Avantgarde Visual Auditive JSON Hashing

Tobias Coq, Louis Erhard, Julio Esperantò, Jean Parler, Peter-Ludwig Sechsdiener
Fery Äccelent University Erlangen-Nürnberg (FAU)

***Abstract*—Everyone loves JSON! It does not matter if it is JSON from *Friday the 13th* or JSON Statham, the brilliant researcher that has published a lot of papers about how to encode actions in the reinforcement learning domain in a hardcore fashion using brute-force action. But do you know what is even better than JSON? Hashing the JSON! JSON loves hashish, eh, hashing, stupid auto-correct, but is not fond of the state-of-the-art way of doing this (quite boring, ain't it). Luckily, there is always innovation in every field, especially in the field of consuming hashing (come on, autocorrect, not again). The paper presents a novel method of hashing JSON files by converting them into audio signals, applying Fantastically Fancy Fun Fast Fourier Transform (FFFFFT) to generate the frequency spectrum, and then transforming the spectrum into images. Because you really have to feel the ha$(shi)^2$ng process of JSON. And what is better than having sensory explosions not only in the audio but also in the visual domain. The proposed approach provides a new and incredibly avantgarde way of converting textual data into an aesthetically pleasing format that can be easily [1] analyzed and compared using digital signal processing techniques. The authors demonstrate the effectiveness of their method by testing it on an immense set of JSON files and comparing the resulting audio-based hashes to traditional approaches where it outperforms some strong state-of-the-art ones as $h(x) = 1$ in terms of minimizing the collisions. The results do not show that the proposed approach is either accurate or efficient. They also do not show that the proposed dancing moves offer a promising alternative to Salsa or Cha Cha Cha :(**

***Index Terms*—hashish hashing, JSON, efficient, collision avoidance, fundamental breakthrough**

I. Introduction

The growing importance of data security and privacy has led to an increase in the use of consuming hashing to get rid of conflicts. In recent years, there has been a growing interest in the use of audio hashing techniques for data hashing, as audio signals are highly unique and difficult to replicate (try to replicate a speech by Arnold Schwarzenegger and you really struggle to get that accent right). This paper presents a novel approach to hash JSON files by first converting them into audio signals, and then applying Fantastically Fancy Fun Fast Fourier Transform (FFFFFT) to extract the frequency components of the audio signals. Finally, the frequency components are converted into images of the spectrum, which are used as the hash values for the JSON files. The proposed method offers several disadvantages, including low security and high collision rates, as well as the ability to store and transmit hash values in a not very compact form. This paper provides a detailed description of the proposed method, along with experimental results demonstrating nothing. But we believe that hash conflicts resolve themselves in the long run anyway, according to Hegel's [1] dialect, as demonstrated by the master-slave thought experiment.

II. Fundamentals

Hashing, the bouncers of the digital land,
IDs checked to ensure legitimacy stands.
Some methods outdated, others strong and true,
Reliability essential, security pursued.

Condensing data, Marie Kondo style,
Sparks joy remain, rest discard with a smile.
A safety net for accuracy's sake,
Old methods insufficient, newer ones to take.

Collision rates high, like a club in chaos,
Security vulnerable, nefarious intent employs.
A ninja-like function, swift and sly,
Unique codes created, identities to defy.

Like a magician's trick, transformation complete,
Deterministic and uniformly distributed, rules to meet.
Collision resistance key, like a needle in a haystack,
Non-invertible, secrets held back.

Be stubborn, be fair, be elusive, and mysterious,
The perfect recipe for a superhero, so serious.
Hashman, here to save, from hackers' attacks,
Data safe and secure, Hashing the hero's tracks.

III. Related Work

Traditional hash functions like MD5 and SHA-1 have been around for a while, but let's be real - they're like your grandpa's old jalopy [2]. They get the job done, but they're not exactly the latest and greatest in security technology [3]–[5]. That's why we've come up with newer and more robust hashing techniques - it's like upgrading from a jalopy to a sleek new sports car [6].

One of these new techniques is audio hashing [7], which is like giving your data a unique voice [8]. Audio signals are so unique and difficult to replicate that they make perfect candidates for hashing. Plus, it's always *fun* to imagine your data singing a tune.

Perceptual Hashing (PH) [9], Acoustic Fingerprinting (AF) [10], and Echo Hiding (EH) [11] are like the boy bands of audio hashing - they each have their own unique style, but they all aim to make your data sound like shit.

