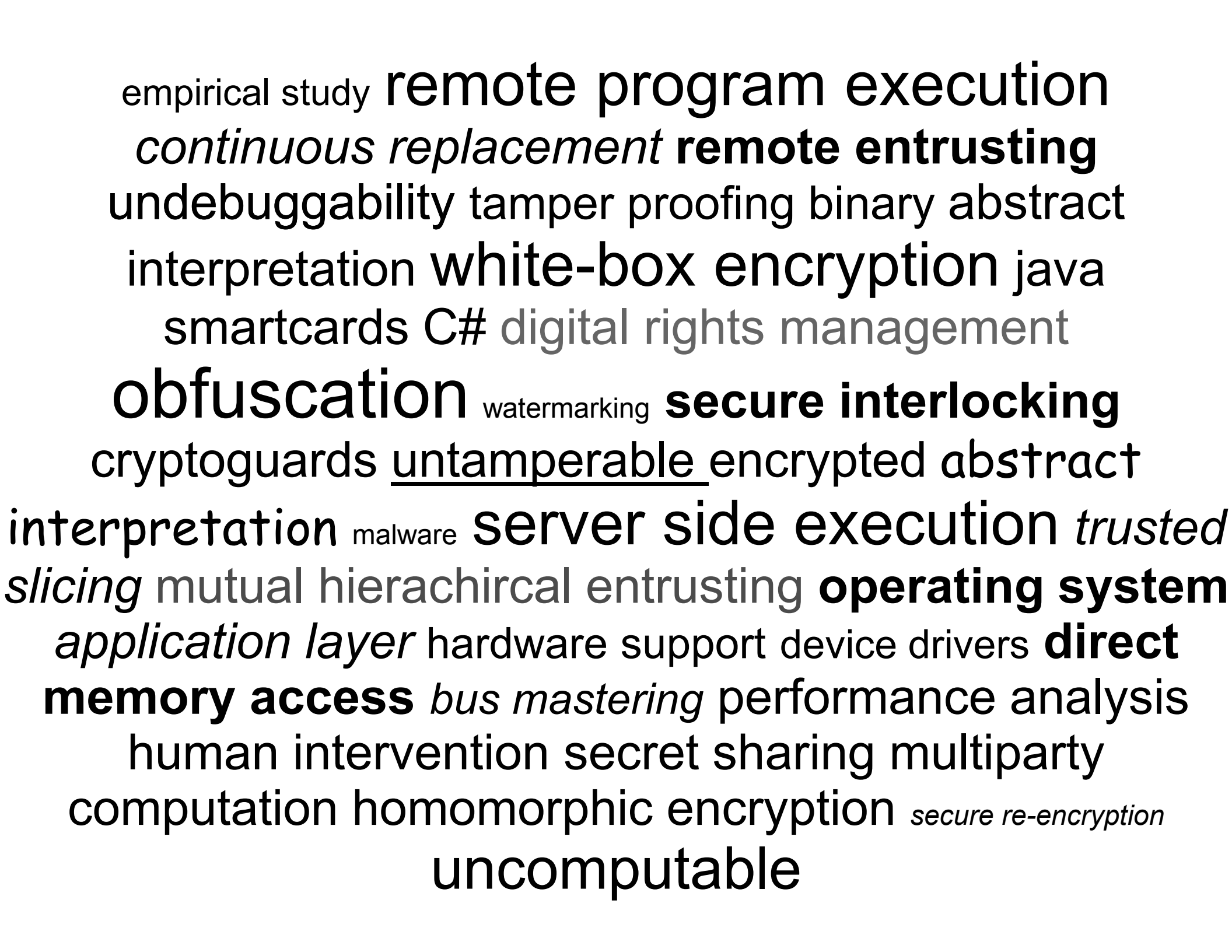
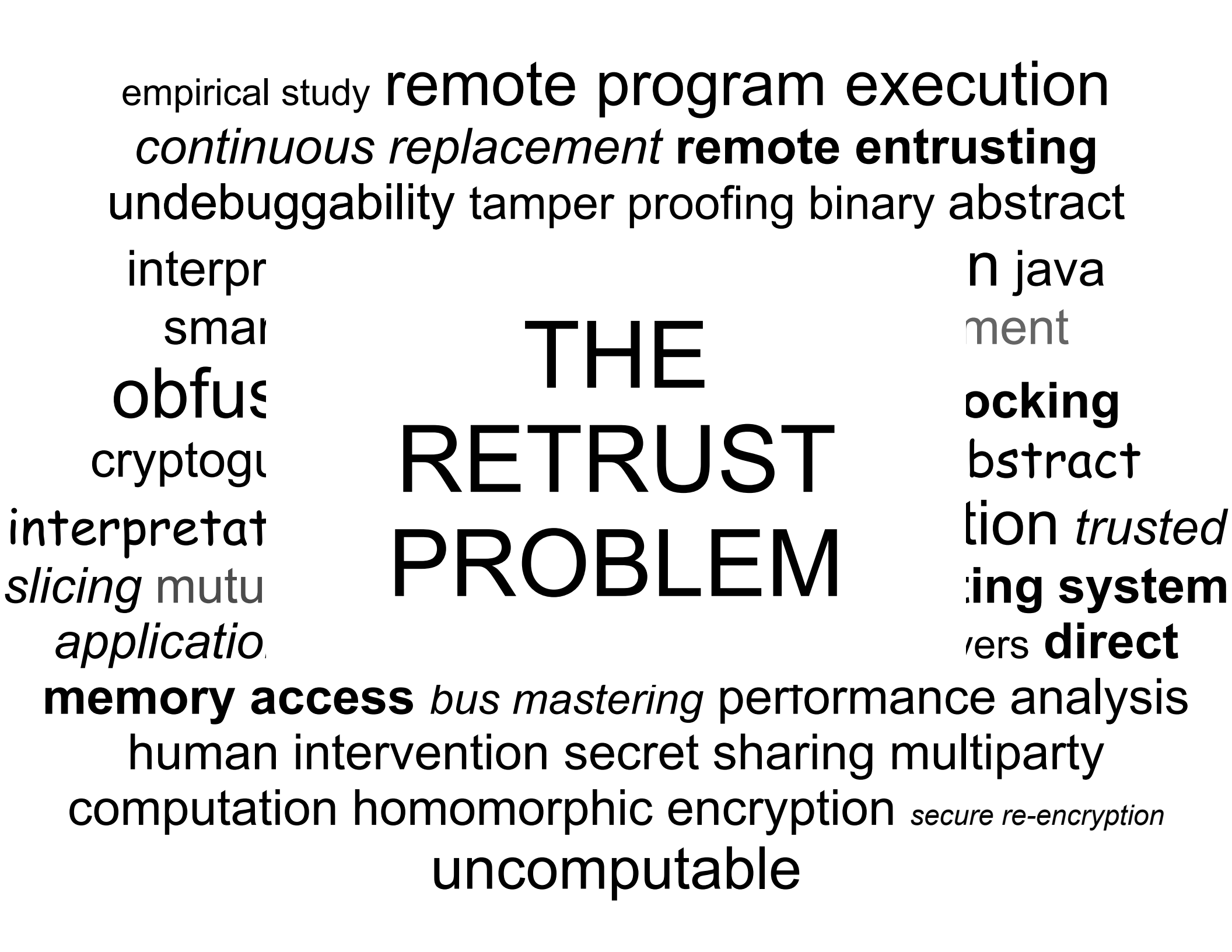

