YDSP Young Defence Scientists Programme

DSTA Defence Science & Technology Agency

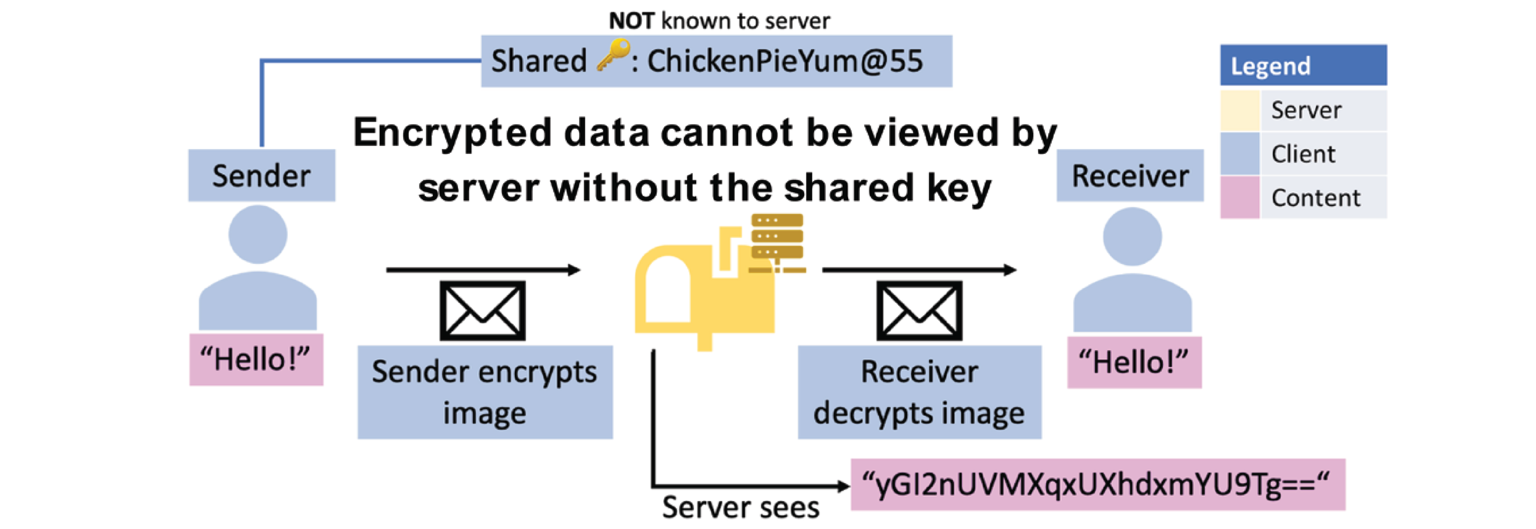
DSO National Laboratories

NOVEL APPROACHES TO CONTENT MODERATION OF END-TO-END ENCRYPTED IMAGES USING PERCEPTUAL HASHES

Members:
Mantha Akshara, Peng Ruijia, Tan Siying
(Raffles Institution)

Mentors:
Chan Ming Kai Alvin, Ruth Ng Ii-Yung
(DSO National Laboratories)

1. Background: Content Moderation of E2EE Images



A) Setting: E2EE Image Communication

Two users (Sender, Receiver) send images via an untrusted server. The server should never get access to the original content being sent, **preserving user privacy**.

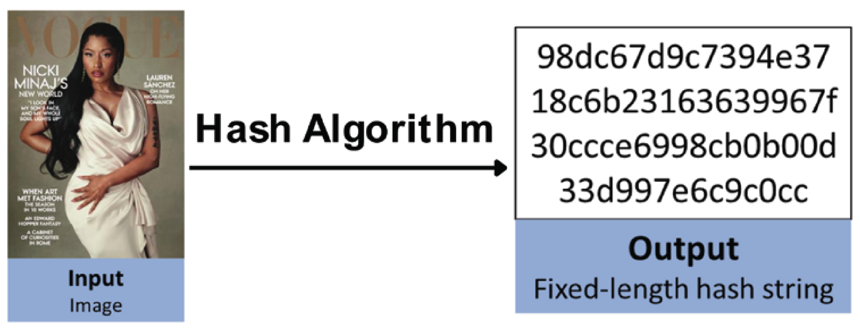
B) Challenge: Content Moderation with E2EE

Server wishes to block undesirable ("R21") images without affecting neutral ("PG") images. **But how does the server efficiently do this without decrypting the images?**

C) Our Solution: Improved Content Moderation in E2EE

Prior work proposed various "perceptual hashes" for comparing encrypted images with known databases of R21 content. We consolidate and improve upon this work by combining them using a novel **decision tree.**

2. Perceptual Hashes of Images



Input : Image
Output : Fixed length hash

- Hashes can be **compared** to determine whether two images are visually similar.