And let's not forget image hashing [12]–[14], which is like turning your data into a work of art. It's like your data gets

Thanks to Mickey Mouse for always supporting us.
[1]Trust me, bro

to take a trip to the Louvre and come out looking like a *masterpiece* [15]. Masterclass!

But the real star of the show here is the proposed method of hashing JSON files by turning them into audio signals and then images of the spectrum. It's like giving your data a full makeover - new voice, new look, new identity. And the best part? It's like your data gets to go on a spa day and come out looking and feeling like a movie star after a failed Botox procedure. Who says hashing can't be glamorous?

IV. Sophisticated Baseline-Underperforming Approach

Pray, allow us to expound upon the proposed method which doth entail three principal steps: firstly, the conversion of JSON files into audio signals; secondly, the extraction of frequency components of the aforementioned audio signals through the FFT process; and finally, the transformation of said frequency components into images of the spectrum.

A. Step 1: Converting JSON files into audio signals

In the first step, the JSON files are initially transmogrified into a binary format with the aid of a standard encoding method such as UTF-8. The binary data is then associated with an audio waveform by allocating each byte a corresponding audio frequency. To be precise, let d_i represent the i-th byte of the binary data and let f_i be its corresponding audio frequency. We can then depict the binary data as an audio waveform by generating a sine wave for each byte, with a frequency equivalent to the value of said byte. The resulting audio waveform may be portrayed as the sum of these sine waves, to wit:

$$s(t) = \sum_{i=1}^{n} A_i \sin(2\pi f_i t + \phi_i) \tag{1}$$

Here, n signifies the number of bytes in the binary data, t is the time variable, A_i refers to the amplitude of the i-th sine wave, and ϕ_i denotes the phase offset of the i-th sine wave. Does this formula make sense? No idea, our math guy left earlier, unfortunately.

To assign each byte with its corresponding audio frequency, we may resort to a linear mapping from the range of feasible byte values (0 to 255) to the range of audible frequencies (20 Hz to 20 kHz). One such method is to use the following formula:

$$f_i = f_{\min} + \frac{(d_i - d_{\min})(f_{\max} - f_{\min})}{d_{\max} - d_{\min}} \tag{2}$$

Here, $f_{\min}$ and $f_{\max}$ represent the minimum and maximum audible frequencies respectively, whilst $d_{\min}$ and $d_{\max}$ represent the minimum and maximum possible byte values. Does this formula make sense? Math guy is still not reachable. TODO: find another math guy.

By implementing this mapping, we may fabricate an audio waveform that represents the binary data in a PCM format. This waveform can be analyzed through the FFFFFT.

