ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ ПЕРЦЕПТИВНЫХ ХЕШ-ФУНКЦИЙ ДЛЯ ЗАЩИТЫ ИНФОРМАЦИИ

Захаров Денис Артурович

студент, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, РФ, г. Санкт-Петербург

Штеренберг Станислав Игоревич

научный руководитель, старший преподаватель, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, РФ, г. Санкт-Петербург

Аннотация. Сами по себе хеш-функции получили широкое распространение. Самый яркий и самый знакомый обывателю пример их применения - технология блокчейн. В её основе лежит использование криптографических хеш-функций, которые как известно, не могут вернуть два одинаковых хеша для разных наборов данных. Кроме того, эти функции обладают так называемым “эффектом лавины”, что означает, что любое, даже самое незначительное изменение исходных данных приведет к заметному изменению результата. Такой подход идеален для работы блокчейна, так как хеш-функции гарантируют «необратимость» всей цепочки транзакций. Однако, хеш-функции не передаются транзакции последовательно, а напротив - собираются в одно хеш-значение при помощи двоичного дерева с хешами (дерево Меркла). Другими словами, общее состояние блокчейна можно выразить одним числом - хешем последнего (нового) блока, причем число это будет уникальным, и изменение любого другого хеша в цепочке повлечет за собой и изменение итогового хеша.

Ключевые слова: Cryptography, Hash, Image Hash, Audio Hash, Video Hash, SHA-256

Рисунок 1 Схема блокчейна

Как мы выяснили, алгоритмы классических хеш-функций оптимально подходят для блокчейна и для решения ряда других задач (в файловых системах, в управлении ключами и т.п.). Тем не менее, свойство неизменности результата хеш-функции не дает использовать их в других задачах, где необходимость его применения кажется очевидной. О таких случаях и пойдет речь далее.

Введение:

Однако, для другой задачи, например, защите авторских прав обычные хеш-функции не подойдут из-за упомянутого ранее “эффекта лавины”. Часть графического, аудио, видео объекта не удастся сравнить с его полной версией! Почему? Ответ прост они могут возвращать совершенно различные значения для схожих объектов. А перцептивное хеширование - это частный случай локально-чувствительного хеширования. Поэтому перцептивные хеш-функции, как и локально-чувствительные, для близких 6 объектов возвращают близкие значения хешей. Перцептивные хеши можно сравнивать между собой и делать выводы о степени сходства двух наборов данных. Название «перцептивных» хеш-функций произошло благодаря тому, что в ходе вычисления значения хеша применяются процессы, имитирующие различные аспекты восприятия информации человеком. Они могут быть использованы для сравнения на схожесть некоторого числа объектов будь то аудио, видео или графика, что делает перцептивный-хеш must-have в сфере защиты информации. Рассмотрим схематично его работу в сравнении с обычным хешом, а также применение на упомянутых выше объектах.

Теория:

Перцептивный хеш - это “отпечаток пальца”, основанный на вводе изображения, который можно использовать для сравнения изображений путем вычисления расстояние Хемминга (что в основном означает подсчет количества различных отдельных битов). Анализируя принцип работы перцептивного хеша с обычным, получаем следующую схему:

