Comparing Randomness on Various Video and Audio Media File Types

Damir Omerasevic
PBH Technologies
PrinTec Group of Companies
Sarajevo
Bosnia and Herzegovina
Email: d.omerasevic@printec.ba

Narcis Behlilovic
Faculty of Electrical Engineering
University of Sarajevo
Sarajevo
Bosnia and Herzegovina
Email: nbehlilovic@etf.unsa.ba

Sasa Mrdovic
Faculty of Electrical Engineering
University of Sarajevo
Sarajevo
Bosnia and Herzegovina
Email: sasa.mrdovic@etf.unsa.ba

Asaf Sarajlic
BH Telecom
ID of Mobile Network
Sarajevo
Bosnia and Herzegovina
Email: asaf.sarajlic@bhtelecom.ba

***Abstract*—This paper analyses randomness in various video and audio media file types, like: Joint Photographic Experts Group (JPEG), Waveform Audio File Format (WAV), Flash video (FLV), high-quality, free and open video format for the web (WEBM) and MPEG-1 Audio Layer 3 (MP3). Analysis is done by executing different statistical test. Statistical tests are usually used for testing different both True Random Number Generators (TRNG) and Pseudo-Random Number Generators (PRNG), but we use them in this paper to analyse various media file types instead of TRNGs and PRNGs output results. Proposed methods for analysing are implemented in C programming language, by using one part of ENT (pseudo random number sequence test program) and making additional scripts for faster processing. Comparison of randomness is done by comparing different file types and accompanying results of statistical tests with each other. The results of comparison are presented.**

***Index Terms*—Randomness, Cryptography, Random Number Generators, TRNG, PRNG, Statistical Tests, Video Media Files, Audio Media Files.**

I. Introduction

Our idea in this paper is to recognize which media files have enough randomness quality to be used for extraction of cryptographic keys, in order to continue our research on open questions in CryptoStego - a novel approach for creating cryptographic keys and messages [1], and to be able to shorten total encryption time, by using already available/created media sources.

The paper is organized as follows. Related work is addressed in section 2. Section 3 explains our idea on how to measure randomness in different types of video and audio media files. Comparison results are presented in section 4. Conclusion and discussion as well as directions for future research work are in section 5.

II. Related work

Idea to use different media files to generate cryptographic keys is not new and has been around for a while. Most of proposed solutions were to generate personalized keys based on biometric features like fingerprint [2], voice [3] or face [4]. Good recent overview of biometric key generation methods and issues can be found in [5]. However, all of ideas mentioned here requiring certain processing time, which prolongs total encryption time.

Using entropy to measure randomness on series of data values is a well-accepted statistical practice in information theory [6].

A TRNG uses a non-deterministic or so called entropy source, together with a processing function (entropy distillation process) to produce randomness [7]. TRNGs have different source of input, which are in rule, bound to some physical phenomena and introduced into a computer, like atmospheric or natural phenomena (atmospheric noise, wind, etc.). It is not possible to determine any exact mathematical formula in order to define output results. Therefore, we have here so-called non-deterministic (or stochastic) processes, like one in [8].

On the other side, we have computer processing power which could be used to generate very good, almost similar (and almost random), output results, by using some kind of mathematical formulas, on which we base output results. We call these generators PRNGs. A very good overview of PRNGs is given in [9].

Quality of both TRNGs and PRNGs is possible to test. Output results from TRNGs and PRNGs are tested by statistical [7] and other kinds of tests on randomness. However, to the best of our knowledge, there were no attempts to analyse this problem from another angle, i.e., to test existing sets of already existing sources like various video and audio media file types on randomness, in order to avoid processing time for generating results from either TRNGs or especially PRNGs.

In the next section we will explain measurement methodology.

III. Measurement of randomness for video and audio media files

A brief explanation of basic idea is provided first. Then a more detailed explanation and results are given.

By using existing sets of already existing sources of media file types, which are good enough from randomness perspective to be used in everyday practice, we are shortening time for encryption (precisely, generation of cryptographic keys, which are based on some kind of randomness) and therefore making the whole encryption/decryption process faster.

