NEW OPTIMAL ALGORITHMS FOR COMPUTING BINARY TREE OVERCOVERS IN RANGE SEARCHABLE ENCRYPTION

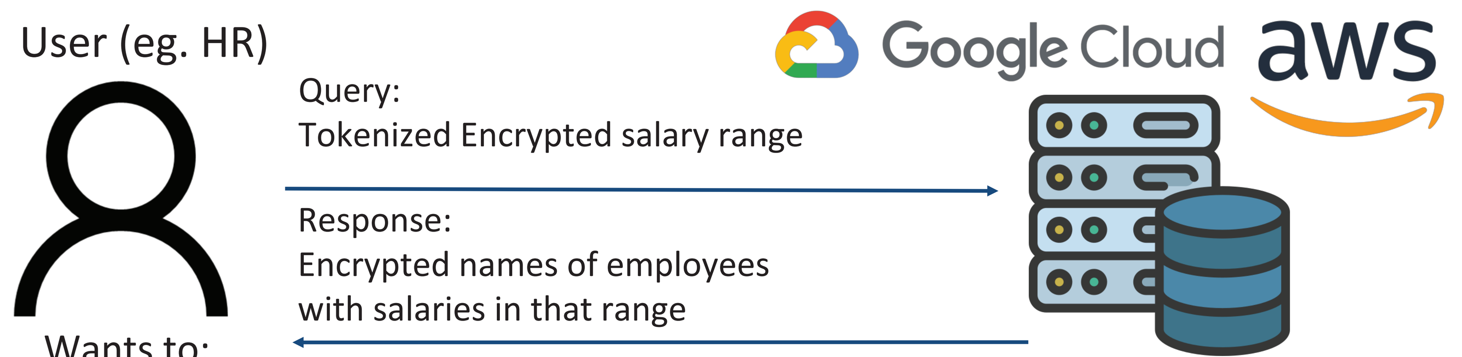
Range Searchable Encryption (RSE):

A way of outsourcing data storage and computation to cloud

Importance

Problems	Solutions
User lacks computation resources to store large database	Take advantage of cloud's speed, accessibility and reliability
Server should not be trusted with sensitive data: data mining/selling, insufficient security	Encrypt database without give server encryption key
User also wants to search for data without downloading everything or revealing search criteria	RSE

Setting



Wants to:
- store employee salary information on cloud by encrypting it
- search for range of salary without giving server key

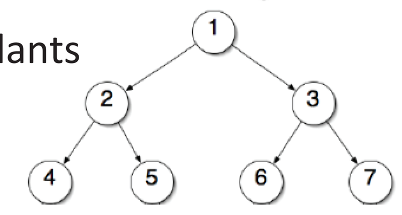
Shouldn't know:
- encryption keys
- salaries and employee names

Our Project: Novel improvements to data encoding and cover generation in RSE

Improvement #1: Data indexing

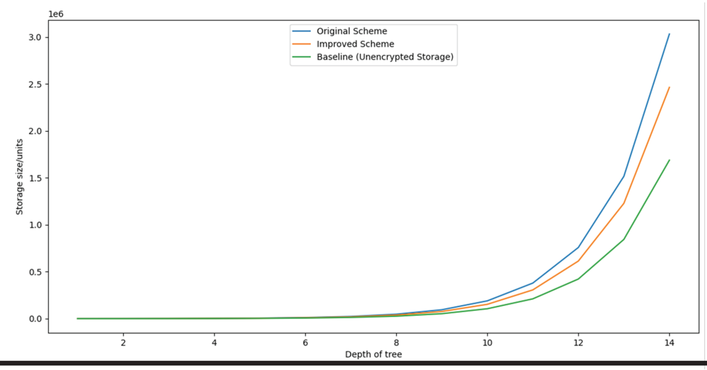
RSE schemes index data with binary trees for efficient searching

Problem: Given node n, derive descendants eg. given 2, return {4,5}



[FJKN+15,1]:	vs	[Ours]:
Lookup table used to associate node to leaves		Ratcheted hashes directly "point" node to descendants at query time

Results:
Improved storage efficiency with lookup table avoided

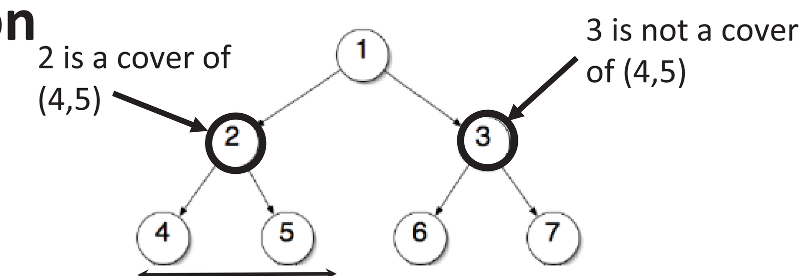


Graph of storage sizes against depth of the tree for different schemes

Improvement #2: Cover Generation

What is a Cover?

