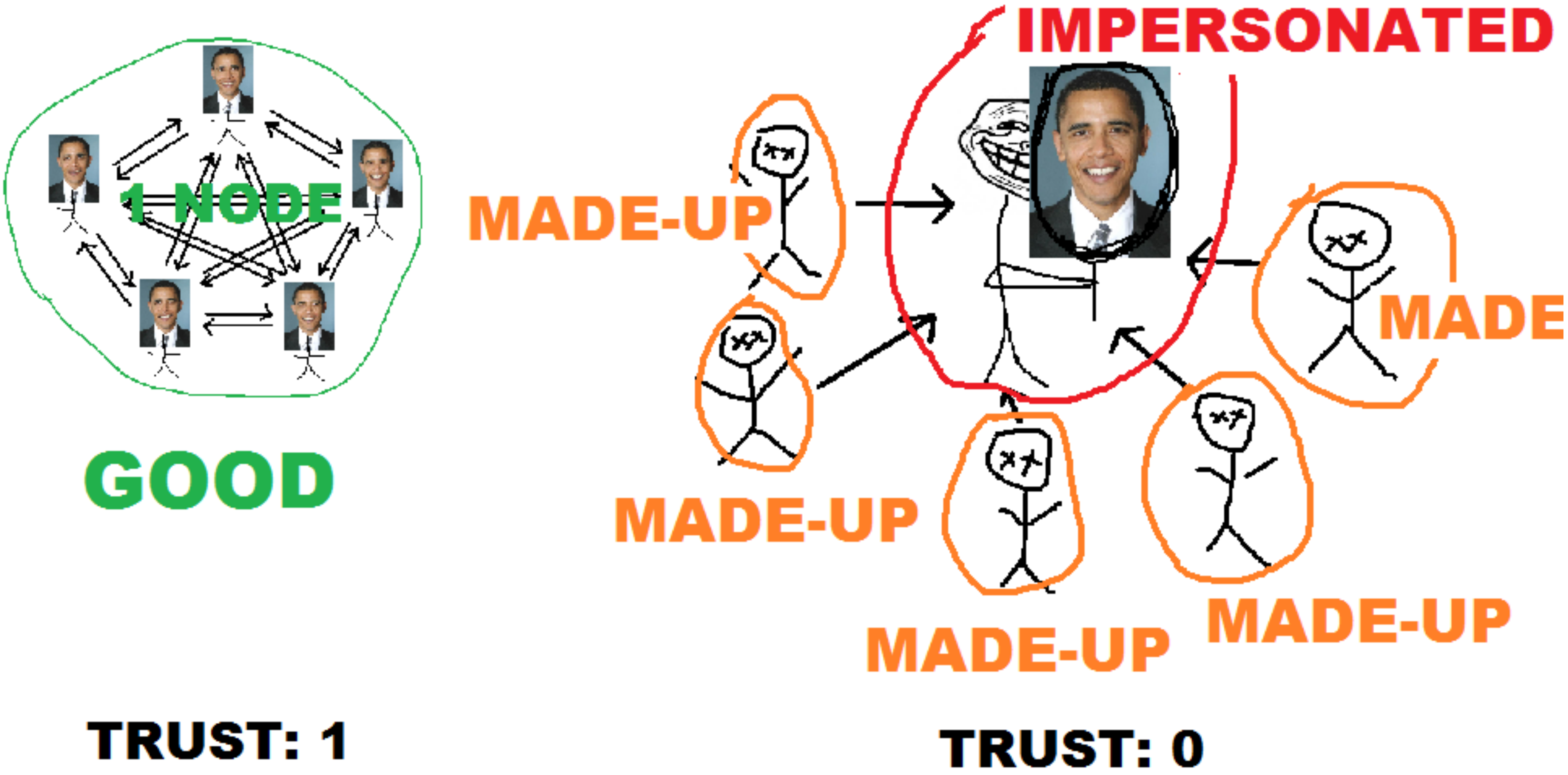
TRUST is Boolean

1 = Trustworthy

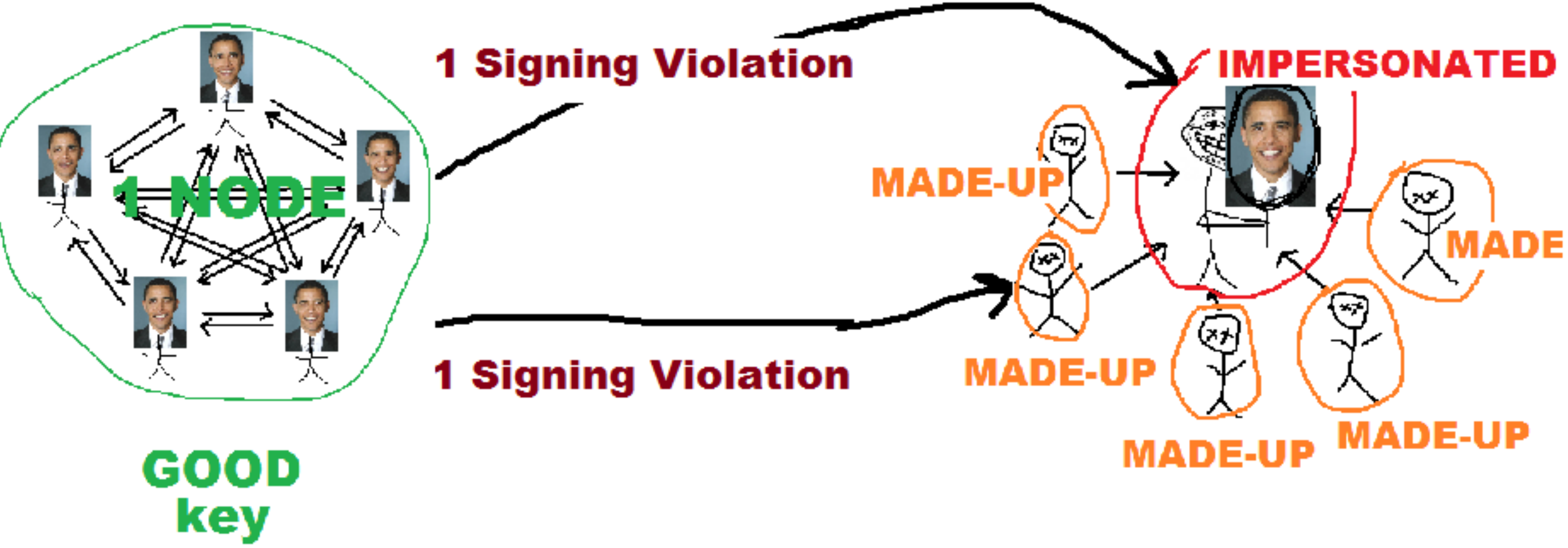
0 = Not Trustworthy

keys:

Good, Impersonated, Made-up



Signing Violation
= Trust 1 signs Trust 0



Trust 1

Trust 0

Goal:

**How to assign trust to each node
to result in
the least violation score?**

Algorithm Inputs

- Directed Graph
- Source person (trust is always 1)
- Person group constraints
- --> calculate trust w.r.t. Source person

Naive Algorithm

Try all possibilities **consistent** with person groups

Consistent
= each person group can have *at most* 1 real person (i.e. 1 key with trust 1)

Exponential time :(

Our Approach: Evolutianary Algorithm

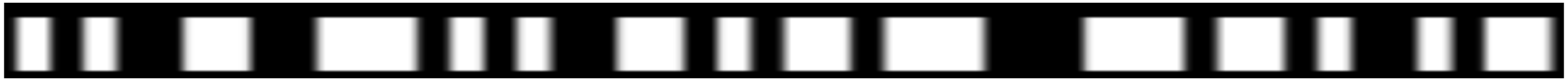
- Gen 0: Randomly assign trust to each key (except src is always 1)

Sort the scores of all 100 assignment sets

- Pick the **best 5** to be parents
- Each parent **produces** 20 children
  - w/ mutations. 1% chance switch trust
- **Score** each of the 100 assignment sets