We utilised Difference Hash (dHash), Perceptual Hash (pHash), Wavelet Hash (wHash) [1] and Non-negative Matrix Factorisation Hash (NMFHash) [2] in our work.

3. Our Work

❓: Can the accuracy of perceptual hashes in detecting visually similar images be improved via a combination of perceptual hash algorithms?

Our Contributions

1) 🔍 **Survey** of Perceptual Hashes
2) 📝 Combined hashes into **novel, Decision Tree** approach
3) 🧪 **Testing** on real-world datasets
4) 📚 **Python library** and **proof-of-concept application**

4. Novel Decision Tree Approach to Content Moderation using combination of Perceptual Hashes

4.1 Server - Client Protocol for E2EE images

1. Setup: Server hashes all images in database using all hash algorithms and stores resultant hashes

dhash	phash	whash	nmfhash
c9b4063a(	ed55f2ce6	fc00ff0ee7	W!!_!!w!!'
aa16a9d6;	c174b480	000074fe5	P!!K!!Q!!e

2. Send: Image to be sent is hashed client-side to produce a list of four hashes. Hashes are appended to encrypted image data and sent to server.

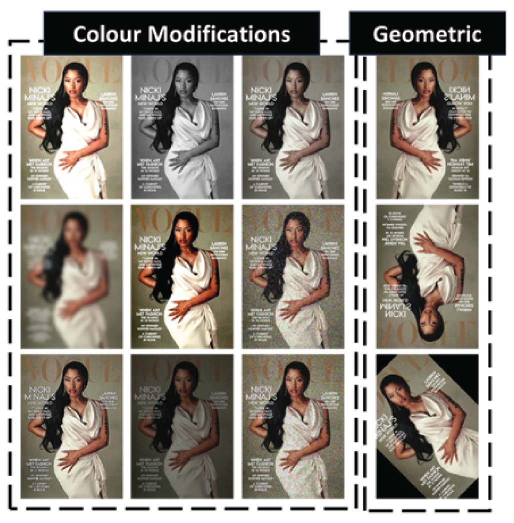
vT...AM|c9b4...25b2|ed54...17e8|fc00...dd00|W!!_...!;!!
Encrypted Image Data|dHash|pHash|wHash|NMFHash

3. Similarity: Server calculates a similarity score for each pair of hashes

0.939 | 0.848 | 0.799 | 0.977 *arbitrary values

4. Verdict: Server passes similarity scores into decision tree to check if image is visually similar to any images in the database.

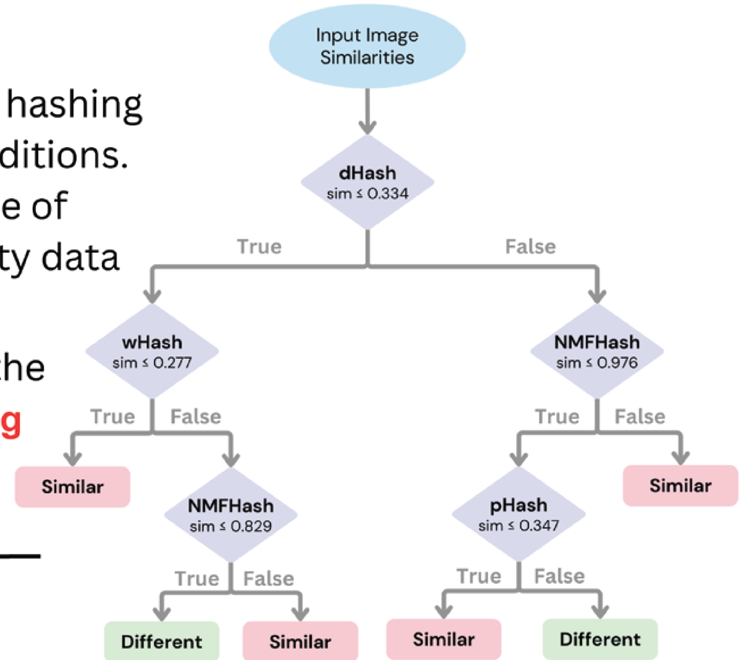
4.2 Visually similar?