A set of nodes wherein their descendants contain all nodes in the range covered by the cover.



Relation to RSE:

Cover algorithms compute a cover for an input range.

This reduces bandwidth and is more secure. Given a cover, RSE schemes can compute and return descendant nodes. The cover algorithm is directly responsible for query and response bandwidth, which is important.

Prior Work & Our Contributions

Cover Algorithms	Non-universal Overcover Algorithm (NUOA)	Universal Overcover Algorithm (UOA)
Exact cover	[FJKN+15,1] and [DPPG16,2]: Optimal Algos	
c-Overcover	[DPPRG16]: Heuristic Algo for c=1	[FJKN+15]: Heuristic Algo for c=3
	[Ours]: New optimal algos for arbitrary c + Mathematical proofs	
	[Ours]: (Faster) Optimal Algorithm for c=1,2	**[Ours]: (Fast) Heuristic Algorithm for c=4,5**

Algorithm Design

Algorithm Design	NUOA	UOA
Cover Types	**Non-universal c-overcover**	**Universal c-overcover**
Traits	restricted number of cover nodes	+ covers of same range size appear indistinguishable to the server
Pros	improve query bandwidth	+ more information hiding from server

Dynamic Programming: "Inductive problem solving"

1. Parameterise the Problem
2. Solve Base Cases
3. Break Down the Problem
4. Memoization(Store computed results)

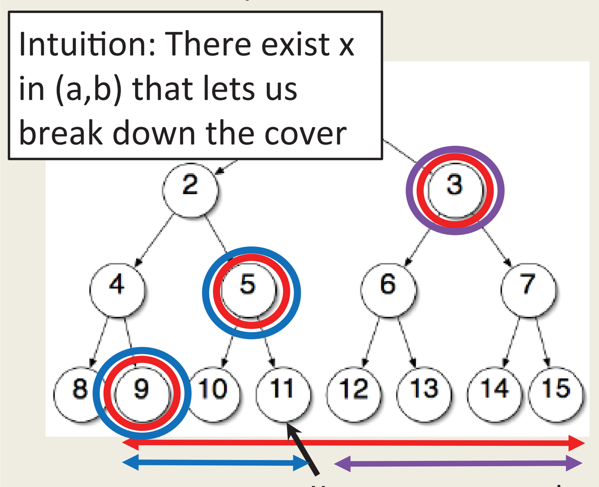
Mathematically Proven Optimal!

Results:
Both types of overcovers are practical as e is relatively insignificant when c is large enough.

NUOA

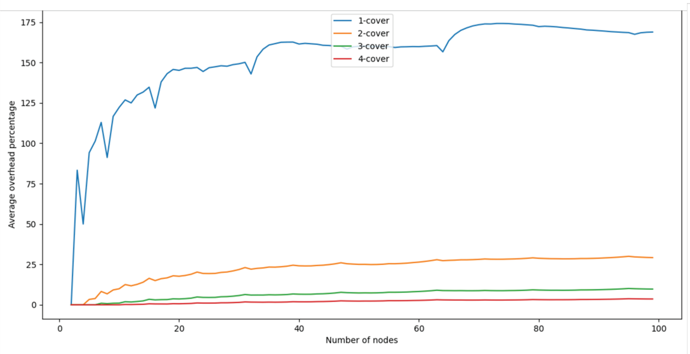
1. Given range **(a,b)**, give cover with **c** number of nodes that has minimum e. Build: T[a,b,c] → c-cover
2. Solve all possible T[a,b,1]→ 1-cover (trivial) for a,b in tree
3. Find a way break down c-cover:

Intuition: There exist x in (a,b) that lets us break down the cover



Result: T[9,15,3]→T[9,11,2] U T[12,15,1]

Repeat: T[9,11,2]→T[9,9,1] U T[10,11,1]



Graph of overhead percentage(e/r) against range size for non-universal overcovers

UOA

Differences

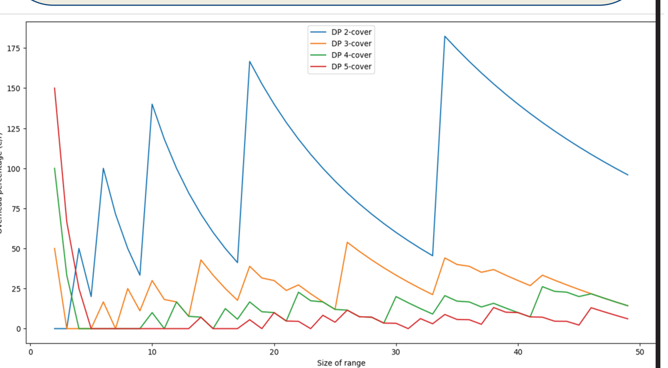
Consider **all ranges (a,b)** of the same range size (r=b-a+1)