A. Randomness Tests

In order to test which media file types are good enough from randomness perspective to be used in everyday practice, we were using different statistical tests [10] [7], namely the following:

1) Entropy Test
   Entropy originally was introduced in thermodynamics and Shannon applied it on digital communications [6]. Entropy is a measure of the uncertainty in a random variable in information theory, so we could interpret entropy as the measurement of randomness. Shannon was interested in determining what was theoretical maximum amount for file compression, i.e. more entropy means less compression (and better quality of randomness) and vice versa. We tested entropy as percentage, which means that results which are the closest to 100.0000% are the best.
2) Arithmetic Mean Test
   Arithmetic Mean Test is simply the result of summing all of bits in tested file and divide with the length of the file. If result is close to random, the result should be close to 0.5.
3) Serial Correlation Test
   Serial Correlation Test measures coefficient or extent to which each byte in tested file depends on the previous byte [10]. If result is close to random, the result should be close to 0.
4) Lempel-Ziv Compression Test [11]
   The purpose of the test is to determine if and how much of testing sequence can be compressed. The sequence is considered to be random if it can not be significantly compressed. If result is close to random, the result should be close to 0. Although this test has some weaknesses [12] [13], we consider it as good for testing, for the purpose of this paper.

B. Testing environment

Testing environment was set on laptop, with the following hardware: CPU Intel Core i7-3610QM, CPU working frequency 2.30GHz, and RAM memory 12 GB.

The laptop had the following software installed: operating system Windows 7 Professional Edition with SP1, and compiler Borland C++ version 5.02. As a source for our set of file types, we used the following sets:

- JPG set of files, where we used our own pictures taken by our camera,
- WAV set of files, where we used files from Windows operating system,
- FLV set of files, which we downloaded from YouTube web-site,
- WEBM set of files, which we downloaded from YouTube web-site, and
- MP3 set of files, which we downloaded from the internet.

Question of randomness of our selection of files might be raised. We made tests on more files for each file type and just selected ten files for each file types, as representative results. For example, we used for MP3 files 50 test files, from different sources, and presented results for ten of them only. Another issue is if the selected files are good representatives of their types. We justify our selection with the final purpose of our testing. We want to check how random are certain media files that are easily available to anyone at any time.

C. Testing procedure

We used compiler Borland C++ and adopted source code from [10] and we were making additional scripts for faster processing. Scripts are done in that way that we use [10] not only for one file, but for the whole folder, so we made efficiency and performance improvement for overall measurement process.

The measurement is done by running scripts, one time for each tested file type, and after that we collected results in one Excel table. We extracted all tables and comparisons which are presented in this paper from the Excel table.

We used file indexes instead of real file names, due to space reduction and better table data clarity. File size is given in bits, for calculating purposes.

D. Test Results

1) JPG Testing: Test results for JPG file types are given in Table I. As we could see from the results, JPG files have very good test results, for the purpose of this work, for all of four tests, as the following:

- File entropy expressed as a percentage varies from 99.8599 to 99.9999, which is very close to 100.0000,
- Arithmetic mean varies from 0.478 to 0.5087, which is close to 0.5,
- Serial correlation varies from -0.035412 to 0.020914, which is grouped around 0, and
- Reduction of compression is expressed as a percentage and is not varying, which means that is exactly equal to 0.

TABLE I
RESULTS OF COMPARISON FOR JPG FILES

File Index	File Size	Entropy (%)	Arithmetic Mean	Serial Correlation	Compression Reduction (%)
JPG1	21521176	99.9783	0.5087	0.011967	0
JPG2	362248	99.9870	0.4933	0.020914	0
JPG3	369864	99.9537	0.4873	0.02086	0
JPG4	383672	99.9822	0.4921	0.017137	0
JPG5	402448	99.9696	0.4897	0.009691	0
JPG6	715472	99.9917	0.4946	0.018443	0
JPG7	614040	99.8599	0.478	-0.017773	0
JPG8	3396208	99.9999	0.4994	0.01005	0
JPG9	2287168	99.9676	0.4894	-0.035412	0
JPG10	12917424	99.9938	0.4954	-0.014707	0

2) WAV Testing: Test results for WAV file types are given in Table II.

- File entropy expressed as a percentage varies from 79.3055 to 99.9210, which is not close to 100.0000,
- Arithmetic mean varies from 0.3149 to 0.4835, which is not close to 0.5,
- Serial correlation varies from 0.121548 to 0.790026, and is not grouped around 0, and
- Compression reduction expressed as a percentages varies from 0 to 20.