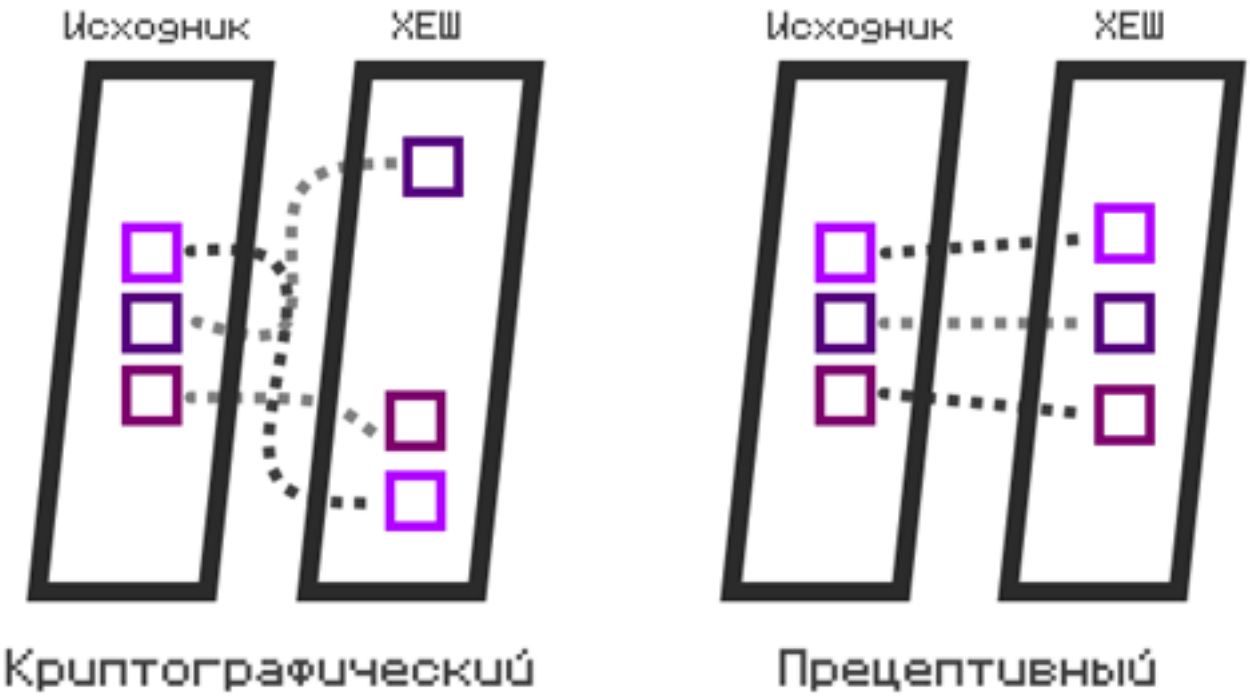

Рисунок 3. Условные схемы работы обоих алгоритмов

Области применения:

Как и полагается, все упирается в наши возможности и фантазии, однако мы выделили следующие три категории, в которых “подробно” рассказывается о возможности применения данной хеш-функции.

Графические объекты:

Существует несколько различных алгоритмов хеширования воспринимаемых изображений, но все они используют одинаковые шаги для создания упомянутого выше “отпечатка”. Самым простым для объяснения является средний хэш (также называемый хэш). Давайте возьмем следующее изображение и посмотрим, как оно работает.

Рисунок 4. Глад Валакас

Модификации объекта, которых можно выделить бесконечное множество различных способов издевательств над нашим исследуемым объектом, для простоты предлагаю выделить три категории:

Поворот изображения; Обрезка; Цветовая коррекция

Таблица 1.

Модификации изображения (Теория)

Модификация	**LV_1**	**LV_2**	**LV_3**	**Единицы**
Поворот	1-15	16-30	31-45	Граду
Размеры изображения	0	60	30	% тек изобра
Цветовая коррекция	Инверсия цвета	Шум	Шум+ Инверсия цвета	-

Показываем:

Таблица 2.

Модификации изображения (Практика)

Модификация	LV_1	LV_2	LV_3
Поворот			
Обрезка изображения			
Цветовая коррекция			

Существующие прецептивные-хеш функции:

- Difference Hash (об. diff_hash или ih_dhash)
- Average Hash (об. average_hash или ih_avghash)
- Perceptual (DCT) Hash (об. dct_hash или ih_phash)
- Wavelet Hash (об. wavelet_hash или ih_whash)

Таблица 3.

Сравнение прецептивных хеш алгоритмов

Модификация	LV_1 (PHash)	LV_2 (PHash)	LV_3 (PHash)	Единицы измерения
Поворот				Градусы (°)
Difference Hash	4e969b1f3d1b1e1b	3c07071b1b1b1f07	060806071f9e4810	Схожесть 35%
Average Hash	e3c3c3cf81cfc7cf	ffe3e1838383c7ff	ffffe3c3c3c7ffff	Схожесть 30.67%
Perceptual (DCT) Hash	b16565869e969b1a	bd67c6989a98c961	e56c9e92396d6494	Схожесть 8.335%
Wavelt Hash	e3c3c3c38583c7c3	ffe3c18181018 7f7	382041000143243c	Схожесть 19.996%
SHA-256	fccb291091d4c813b3a2b6c8fe7a6fe1962be13db4785ab05ebb051dfc3d87b2	036de85d1c0e3cefe3822ef1ebebce4daeb725f49d30b61693a3e36858e6e20e	a06d84bb03853253b1ed986add089e701385c25b66e7cb855637a3d4a1a51b14	Не поддаётся сравнению
Обрезка изображения	LV_1 (PHash)	LV_2 (PHash)	LV_3 (PHash)	% текущего изображения
Difference Hash	4f1f0f8bc7675b8f	464e9d0f0e9b8ec7	4e969b1f3d1b1e1b	Схожесть 50%
Average Hash	e3e7e7c363210103	e3e3c7c7e7c3e363	e3c3c3cf81cfc7cf	Схожесть 66,66%
Perceptual (DCT) Hash	a3e1274de06fe624	a1e744e5a6dc9a92	b16565869e969b1a	Схожесть 66,66%
Wavelt Hash	e3efe7c363210107	e3e3c1c7c3c34361	e3c3c3c38583c7c3	Схожесть 43,33%