1. Find covers that have the same number of **extra nodes** (e)
2. Consider extra nodes on **left and right** (e1 and e2)

Problem Solving Sketch

Following dynamic programming steps:

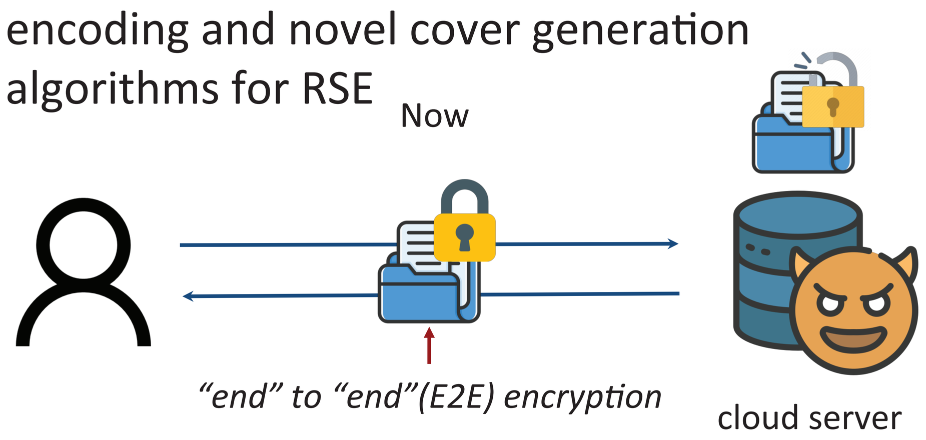
1. **Build**: T[a,b,c,e] → c-cover
2. **Base case**: T[a,b,1,e1] (building from left to right
3. **Break down**:
   a. Break right side error off:
   T[a,b,c,e] → T[a,x,c-1,e1] U T[x+1,b,1,e2]
   a. Break left side cover by cover
   T[a,x,c-1,e1] → T[a,y,c-2,e1] U T[y+1,x,1,0]
1. Increase e until all ranges (a,b) have valid covers with e



Graph of overhead percentage(e/r) against range size for Universal overcovers

Impact

Novel improvements to data encoding and novel cover generation algorithms for RSE



Designers of cloud storage systems have more options in designing RSE

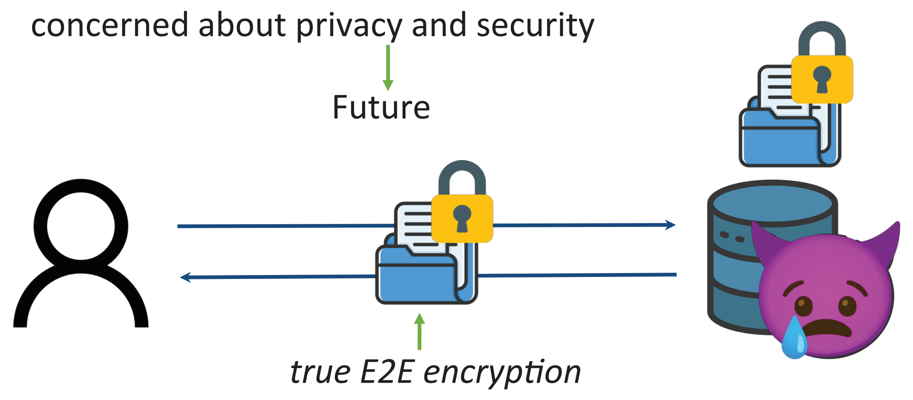
↓

Practical implementations of RSE schemes (only proposed in literature now)

↓

Better E2E encryption systems available to users concerned about privacy and security

↓



Future work

- Further optimization of algorithms with proofs
- Open problems: overcover algorithms that work for documents of inconsistent size
- A complete implementation of RSE with benchmarking

References:

[1] Demertzis, I., Papadopoulos, S., Papapetrou, O., Deligiannakis, A. & Garofalakis, M. Practical private range search revisited. p3-7 Proceedings of the 2016 International Conference on Management of Data (2016). doi:10.1145/2882903.2882911

[2] Faber, S. et al. Rich queries on encrypted data: Beyond exact matches. p3-8 Computer Security -- ESORICS 2015 123–145 (2015). doi:10.1007/978-3-319-24177-7_7

Members:

Richard Ong Jun Quan, NUS High School of Mathematics and Science

Claire Guan Keer, Nanyang Girls' High School

Mentor:

Dr Ruth Ng Ii-Yung, DSO National Laboratories

YDSP Young Defence Scientists Programme

DSTA Defence Science & Technology Agency

DSO National Laboratories