Results

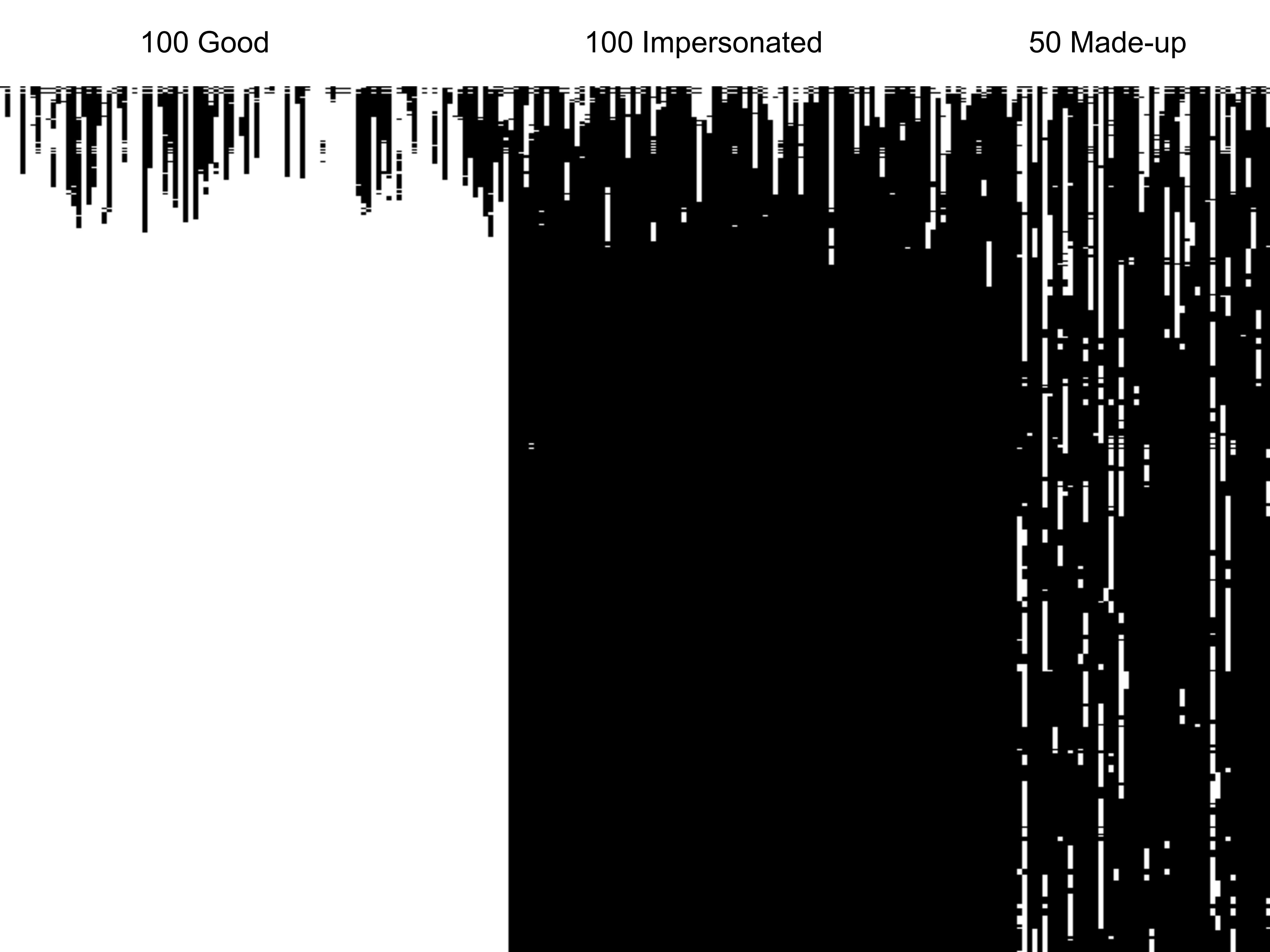
Genetic Alg Time Image



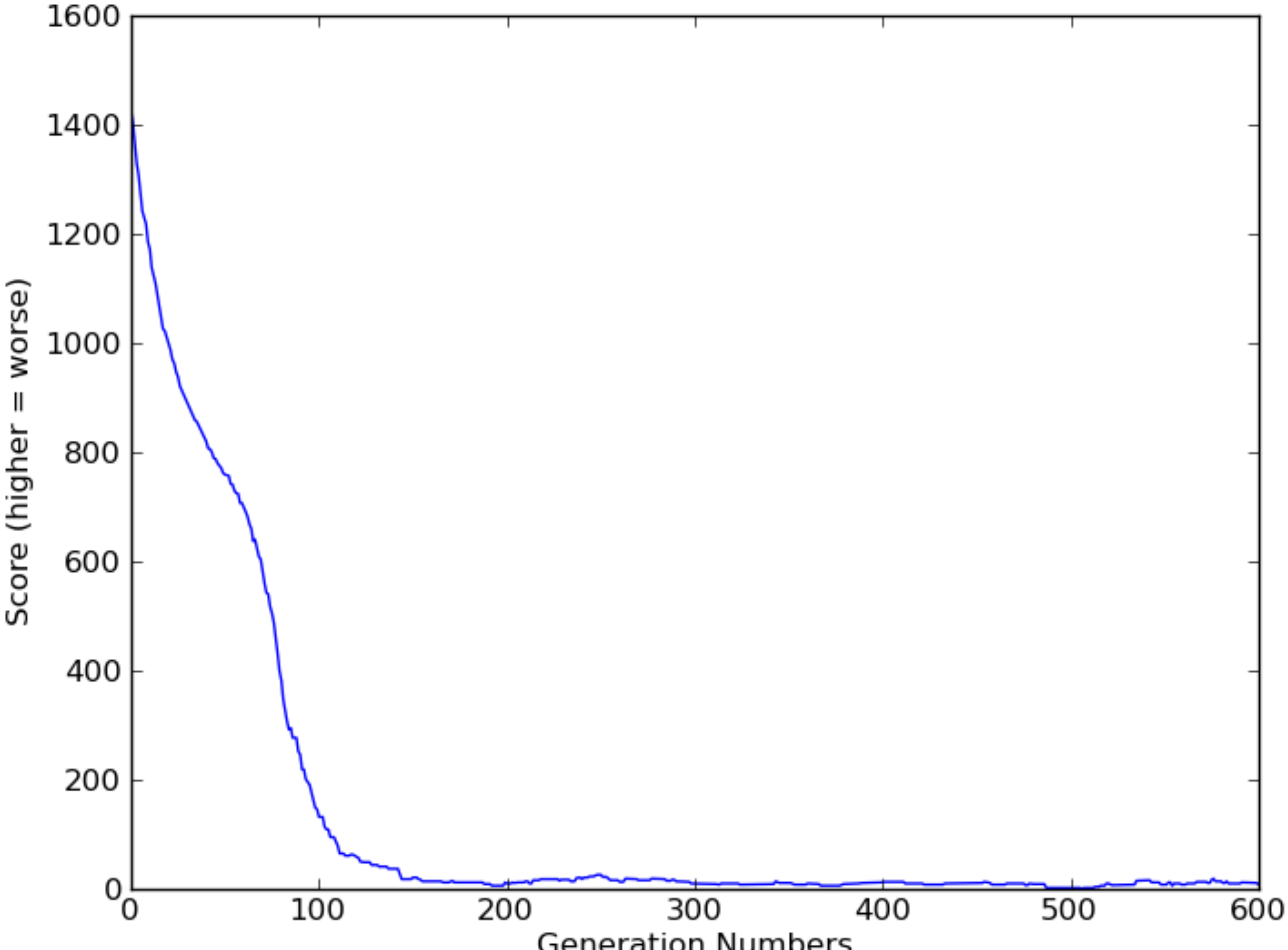
- 1 Row = 1 Generation
- 1 Column = 1 Key