SHA-256	eab2ec878e8bd461e6e4f0f3d8ad1b60800ec78c7baa41dbea0b3db0e4d3f23c	6de80bb07ce7c3eb8ea139140e24b4b286ac72e3ecd871c069ce47a273f14985	fccb291091d4c813b3a2b6c8fe7a6fe1962be13db4785ab05ebb051dfc3d87b2	Не поддаётся сравнению
Цветовая коррекция	LV_1 (PHash)	LV_2 (PHash)	LV_3 (PHash)	—
Difference Hash	316964e0c2e4e1e4	4e969b1f3d1b1e1b	b16964e0c2e4e1e4	Схожесть 29,33%
Average Hash	1c3c3c307e303830	e3c3c3cf81cfc7cf	1c3c3c307e303830	Схожесть 72,83%
Perceptual (DCT) Hash	ce9a9a79216964e5	b96465869e969b1a	c69a9a79616964e5	Схожесть 26,03%
Wavelt Hash	1c3c3c3c7a7c383c	e3c3c3c38583c7c3	1c3c3c3c7a7c383c	Схожесть 48,68%
SHA-256	67a6ff1067b82c27ad9a8bcb70efd36ba904c5b53a167400eb5ad8d395f18b16	8a60acb26fd1da6cdbfdea13e4b97297a84ac599fc3696ce23e6f18d604460e8	6555cce8b706dc6e476b08789d8062fa7f0f968017ed529350b4b14c8cf0f7e1	Не поддаётся сравнению

Таблица 4.

Сравнение прецептивных хеш-алгоритмов

Название алгоритма	EDM расстояние	Гистограмма IoU	Оптимальный F1-счет
Difference Hash	12.54	0.20	0.79
Average Hash	8.79	0.25	0.70
Perceptual (DCT) Hash	8.03	0.30	0.69
Wavelet Hash	12.23	0.21	0.77

Применение:

Защита от перерепостов, распознавание объектов по фотографии, одним словом всевозможная защита авторских прав.

Аудио объекты:

1. Х.Ф. На периодической основе

Мы считаем, что профиль периодичности аудио кадра может быть использован в качестве подписи для идентификации и контроля несанкционированного доступа. Свойство периодичности звуковых сигналов было использовано в таких приложениях, как обнаружение голосовой активности, обнаружение тишины и сжатие речи. Мы рассмотрели два различных метода оценки периодичности, один из которых основан на параметрической оценке, а другой - на методе основано на корреляции. Блок-схема общего хэша на основе периодичности на Рисунке 1. Входящий аудио объект обрабатывается кадр за кадром, и с одинарной периодичностью. значение извлекается для каждого кадра. Аудиосигнал препроцессируется для того, чтобы вывести на передний план периодическую динамику сигнал. В идеале, цель сглаживания в препроцессировании этап - спектральное усиление, т.е. удаление спектральных характеристик в связи с аудиоконтентом, при выходе из него спектральная тонкая структура (основная частота) нетронута. Фильтрация обратного линейного предсказания (LP) является распространенным способом выполняя это задание. Сначала применяется фильтр нижних частот, а затем инверсный фильтр прогнозирования с 4 ответвлениями. Этот сигнал обозначается как s0(i), i = 1, ... N. Затем аудиосигнал сегментируется на перекрывающиеся кадры, и каждый кадр заостряется в окне удара, чтобы уменьшить эффект края. Скорость кадров составляет 25 миллисекунд, а процент перекрытия - 50% (т.е. длина перекрытия составляет 12,5 миллисекунд), что достаточно для извлечения квазистационарных сегментов из звукового сигнала. Оценщик периода работает с каждым таким обработанным аудио кадром. Наконец, расчетные временные ряды покадровых интервалов подвергаются постобработке с помощью фильтра конечных импульсов с семью нажатиями для смягчения последствий искажений, которые могут привести к эффекту десинхронизации. Термин "десинхронизация" относится к тому факту, что при поиске аудио документа с коротким клипом (скажем, 5 секунд) начальная и конечная точки его хэша будут отображаться как произвольно расположенные. на всей хэш-фрагменте документа. Сглаживание смягчает этот резкий пуск и остановку гашиша. часть клипа.