TABLE II
RESULTS OF COMPARISON FOR WAV FILES

File Index	File Size	Entropy (%)	Arithmetic Mean	Serial Correlation	Compression Reduction (%)
WAV1	272128	89.8740	0.3149	0.3149	10
WAV2	44512	90.3430	0.3191	0.379226	9
WAV3	3718752	95.4437	0.375	0.121548	4
WAV4	246528	93.0312	0.3459	0.609032	6
WAV5	0.3358	92.0769	0.3358	0.540125	7
WAV6	526160	90.0708	0.3167	0.546205	9
WAV7	438864	79.3055	0.2388	0.790026	20
WAV8	235536	97.0255	0.3988	0.451115	2
WAV9	315056	99.9210	0.4835	0.245809	0
WAV10	1521664	99.6657	0.466	0.39954	0

3) FLV Testing: Test results for FLV file types are given in Table III. As we could see from the results, FLV files have very good test results, for the purpose of this work, for all of four tests, as the following:

- File entropy expressed as a percentage varies from 99.9531 to 100.0000, which is very close to 100.0000,
- Arithmetic mean varies from 0.4939 to 0.50004, which is very close to 0.5,
- Serial correlation varies from 0.001153 to 0.019413, which is very close to 0, and
- Reduction of compression is expressed as a percentage and is not varying, which means that is exactly equal to 0.

Results for FLV file type are the best, comparing with all other file types tested in this paper.

4) WEBM Testing: Test results for WEBM file types are given in Table IV. As we could see from the results, WEBM files have very good test results, for the purpose of this work, for all of four tests, as the following:

- File entropy expressed as a percentage varies from 99.9295 to 99.9998, which is very close to 100.0000,
- Arithmetic mean varies from 0.4844 to 0.4993, which is very close to 0.5,
- Serial correlation varies from 0.003173 to 0.014056, which is very close to 0, and
- Reduction of compression is expressed as a percentage and is not varying, which means that is exactly equal to 0.

TABLE III
RESULTS OF COMPARISON FOR FLV FILES

File Index	File Size	Entropy (%)	Arithmetic Mean	Serial Correlation	Compression Reduction (%)
FLV1	149177272	99.9531	0.4872	0.019413	0
FLV2	59895888	99.9994	0.4985	0.005974	0
FLV3	158340880	99.9984	0.4977	0.00546	0
FLV4	700971952	100.0000	0.5004	0.001153	0
FLV5	33027968	99.9891	0.4939	0.012764	0
FLV6	361880112	99.9993	0.4985	0.003405	0
FLV7	460491968	99.9983	0.4975	0.00545	0
FLV8	309196744	100.0000	0.4999	0.002019	0
FLV9	58047696	99.9982	0.4975	0.005092	0
FLV10	156009472	99.9983	0.4975	0.00545	0

TABLE IV
RESULTS OF COMPARISON FOR WEBM FILES

File Index	File Size	Entropy (%)	Arithmetic Mean	Serial Correlation	Compression Reduction (%)
WEBM1	113135048	99.9295	0.4844	0.020691	0
WEBM2	1244183048	99.9998	0.4993	0.003173	0
WEBM3	311626752	99.9834	0.4924	0.005272	0
WEBM4	101210448	99.9960	0.4963	0.014056	0
WEBM5	365222584	99.9995	0.4986	0.011864	0
WEBM6	320629952	99.9969	0.4967	0.009266	0
WEBM7	228198976	99.9995	0.4987	0.006817	0
WEBM8	219042400	99.9909	0.4944	0.012517	0
WEBM9	314455432	99.9862	0.4931	0.006153	0
WEBM10	601979712	99.9976	0.4971	0.009781	0

5) MP3 Testing: Test results for MP3 file types are given in Table V. As we could see from the results, MP3 files do not have good test results for two of four tests, for the purpose of this work, as the following:

- File entropy expressed as a percentage varies from 97.2269 to 99.9950, which is close to 100.0000,
- Arithmetic mean varies from 0.4023 to 0.4959, which is not very close to 0.5,
- Serial correlation varies from -0.007505 to 0.044548, and is grouped around 0, and
- Compression reduction expressed as a percentages varies from 0 to 2.

IV. CONCLUSION

Easily available sets of already existing sources of media file types were tested for randomness. Files with content that is random could be source for short lived cryptographic keys. Otherwise key generation could take time. Using such files could make the whole encryption/decryption process faster.

