2017-02-19

On the Development of a One-Time Pad Generator for Personalising Cloud Security

Paul Tobin
Technological University Dublin, paul.tobin@tudublin.ie

Lee Tobin
University College Dublin, lee.tobin@ucdconnect.ie

Michael McKeever
Technological University Dublin, mick.mckeever@tudublin.ie

See next page for additional authors

Recommended Citation

Tobin, P., Tobin, L., McKeever, M and Blackledge, J. (2017) On the Development of a One-Time Pad Generator for Personalising Cloud Security. *Eighth International Conference on Cloud Computing, GRIDs, and Virtualization, CLOUD COMPUTING 2017 February 19 - 23, 2017 - Athens, Greece.*

On the Development of a One-Time Pad Generator for Personalising Cloud Security

Paul Tobin*, Lee Tobin†, Michael McKeever‡, and Jonathan Blackledge§
* School of Electrical and Electronic Engineering
Dublin Institute of Technology, Dublin 2, Ireland
Email: paul.tobin@dit.ie
† CASL Institute Level 3, UCD Science Centre East
University College, Belfield, Dublin 4, Ireland,
Email: lee.tobin@ucdconnect.ie
‡ School of Electrical and Electronic Engineering
Dublin Institute of Technology, Dublin 2, Ireland
Email: mick.mckeever@dit.ie
§ Military Technological College
Sultanate of Oman,
Email: Jonathan.blackledge59@gmail.com

***Abstract*—Cloud computing security issues are being reported in newspapers, television, and on the Internet, on a daily basis. Furthermore, in 2013, Edward Snowden alleged backdoors were placed in a number of encryption systems by the National Security Agency causing confidence in public encryption to drop even further. Our solution allows the end-user to add a layer of unbreakable security by encrypting the data locally with a random number generator prior to uploading data to the Cloud. The prototype one-time pad generator is impervious to cryptanalysis because it generates unbreakable random binary sequences from chaos sources initiated from a natural noise. Specialised one-to-Cloud applications for this device means key distribution problems do not exist, even when used at different locations. A JavaScript application maximised the encryptor key entropy using a von Neumann algorithm and modulo-two arithmetic, where the key passed the National Institute of Standards and Technology statistical suite of tests. It is hoped that the final size of the generator should be similar to a typical Universal Serial Bus device.**

Keywords–Cloud security, Snowden, backdoors, one-time pad, chaos, noise, entropy, von Neumann.

I. Introduction

To address the problems of poor security on the Cloud, a prototype random number generator was created to encode data locally before being stored on the Cloud. Traffic on the Cloud Infrastructure as a Service (IaaS) is forecast to increase by twenty percent by 2019 [1], but security issues are affecting public confidence in this service. Breaches in security are rarely discovered instantly [2] and up to six months may elapse before being reported. The elapsed time between discovering security breaches has been reduced [3], but to satisfy the European Union (EU) General Data Protection Regulations (GDPR) coming into law in May 2018, mandatory breach notification must be reported by companies within 72 hours. Heavy fines of up to 4 percent of the annual turnover of a company will be imposed if they fail to report within this time [4].

Hacking on servers is reported almost daily in newspapers, TV and online [5], a problem compounded by the alleged presence of backdoors in public encryption. Microsoft employees, Dan Shumow and Niels Ferguson gave a presentation in 2007 and hinted at the possibility of a backdoor in a random number generator: On the Possibility of a backdoor in the National Institute of Standards and Technology (NIST) SP800-90 Dual Elliptic Curve Pseudo Random Number Generators [6] [7]. Interestingly, in April 2014, NIST dropped the Dual EC PRNG from their standards [8]

The New York Times in 2013 linked this presentation to documents leaked by Edward Snowden [9], where he alleged backdoors were placed by the National Security Agency (NSA) in certain public encryption systems [10]. Hence, nobody knows for sure what weaknesses exist in public encryption but probably the Advanced Encryption Standard (AES) algorithm is secure and has no backdoors. That said, an encryption system whereby the client can encode his data locally before being stored in the Cloud, is a solution to certain Cloud security problems. Our prototype provides a layer of security using the unbreakable One-Time Pad (OTP) random binary number stream generated from two chaos generators. This is not a novel idea but how the sources are connected and initialised using a cosmic noise source, is.

A. One-time pad history

Figure 1 shows the SIGSALY encryption system developed by A. B. Clarke and Alan Turing in Bell Labs during WWII [11]. SIGSALY weighed 55-tonnes and had a key distribution problem requiring a key the same length as the plaintext. Nevertheless, it produced unbreakable OTP encryption ciphers for encrypting transatlantic conversations between Churchill and Roosevelt. Similar OTP systems were used between Russian and American governments in the 60's, for securing the famous "hotline". Our prototype random number generator should be no bigger than a typical Universal Serial Bus (USB) device and with no key distribution problems because it stays with the client who may use it at different locations.