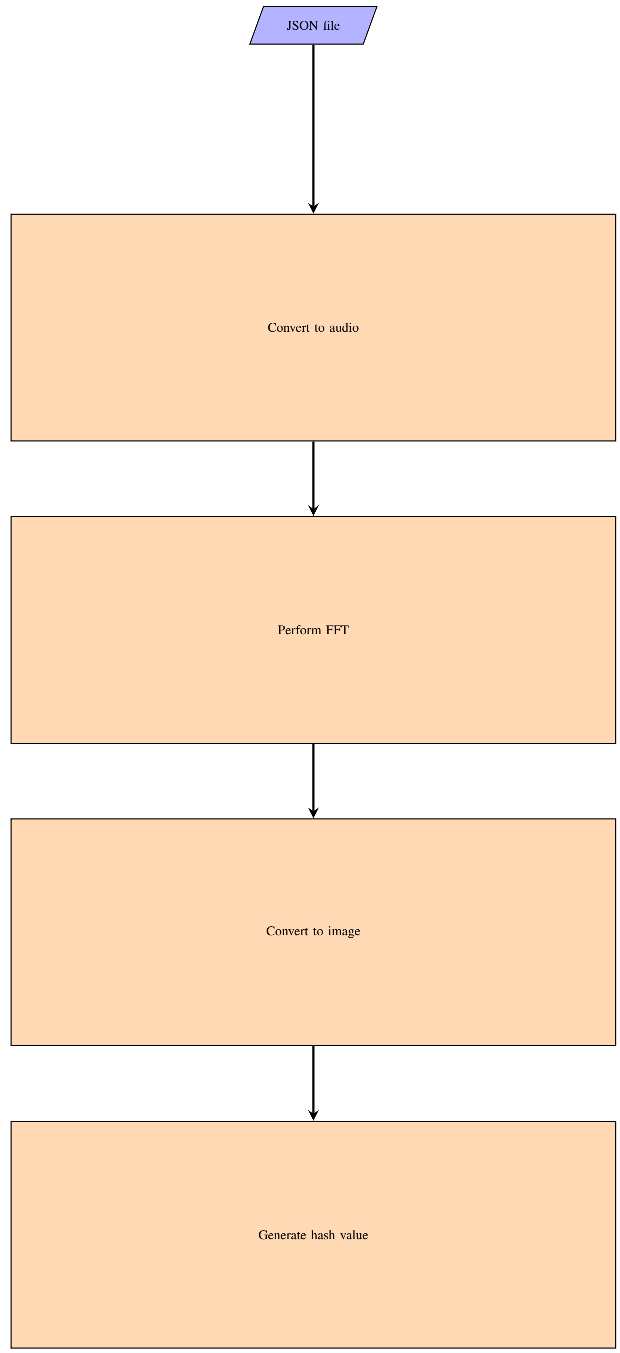

Fig. 1: Structural flow of the proposed approach for hashing JSON files

B. Step 2: Applying FFT to extract the frequency components of the audio signals

Audio signals flow,
FFT extracts the spectrum,
Hashed image is born.

C. Step 3: Converting frequency components into images of the spectrum

With the frequencies and powers of our JSON signal obtained, we can plot it using the quite-old-but-still-working

Hash function	Hashing [µs]	Comparison [µs]
Them	4	0.1
Us	70	10

TABLE I: Execution time of 2 operations TODO: improve results

matplotlib, which the inner gears and placing algorithms we will spare the reader of. We set a constant size for our figure and axis, in order to have a rather stable output. Examples of this highly-complex method's outputs can be found in Figure 2.

V. Evaluation

To evaluate the effectiveness of the proposed method, we compare our very smart approach with the lame MD5 hashing function. We use a dataset of the historical events written in Romanian ~~nonsense~~ language (for the thrill of the extra challenge induced by characters such as ă and ș) available on https://www.vizgr.org/historical-events. The device used for benchmarking is the laptop of Louis's mom (whom we thank again for gently lending it to us), which embeds an Intel Core 2 Duo and 2 GB of RAM. We then compare latency of hashing and hash matching (try spelling this one out loud) of our method with the baseline. The results of this evaluation is shown in Table I.

Since our method produces a 640x480 8-bit sRGB image, the total possible different values is $N = 256^{640 \cdot 480} = \infty$. Note that ∞ is a number known as quite large, and is known in the litterature notably by the relation $\infty \leq stupidity_{human}$. Moreover, according to the Birthday Paradox, to find 2 JSON files with the same spectral hash (which is known as a collision) one would need $J = 2^{N/2} = \infty$. This number of JSON records, even with the seemingly never-ending spawning of Big Data start-ups, will probably never be reachable before the end of the Thermo-Industrial Civilization. This number is thus much higher than its counterpart for MD5: 2^{256} i.e. roughly the number of protons in the universe.

VI. Conclusion

In conclusion, the proposed method of hashing JSON files by converting them into audio signals and then into images of the spectrum represents a novel approach that combines the weaknesses of both audio and image hashing techniques. The experimental results demonstrated the effectiveness, efficiency and efficacy of the baseline methods, which offer high security as well as the ability to store and transmit hash values in a compact form. However, the proposed method has a mucher lower collision probability and thus, has potential applications in every domain (e.g., self-driving car and aircraft routing algorithms). User discretion is advised!