- Trust 1 Key: White
- Trust 0 Key: Black

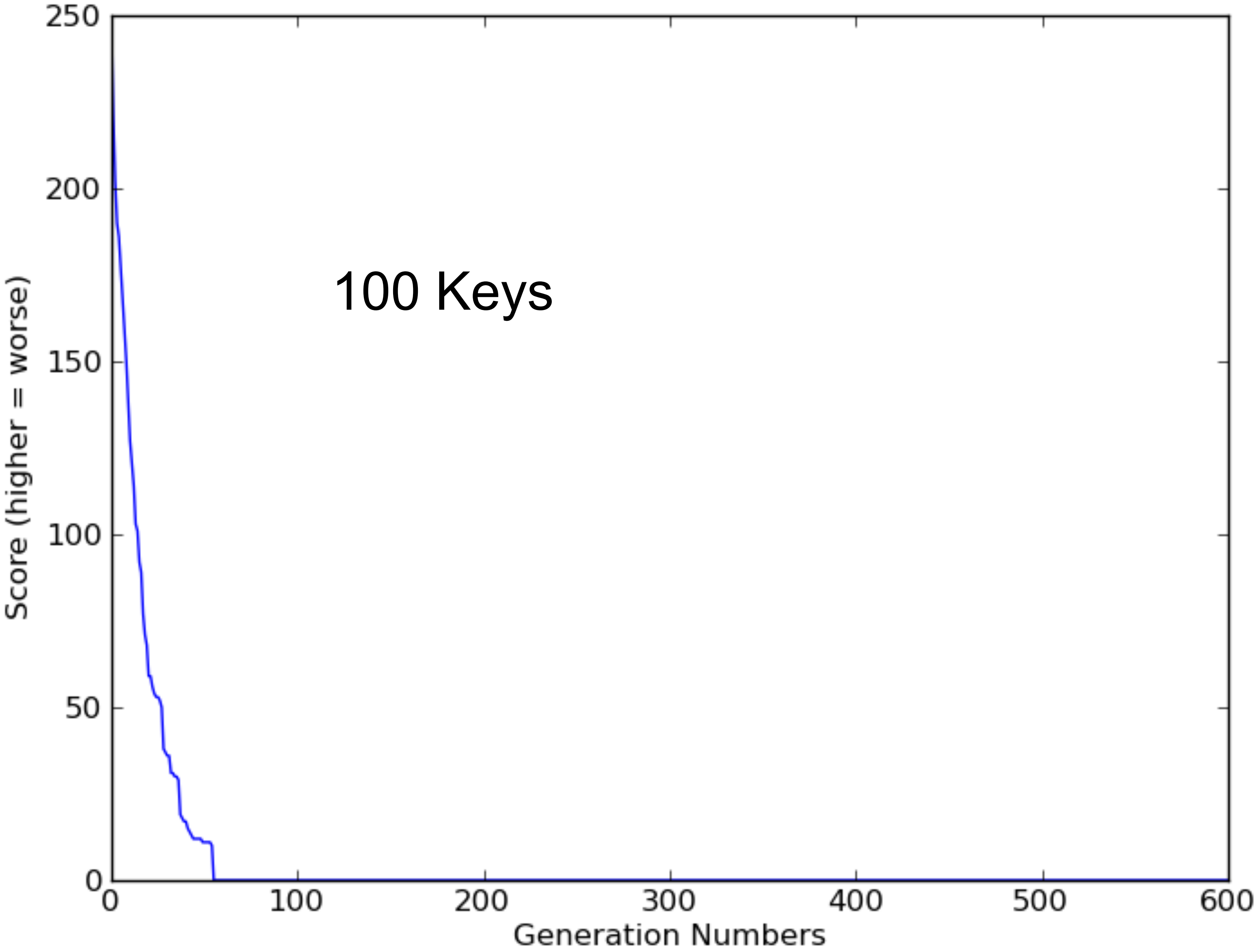
- **Time** goes down
- Generations ++


100 Good
100 Impersonated
50 Made-up

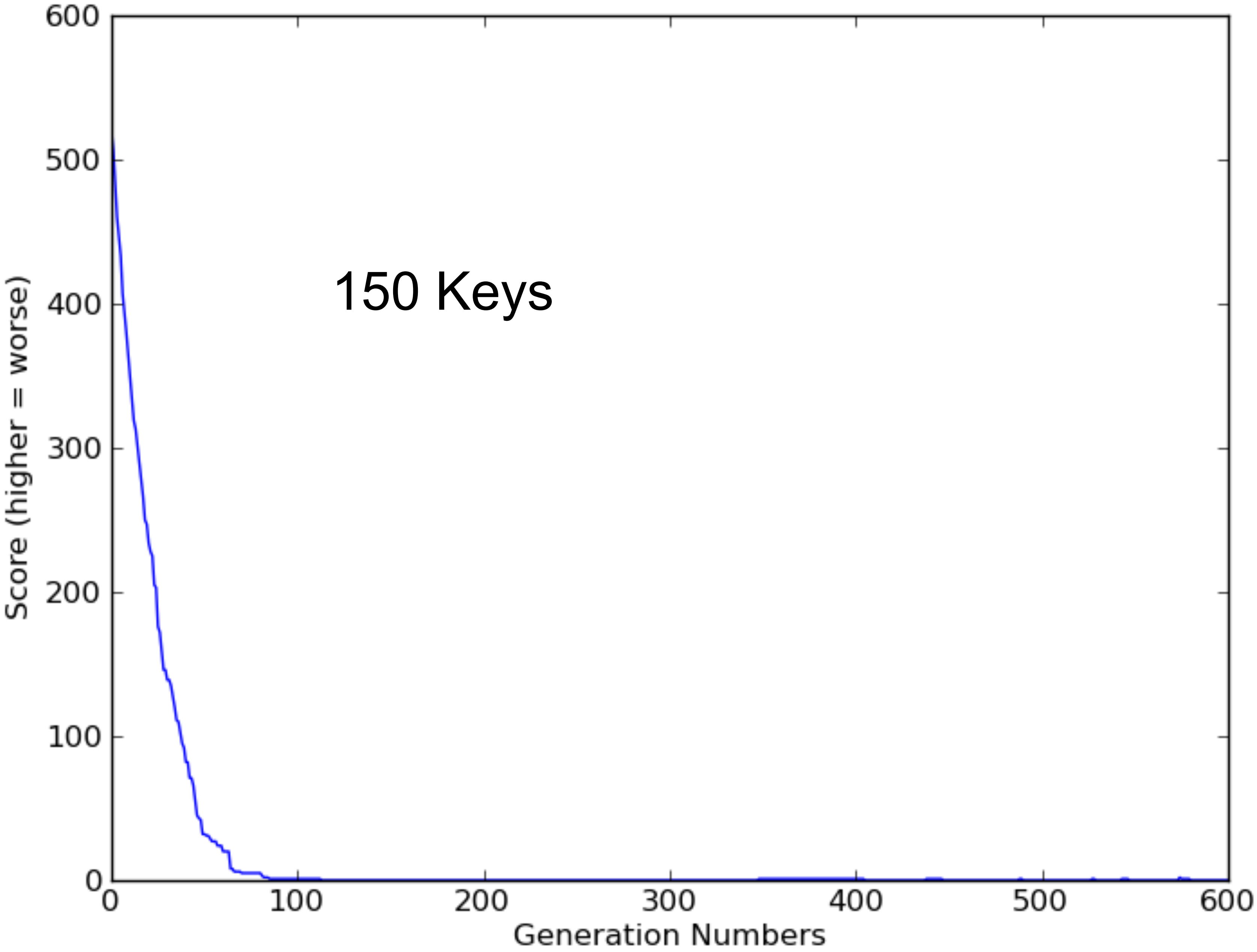
Violation Scoring



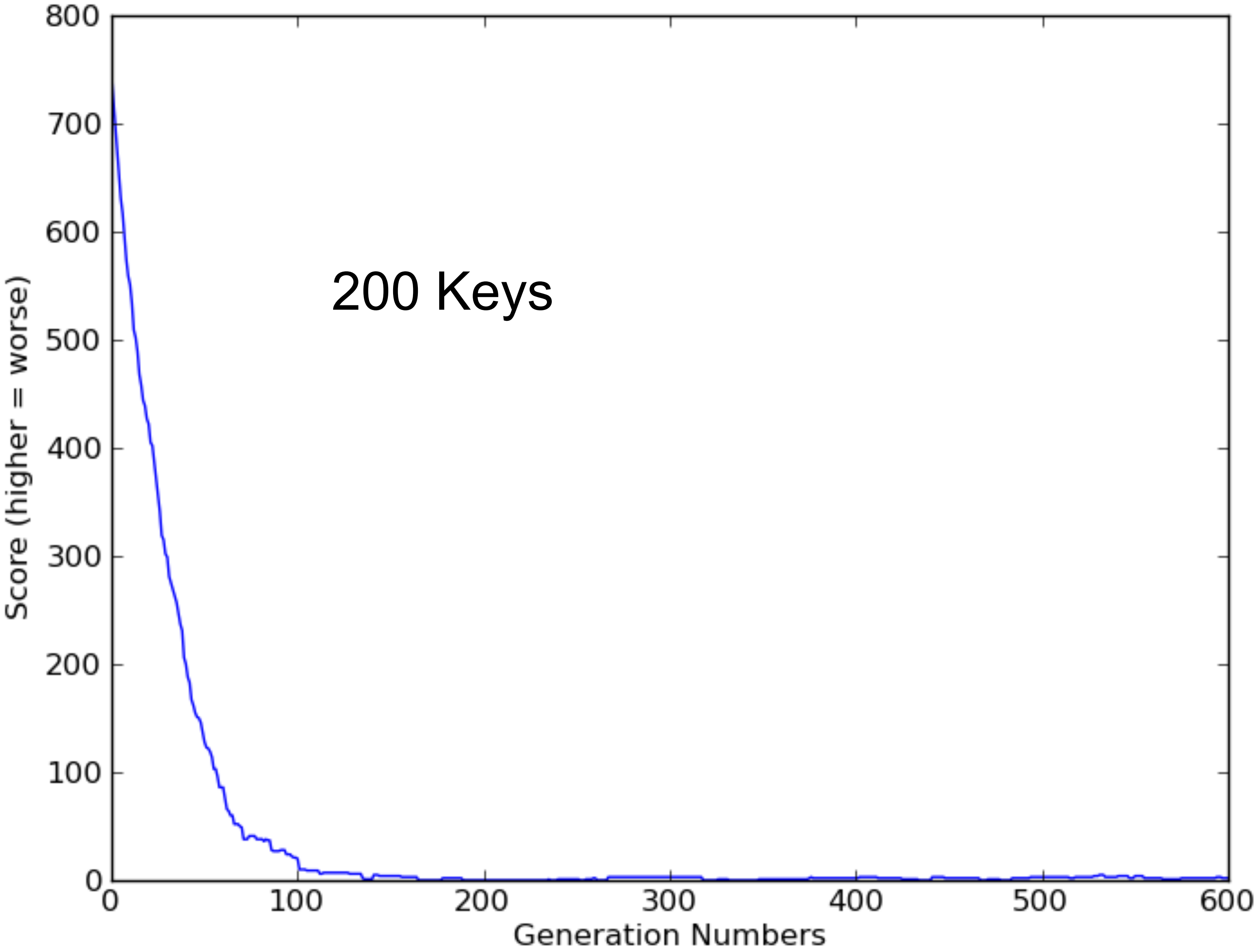