Malicious users may apply **colour** or **geometric** filters to their images, creating **modified** images to circumvent detection. Hence, we test the decision tree's ability to detect that these **modified** images are visually similar to the **original** image.

4.3 Our Decision Tree

- Each decision node makes use of one of the hashing algorithms with their specific threshold conditions.
- Threshold values are obtained by making use of machine learning to find patterns in similarity data from all four hash algorithms as a whole.
- Machine learning model **does NOT analyse** the original images directly -- privacy-preserving
- **Decision Tree** remains static after its initial construction



4.4 Evaluation Methodology

A) Baseline Approaches

We compared our decision tree approach to two baseline approaches:

- Individual Hashes, where images are classified as similar or different considering only one hash algorithm.
- Majority Decision, where verdicts of all four hash algorithms are considered separately, and the majority verdict is taken as the final.

B) Performance Metrics

- **Accuracy** - overall effectiveness considering both positive and negative predictions
- **Precision** - how many predicted positives are actually positive
- **Recall** - how many actual positives were correctly identified
- **F1 Score** - provides a balanced view of precision and recall

5. Decision Tree Efficacy

5.1 Summary of Results

Approach	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
dHash	89.18	100.00	78.22	87.78
pHash	88.81	100.00	77.48	87.31
wHash	88.76	99.18	78.04	87.34
NMFHash	75.45	97.73	51.81	67.72
Majority Decision	89.18	100.00	78.22	87.78
Decision Tree	**95.12**	**99.80**	**90.36**	**94.84**

5.2 Observations

Using the best performing individual hash (dhash) as benchmark,

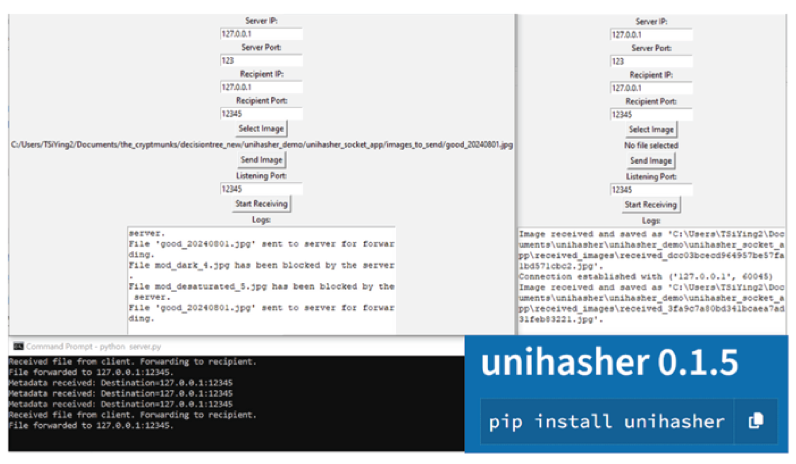
- Majority Decision approach has **limited ability** to improve performance beyond that of any single hash algorithm.
- The **Decision Tree**, using a combination of all four hash algorithms, shows a **significantly improved accuracy** and F1 Score.

5.3 Discussion

In particular, the **Decision Tree** produced significantly fewer false negatives, such that fewer visually similar images go undetected. For content moderation of images, this is desirable as it **prevents potential cascading effects** resulting from the spread of **harmful but unblocked** content.

We also tested our Decision Tree on a completely separate dataset of 10,000 images, with the **exact same decision nodes** as obtained in 4.3 above. Our Decision Tree approach achieved a **high accuracy of 91.94%** and **F1 Score of 91.42%**, showing its **high generalisability** - the same thresholds can be easily applied to different types of images, thus having **high potential** for application amongst wider contexts and more general use cases without requiring retraining of the tree.

6. Software Contributions



- **GitHub library, "unihasher"**, for developers to incorporate our decision tree solution into their applications.
- Proof-of-concept **chat application** demonstrating our library in action.

7. Future Work

- Extending to videos or animated images
- Exploring more different hash algorithms
- More complex decision tree with confidence score

8. References

[1] Buchner, J.. Image hash Python library. Available at: https://github.com/JohannesBuchner/imagehash?tab=BSD-2-Clause-1-ov-file#readme

[2] Z. Tang, X. Zhang and S. Zhang, "Robust Perceptual Image Hashing Based on Ring Partition and NMF," in IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 3, pp. 711-724, March 2014, doi: 10.1109/TKDE.2013.45.