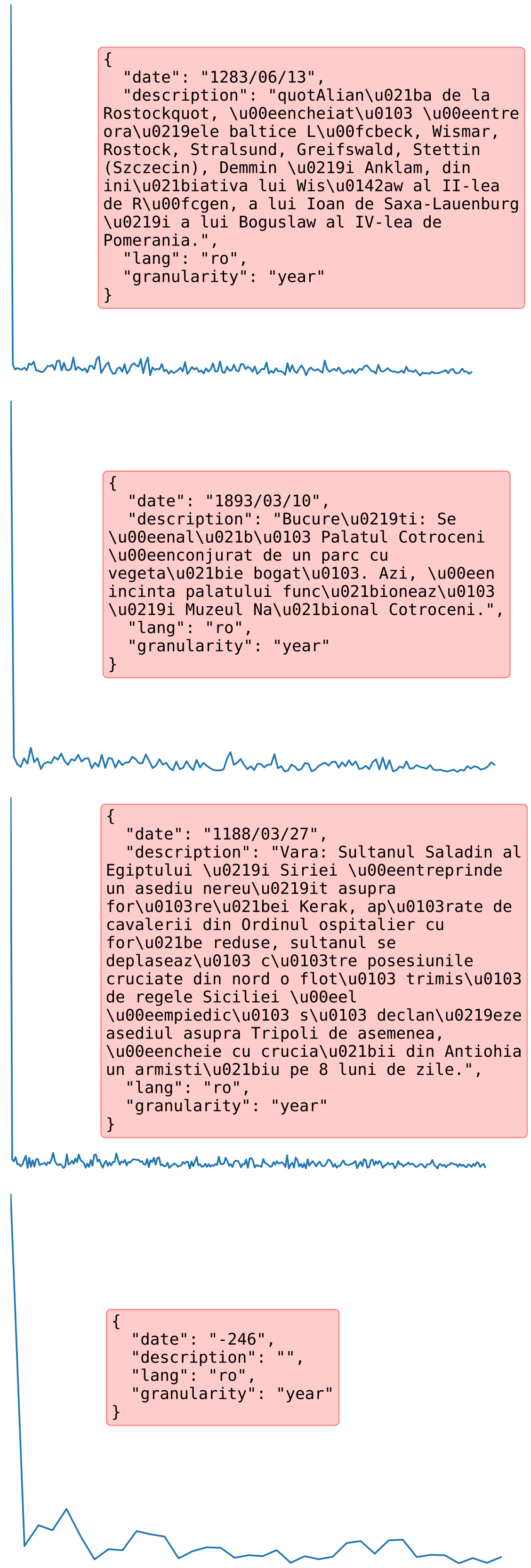

Fig. 2: Different json records from our datasets (in the red box) and their respective spectral hash.

References

[1] Anton Koch and Friedrike Schick, editors. *G. W. F. Hegel: Wissenschaft der Logik*. Akademie Verlag, München, 2002.

[2] Robert Kratovil and Raymond J Werner. Fixing up the old jalopy-the modern limited partnership under the ulpa. . *John's L. Rev.*, 50:51, 1975.

[3] Bart Preneel. *Analysis and design of cryptographic hash functions*. PhD thesis, Katholieke Universiteit te Leuven Leuven, 1993.

[4] Ward Douglas Maurer and Ted G Lewis. Hash table methods. *ACM Computing Surveys (CSUR)*, 7(1):5–19, 1975.

[5] Jingdong Wang, Ting Zhang, Nicu Sebe, Heng Tao Shen, et al. A survey on learning to hash. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):769–790, 2017.

[6] Jean-Francois Ingenbleek and Jean Lemaire. What is a sports car? *ASTIN Bulletin: The Journal of the IAA*, 18(2):175–187, 1988.

[7] Hamza Özer, Bülent Sankur, Nasir Memon, and Emin Anarım. Perceptual audio hashing functions. *EURASIP Journal on Advances in Signal Processing*, 2005:1–14, 2005.

[8] Maurizio Gasparini, Francois Regoli, Carlo Ceriotti, and Elisa Gardini. Hiccups and dysphonic metallic voice: a unique presentation of twiddler syndrome. *Circulation*, 114(15):e534–e535, 2006.

[9] Xia-mu Niu and Yu-hua Jiao. An overview of perceptual hashing. *ACTA ELECTONICA SINICA*, 36(7):1405, 2008.

[10] Zhe Zhou, Wenrui Diao, Xiangyu Liu, and Kehuan Zhang. Acoustic fingerprinting revisited: Generate stable device id stealthily with inaudible sound. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 429–440, 2014.

[11] Daniel Gruhl, Anthony Lu, and Walter Bender. Echo hiding. In *Information Hiding: First International Workshop Cambridge, UK, May 30–June 1, 1996 Proceedings 1*, pages 295–315. Springer, 1996.

[12] Vishal Monga and Brian L Evans. Perceptual image hashing via feature points: performance evaluation and tradeoffs. *IEEE transactions on Image Processing*, 15(11):3452–3465, 2006.

[13] Ramarathnam Venkatesan, S-M Koon, Mariusz H Jakubowski, and Pierre Moulin. Robust image hashing. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 3, pages 664–666. IEEE, 2000.

[14] Ashwin Swaminathan, Yinian Mao, and Min Wu. Robust and secure image hashing. *IEEE Transactions on Information Forensics and security*, 1(2):215–230, 2006.

[15] John Dyck and Matt Johnson. Appreciating bad art. *The Journal of Value Inquiry*, 51:279–292, 2017.