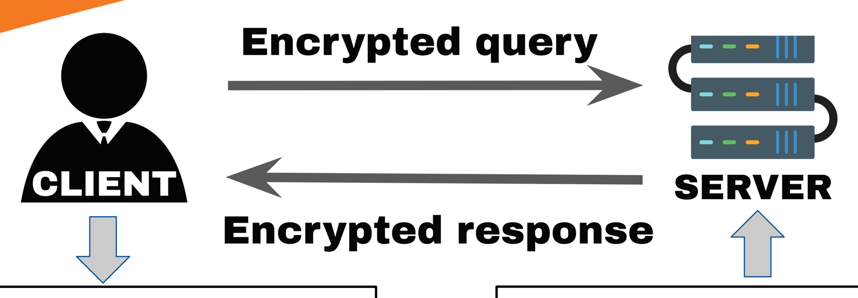
VOLUME-HIDING DICTIONARY ENCRYPTION: NEW SCHEMES AND BENCHMARKING RESULTS



Wants to store dictionary on cloud → Data needs to be **secure**: utilise encryption schemes using STE

Structured Encryption (STE)

Dictionary Example:

Label (Key)	Value
Mouse	Mickey
Dog	Pluto, Goofy
Cat	Tom

However, standard STE returned will reveal **differences in query length (eg.** Dog returns 2 values while Cat returns 1 value

Need more secure STE that achieves **volume-hiding: same value length** returned for every query: **secure** (adversary learns less about query)

Current literature	What we did to improve
Various schemes introduced, but **not optimised**	• **Came up** with several **improvements** and selected **best improvements** after experimentation • **Constructed** an **improved variant of each scheme**
Schemes studied individually, and **not compared**	**Benchmark improved variants** to identify **most suitable schemes** for different types of datasets

Comparison Metrics: **Storage** and **Query Bandwidth (QB)**, since security of all schemes are the same: volume-hiding

INTRA-SCHEME COMPARISON

Using **novel techniques**, we **improved existing schemes.** The table below shows **all the improvements we experimented with** (the ones in **green** were **implemented** eventually). The results show the **magnitude** of the improvements (**negative values preferred**: shows reduction in storage and QB)

Existing Encryption Schemes	Novel Techniques					Our Schemes	Results: New vs Old Storage	Results: New vs Old QB
Naive Volume-Hiding (NVH) [KM19]	Parametrization	Record length of values	Storing start & end position of values			PVH	-78.6%	+15.1%
Greedy Graph Volume-Hiding (GVH) [NPG14]	Graph-matching	Storing used edges	Storing used counters	Bitmap	Frog-hopping	New GVH	-9.2%	-64.0%
Bucket Volume-Hiding (BVH) [KM19]	Modified bit-map	Frog-hopping	Bitmap-froghopping fusion			New BVH	-6.3%	-25.4%
Cuckoo Volume-Hiding (CVH) [PPYY19]	3-bit map					New CVH	-9.6%	-19.0%

Our novel improvements to state-of-the-art schemes significantly improve trade-offs, making the schemes more practical for real world use.

Literature: Naive Volume-Hiding (NVH)

Pad all values to the same maximum length

Label (Key)	Value
Mouse	Mickey||pad
Dog	Pluto||Goofy||pad
Cat	Tom||pad

⊖ Large memory size - large amount of padding to achieve same maximum length

★ **Improvement #1: Parametrization:**
- Truncate hashed labels to fixed length **h** and concatenate "collided" values together
- Pad all "new" values to the same max. length

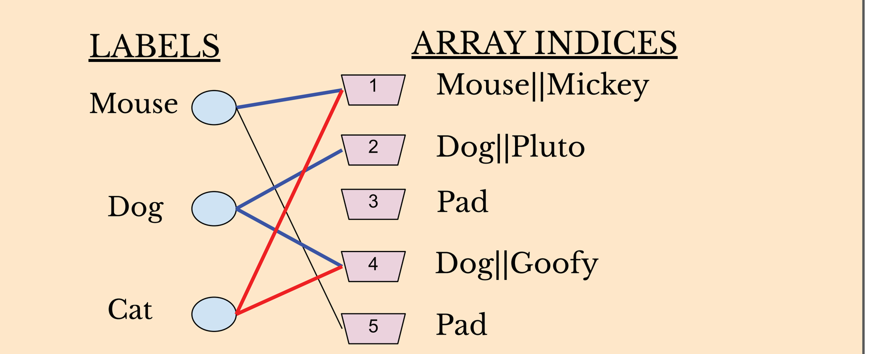
★ **Improvement #2: Encoding**
- Attach only one of each label, then the number of bits in its corresponding values (for easier identification of the labels the values belong to)