Yes, it runs in Polynomial Time



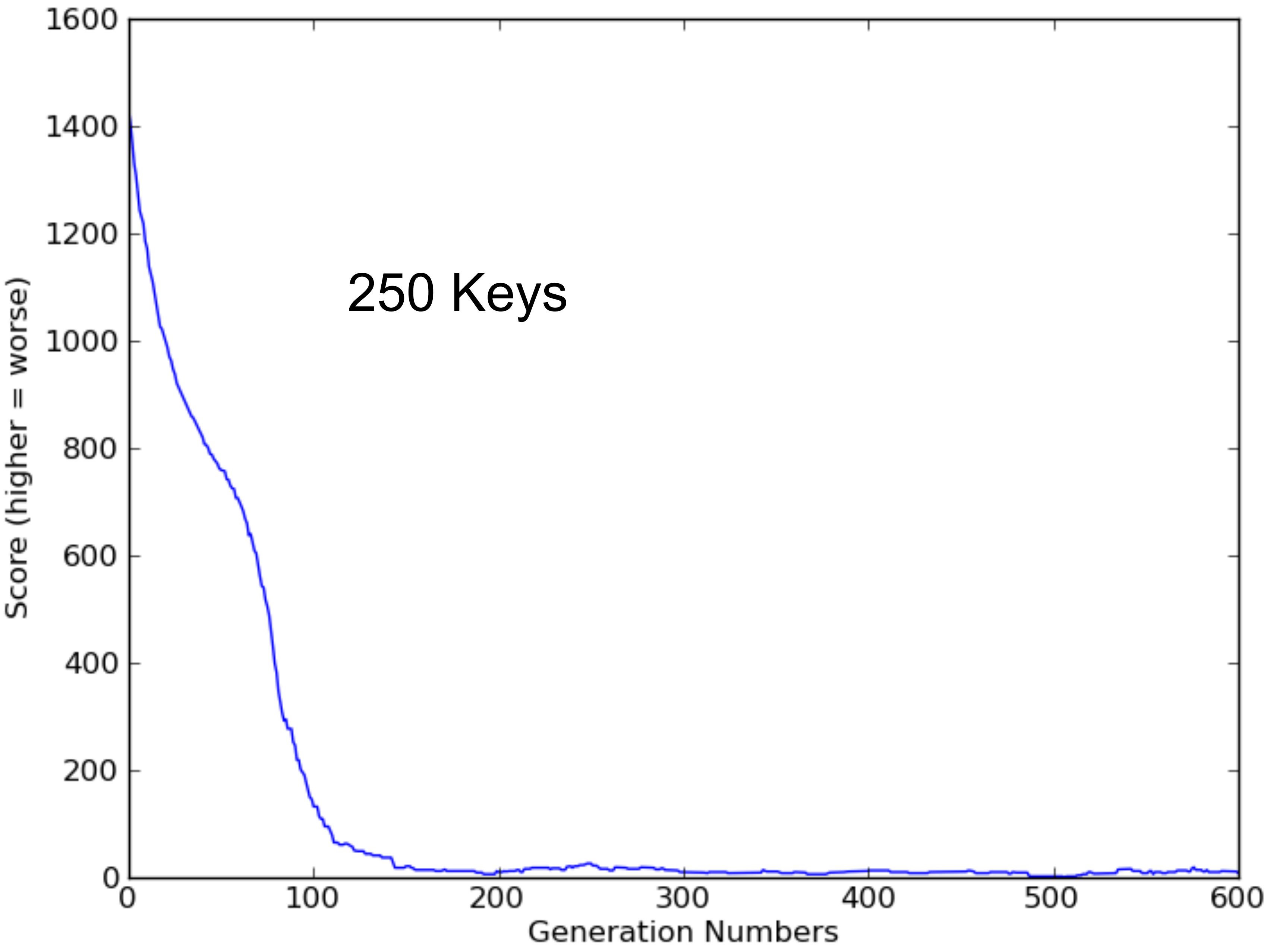
Yes, it runs in Polynomial Time



Yes, it runs in Polynomial Time

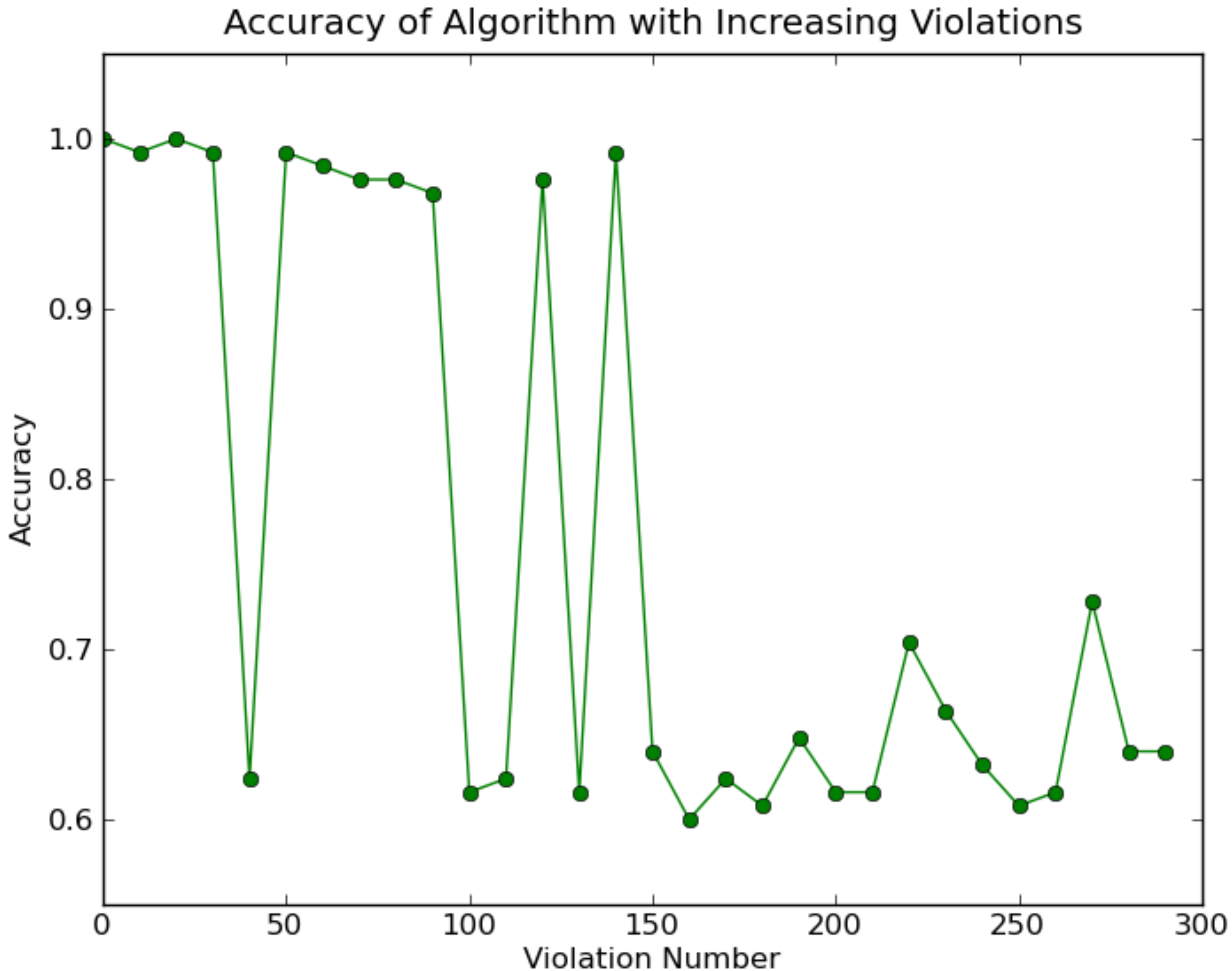


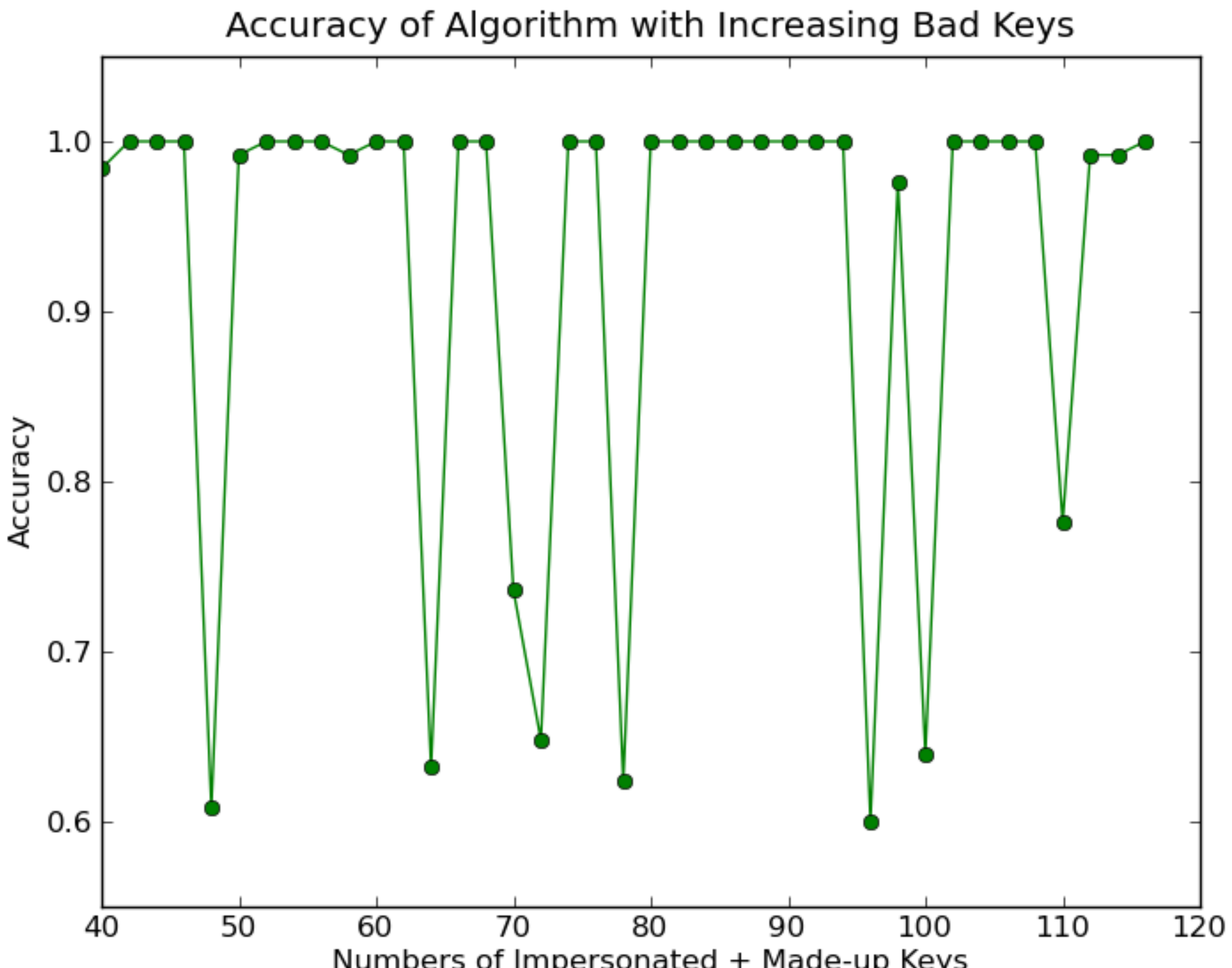
Yes, it runs in Polynomial Time



In Imperfect World (with Violations)

show alg is **resistant to**
- **signing violations**
- high volume of **bad keys**


Accuracy of Algorithm with Increasing Violations
Accuracy
1.0
0.9
0.8
0.7
0.6
Violation Number
0
50
100
150
200
250
300


Accuracy of Algorithm with Increasing Bad Keys
Accuracy
1.0
0.9
0.8
0.7
0.6
40
50
60
70
80
90
100
110
120
Numbers of Impersonated + Made-up Keys

Conclusion

- we made Android signing work

- Given a graph and a node, **used evolutionary algorithm** to assign trust value to each other node

- **Fast**: Runs in sub-exponential time

- **Resistant to signing violations**: (can tolerate about 20% of all certificates signed by good guys
are violations)

- **Resistant to flood of bad guys**: at least 80% of all keys can be either impersonated or madeup without significantly decreasing the accuracy of the algorithm.