Рисунок 5. Блок-схема

Несколько слов в порядке для выбранного диапазона звуковых частот. Известно, что типичная доминирующая частота человеческого голоса находится в диапазоне 50-400 Гц, в то время как она составляет может быть гораздо шире для музыкальных сигналов. Однако, несмотря на то, что частоты, присутствующие в музыке, могут быть гораздо шире. (около 50-4000 Гц), диапазон 50-400 Гц все еще охватывает большую часть музыкальных звуков.

2. Х.Ф. Трансформация Домена

Аудио сигнал делится на возможно перекрывающиеся кадры, и каждый кадр представлен его частотными кепстральными коэффициентами (ЧКК), которые представляют собой

краткосрочные спектральные характеристики. Разложение по сингулярным числам (РСЧ) дополнительно суммирует эти особенности. Обратите внимание, что в методе на основе РСЧ мы используем исходный сигнал, а не его версию с фильтром нижних частот, как в схемах на основе периодичности

Блок-схема вычислительной процедуры для Характеристики ЧКК показана ниже. Вычисляется дискретное преобразование Фурье (ДПФ) каждого оконного кадра, и логарифмические величины этих коэффициентов сохраняются.

Рисунок 6. Блок-схема ЧКК

Этот спектр затем делится на частоту распада расплавов. в соответствии с нелинейным восприятием слуховой системы человека, линейным ниже 1 кГц и логарифмическим выше. Спектральные компоненты расплава усреднены, чтобы получить гладкий спектр через фильтрацию расплава. У само-фильтров нелинейные и перекрывающиеся лаки расплава. Наконец-то Характеристики ЧКК получены путем косинусного преобразования (ДКТ) на расплавно-спектральных векторах. Точнее говоря, можно начать с вычисления N точек входного сигнала.

$$S(k) = \sum_{i=0}^{N-1} s(i)e^{-j2\pi ik/N}, \quad 0 \le k \le N-1$$

Формула 1

Также определяется фильтровбанк фиотров M, где треугольные фильтры, Hm(k); m = 1, 2, ... M; k = 0, 1, 1.... N - 1, имеют увеличенную полосу пропускания в соответствии со шкалой mel. Эти фильтры используются для вычисления среднего спектра вокруг каждой центральной частоты в (6). лог-спектр энергии на выходе каждого фильтра вычисляется как

$$\Psi(k) = \ln\left[\sum_{k=0}^{N-1} |S(k)|^2 H_m(k)\right], \quad 1 \le m \le M$$

Формула 2

Наконец, расплавно-частотный cepstrum вычисляется как ДКТ выходов фильтра *M*:

$$c(n) = \sum_{m=0}^{M-1} \Psi(m) \cos\left(\frac{\pi n(m-0.5)}{M}\right), \quad 1 \le n \le M$$

Формула 3

Количество ЧККК аспектов в «Формуле 3» обычно составляет от 24 до 40, хотя для речи часто используются первые 13 коэффициентов cepstrum.

В результате получается матрица $F \times M$, в которой каждая строка состоит из значений MFCC для одного кадра, и F строк, количество кадров, в которые был сегментирован весь аудиосигнал. Эта матрица отражает эволюцию сигнала во временно-частотном ландшафте. Краткое описание этой ситуации рассчитывается через ЧКК этой матрицы. Разложение сингулярного значения эффективно уменьшает $F \times M$-мерную M-матрицу ЧКК -характеристики на гораздо меньшую инвертируемую квадратную матрицу. Таким образом, данная $F \times M$ матрица разлагается как $A = UDV^T$, где A - $F \times M$ матрица, которую мы хотим обобщить, D - как $F \times M$ матрица с минимальными (F, M) диагональными элементами, U - как $F \times F$ ортогональная матрица, а V - как $M \times M$ ортогональная матрица. В целом, несколько сингулярных значений (первые несколько составляющих диагональной матрицы D) дают хорошее обобщение матрицы A. В нашем исследовании мы использовали только от одной до трех единичных ценностей.