PVH achieves significantly better storage due to parametrization & mitigates QB blow up with encoding

Our scheme: Parametrised Volume-Hiding (PVH)

Label (Key)	Value
Mouse.truncate(h) / Cat.truncate(h)	Mouse||6||Mickey||Cat||3||Tom||pad
Dog.truncate(h)	Dog||10||Pluto||Goofy||pad

Literature: Greedy Graph Volume-Hiding (GVH)



⊖ Matching fails: Cat is not assigned an index in the array

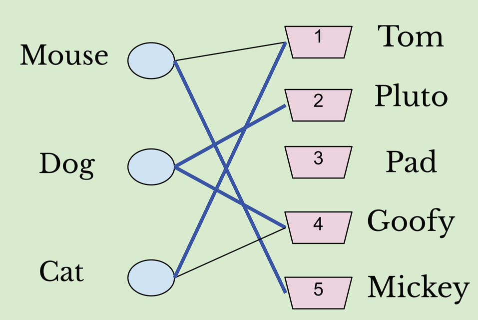
- Assignment of data to indices done via graph matching
  - Greedy matching not optimal, leads to GVH failing

★ **Improvement #1:** Optimal maximum bipartite graph matching reduces failure rate at no cost (Values may be stored with labels to disambiguate)

★ **Improvement #2:** Encode alternate data structure storing hash indices

New GVH achieves better success

Our scheme: New GVH



LABELS	STORES
Mouse	[2]
Dog	[1, 2]
Cat	[1]

For BVH & CVH, similar improvements were implemented, resulting in new BVH and new CVH respectively. Both new schemes have lower storage and QB. Next, we implemented all of our new schemes on Zipfian and linear datasets, with varying value lengths from 2^{12} to 2^{18}

INTER-SCHEME COMPARISON

CONCLUSION

★ TAKEWAYS
- ☆ CVH is the most suitable for Zipfian datasets
- ☆ PVH is the most suitable for linear datasets

★ IMPACT
- ☆ Our research will be impactful to people who are looking to store sensitive data on external cloud servers, as it:
  - Makes the encryption more secure due to the volume-hiding nature
  - Reduces storage and query bandwidth, minimising costs
  - Recommends most suitable scheme for each type of dataset

Zipfian Storage: NVH > PVH > BVH > GVH > CVH
Zipfian QB: BVH > GVH ~ CVH > PVH > NVH
Linear Storage: BVH > CVH ~ GVH ~ PVH > NVH
Linear QB: BVH > CVH ~ PVH > GVH > NVH

★ FUTURE WORK
- ☆ Exploring different definitions of security and efficiency (e.g. time efficiency)
- ☆ Fine-tuning our parameters to further optimize the schemes
- ☆ Look at dynamic datasets where information can be added or updated

REFERENCES

[1] Kamara, S., & Moataz, T. (2019). Computationally volume-hiding structured encryption. *Advances in Cryptology – EUROCRYPT 2019*, 183–213. https://doi.org/10.1007/978-3-030-17656-3_7

[2] Patel, S., Persiano, G., Yeo, K., & Yung, M. (2019). Mitigating leakage in secure cloud-hosted data structures. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. https://doi.org/10.1145/3319535.3354213

[3] Naveed, M., Prabhakaran, M., & Gunter, C. A. (2014). Dynamic searchable encryption via blind storage. *2014 IEEE Symposium on Security and Privacy*. https://doi.org/10.1109/sp.2014.47

[4] Chase, M., & Kamara, S. (2010). Structured encryption and controlled disclosure. *Advances in Cryptology - ASIACRYPT 2010*, 577–594. https://doi.org/10.1007/978-3-642-17373-8_33

Members:
Jemma Lee Miin Yee, Raffles Institution
Cadence Wern Sea Loh, Raffles Institution
Sheng Yu Fei Carol, Raffles Institution
Mentor:
Dr Ruth Ng Ii-Yung, DSO National Laboratories

YDSP Young Defence Scientists Programme
DSTA Defence Science & Technology Agency
DSO National Laboratories