empirical study remote program execution
continuous replacement remote entrusting
undebuggability tamper proofing binary abstract
interpretation white-box encryption java
smartcards C# digital rights management
obfuscation watermarking secure interlocking
cryptoguards untamperable encrypted abstract
interpretation malware server side execution trusted
slicing mutual hierachircal entrusting operating system
application layer hardware support device drivers direct
memory access bus mastering performance analysis
human intervention secret sharing multiparty
computation homomorphic encryption secure re-encryption
uncomputable


empirical study remote program execution
continuous replacement **remote entrusting**
undebuggability tamper proofing binary abstract
interpr
n java
smar
ment
obfus
ocking
cryptogu
bstract
interpretat
tion *trusted*
slicing mutu
ing system
applicatio
vers **direct**
THE RETRUST PROBLEM
memory access *bus mastering* performance analysis
human intervention secret sharing multiparty
computation homomorphic encryption *secure re-encryption*
uncomputable

Outcome

- Project-wide Trust Model
  - Sufficiently Detailed: Comparison Possible
  - Sufficiently Open: No Solution Is Eliminated
  - Sufficiently Practical: Useful for builders
  - Sufficiently Theoretical: Useful for provers
  - Sufficiently Simple: No Time “Wasted”
- Fitting Existing Solutions
  - Trusted Slicing
  - WBRPE

Model Summary

RE-TRUST aims to preserve some set of properties of the set of all execution traces of a remote program.

Additional Properties:

- Confidential properties
- Prevent offline execution
- Amount of work performed by server
- Total work performed

Example Problem: Car Race Game

Attacker's Task:

- Modify a particular variable

Attackers Property:

- State trace of the program except for modification of a specific variable

Defenders Property:

- State trace of a subset of variables

Additional Properties:

- Server is scalable
- Trace is not re-executable on new input

Example Problem: SETI

Attacker's Task:

- Return fake results

Attackers Property:

- State trace of the program except for modification of a specific variable

Defenders Property:

- State trace of a subset of variables

Additional Properties:

- The value of a key remains secret

Example Solutions

Slicing	Static Checksum	WBRPE
Maps from execution trace to state trace	Checksum on the execution trace implies static checksum	