Применение:

Такие перцептивные хэш-функции могут быть использованы в качестве инструмента для поиска конкретной записи в базе данных,

для проверки подлинности содержания записи, для мониторинга некоторой трансляции, для автоматического индексирования мультимедийных библиотек, для обнаружения атак фальсификации контента и т.д. Например, при поиске по базам данных и мониторинге вещания, вместо этого сравнения всего набора образцов, последовательность хэширования будет достаточно для быстрой идентификации контента. Более того, в контексте водяных знаков желательно внедрить следующее: В контексте водяных знаков желательно, чтобы в документ вставлялась зависящая от содержания подпись в сочетании с именем владельца или авторского права. Такие зависящие от содержимого водяные знаки помогают от копировальных атак, когда злоумышленник может попытаться обмануть систему, скопировав встроенный водяной знак из одного документа и перенести его на другой документ. Хэш-значения могут также использоваться для синхронизации в водяных знаках, где в качестве решения против атак по десинхронизации часто используется многократное встраивание.

Видео объекты:

Вытекает из графических объектов, по сути у нас появляется возможность использовать работу алгоритма в целях поиска объекта на изображении в действии. В этом случае потеря информации вокруг основной части изображения не играет особой роли. Развитие этой области прецептивного хеширования я вижу как некоторый механизм входящий в набор AAA протоколов, например способ действенный способ распознавания лиц и предметов, защита от ботов, отказ от капчи, работа с AR, да и в целом обучение в реальном времени.

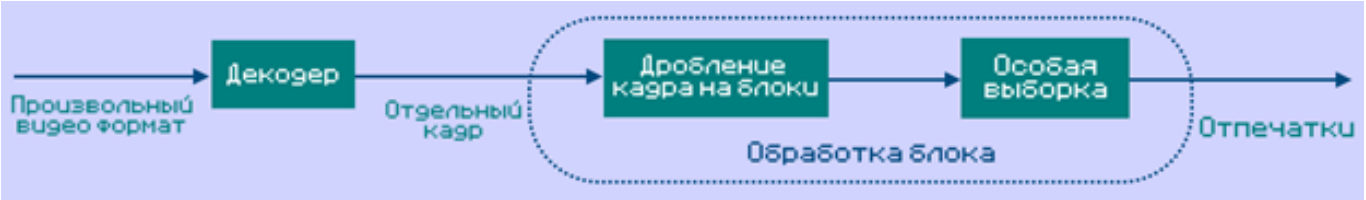

Рисунок 7. Схема обработки видео отпечатка

Заключение:

Исходя из общего положения и опираясь на совокупность всех ранее выше показанных и упомянутых операций можно заключить, что в результате грамотного и успешного изучения возможности использования перцептивных хеш-функций для защиты информации мы способны реализовать защиту Графических, Аудио и Видео объектов для следующих целей:

Защита от перерепостов; распознавание объектов по фотографии; Защита авторских прав, Защита от различного рода имитаций.

Список литературы:

3. Балакирский В.Б., Коржик В.И., Кушнир Д.В. Принципы квантовой криптографии // ВАК Защита информации. Конфидент. 1995. № 3-1. С. 43.

4. Никитин В.Н., Юркин Д.В. Сравнение стойкости реализаций протокола при выборе различных криптографических систем // ВАК Защита информации. Инсайд. 2008. № 6 (24). С. 68-72.

5. Никитин В.Н., Юркин Д.В. Анализ протоколов шифрования. // ВАК Журнал радиоэлектроники. 2009. № 4. С. 7.

6. Dr. Neal Krawetz Looks Like it // hackerfactor Защита информации. 2011-05-26

7. Dr. Neal Krawetz Kind of Like That // hackerfactor Защита информации. 2013-01-21

8. Dmitry Petrov Wavelet image hash in Python // FullStackML Защита информации. 2013-01-21