Randomness measuring was performed using different statistical tests. Testing showed that FLV set of files, compared with all other above mentioned audio and video files, have the best results for all given statistical tests, and have not perfect, but very good randomness. This level of randomness is good

TABLE V
RESULTS OF COMPARISON FOR MP3 FILES

File Index	File Size	Entropy (%)	Arithmetic Mean	Serial Correlation	Compression Reduction (%)
MP31	69297632	99.3450	0.4524	0.030562	0
MP32	72783408	97.2269	0.4023	0.024078	2
MP33	71262040	99.2258	0.4482	0.024252	0
MP34	26025480	99.9950	0.4959	0.013389	0
MP35	73769792	99.4692	0.4571	0.005453	0
MP36	77397680	99.3962	0.4543	0.012867	0
MP37	78634840	98.9301	0.4392	0.044548	1
MP38	85857176	99.1732	0.4465	0.026074	0
MP39	56182072	99.4412	0.456	0.014138	0
MP310	54027072	99.6054	0.463	-0.007505	0

enough for short lived cryptographic keys, like session keys, or one-time keys.

Very close results to FLV set of files are results extracted from WBEM set of files. JPG set of files showed close results to WBEM.

WAV and MP3 set of files did not show results which could be accepted as good enough to be used as key generators.

It further means that the best option from all of tested media file types is to use YouTube web-site as a source for files which could be used as key generators, especially for [1], where we used one-time keys. JPG files could also be used, because of very good results.

Our future work is oriented towards defining of protocols for agreement on both sets and ordering of files, which are needed for generation of cryptographic keys.

REFERENCES

[1] D. Omerasevic, N. Behlilovic, and S. Mrdovic, "Cryptostego - a novel approach for creating cryptographic keys and messages," in *Systems, Signals and Image Processing (IWSSIP), 2013 20th International Conference on*, 2013, pp. 83–86.

[2] C. Soutar and G. Tomko, "Secure private key generation using a fingerprint," in *Cardtech/Securetech Conference Proceedings*, vol. 1, 1996, pp. 245–252.

[3] F. Monrose, M. Reiter, Q. Li, and S. Wetzel, "Cryptographic key generation from voice," in *Security and Privacy, 2001. S P 2001. Proceedings. 2001 IEEE Symposium on*, 2001, pp. 202–213.

[4] A. B. Teoh, D. C. Ngo, and A. Goh, "Personalised cryptographic key generation based on facehashing," *Computers & Security*, vol. 23, no. 7, pp. 606 – 614, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404804001701

[5] L. Ballard, S. Kamara, and M. K. Reiter, "The practical subtleties of biometric key generation," in *17th USENIX Security Symposium*, 2008.

[6] C. E. Shannon and W. Weaver, *A Mathematical Theory of Communication*. Champaign, IL, USA: University of Illinois Press, 1963.

[7] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic application," http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA393366, pp. 2–3, 2001, [Online; accessed 29.7.2013.].

[8] M. Ben-Romdhane, T. Graba, and J.-L. Danger, "Stochastic model of a metastability-based true random number generator," in *Trust and Trustworthy Computing*, ser. Lecture Notes in Computer Science, M. Huth, N. Asokan, S. apkun, I. Flechais, and L. Coles-Kemp, Eds. Springer Berlin Heidelberg, 2013, vol. 7904, pp. 92–105. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38908-5_7

[9] P. Hellekalek, "The theory behind pseudorandom number generators," http://statmath.wu.ac.at/prng/doc/prng.html#Theory, 2013, [Online; accessed 28.7.2013.].

[10] J. Walker, "Ent - a pseudorandom number sequence test program," http://www.fourmilab.ch/random/, 2008, [Online; accessed 13.9.2013.].

[11] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *Information Theory, IEEE Transactions on*, vol. 23, no. 3, pp. 337–343, 1977.

[12] P. Ebermann, "Why did nist remove the lempel-ziv compression test from the statistical test suite?" http://crypto.stackexchange.com/questions/129/why-did-nist-remove-the-lempel-ziv-compression-test-from-the-statistical-test-su, 2011, [Online; accessed 13.9.2013.].

[13] A. Doanaksoy and F. Glolu, "On lempel-ziv complexity of sequences," in *Sequences and Their Applications SETA 2006*, ser. Lecture Notes in Computer Science, G. Gong, T. Helleseth, H.-Y. Song, and K. Yang, Eds. Springer Berlin Heidelberg, 2006, vol. 4086, pp. 180–189. [Online]. Available: http://dx.doi.org/10.1007/11863854_15