B. Paper Organisation

Section I explains why local encryption is necessary and explains how the OTP was successfully used during WWII and again in the sixties, to give unbreakable security protecting conversations between heads of state.

Figure 1. The 55-ton SIGSALY encoding system.

Section II outlines the structure of the OTP encoder and discusses the nature of a random generator initialised using cosmic natural noise and how it may be classified as a true random binary number generator. A medical application example for protecting patient confidentiality is given in Section III. Section IV explains the prototype design and discusses how OTPs were generated from chaotic analogue oscillators. In Section V, we discuss the JavaScript application for maximising OTP entropy and show how it interfaces with the data. NIST randomness p-test results for simulation and prototype circuits are discussed in Section VI, and the conclusion stated in Section VII.

II. The OTP prototype

Figure 2 outlines the system for generating OTP random bit streams for encoding data locally prior to uploading to the Cloud.

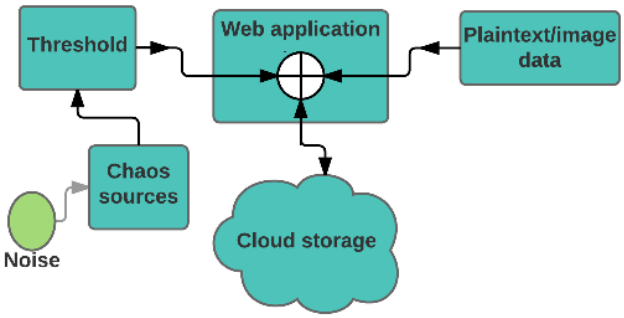

Figure 2. Prototype OTP generation.

Chaotic oscillators generated on a computer produce random binary sequences that have finite sequence lengths and hence are not truly random. This is due to the finite state of computer arithmetic [12] [13] and produce cryptographically poor ciphers. Random sequences generated from chaotic maps implemented on computers, similarly, have repeatable sequence lengths and also produce weak keys [14] [15] [16]. However, random binary sequences from analogue chaos circuits initiated from natural noise, have an infinite number of states and so produce random binary streams which have, in theory, infinite sequence lengths and generate excellent ciphers. The prototype can produce unlimited amounts of unbreakable OTP ciphers from deterministic chaos sources initialised with noise from a Frequency Modulation (FM) receiver and qualifies the prototype as a truly random source, rather than a pseudo random source [17] [18] [19]. Initial Conditions (IC) for each generator are applied to each chaos source, but for simulation only, the IC noise was provided from a random noise generator in PSpice called ***RND***. In [20], we explained how the OTP was exported from the simulator circuit and stored to a text file using PSpice ***VECTOR1*** parts and processed in the JavaScript application. However, a different technique must be used for the prototype and is considered in the following section.

A. Storing the OTP in an Arduino Shield

The OTP stream from the prototype was stored in an Arduino memory shield attached to the main Arduino board. An exclusive OR gate connected across two monostables created a clock stream from the two gate inputs and was used for writing 'ones' to the shield. Effectively, the gate removed the random temporal element from the bit stream (but not the randomness). The complete prototype is undergoing tests at present, but initial tests show it is producing cryptographically-strong encryptors. Only one chaotic oscillator was examined in this paper; the other chaos source is a novel implementation of the Chua oscillator [21].

III. A medical application for the Prototype

There are many potential applications for the prototype generator and here a medical application explains how patient information displayed on medical images, is protected. The following scenario is a patient who has persistent headaches and high blood pressure and is recommended by the doctor to have a Magnetic Resonance Imaging (MRI) scan at the nearest hospital. Such scans are produced using the international standard for storing, distributing and processing medical images and is referred to as the Digital Imaging and Communications in Medicine (DICOM) format [23]. However, these images show patient private information around the peripheral of the image [22] and must be protected from unwanted interception, otherwise patient confidentiality is compromised.

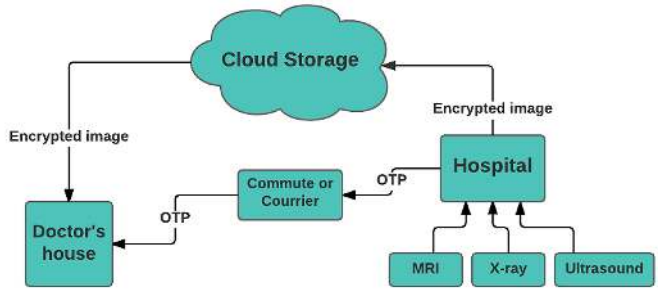

Figure 3. Encoding medical images.

At present, the procedure for sending MRI scans to the doctor's office is not secure. After scanning the patient, the hospital sends the MRI images containing patient personal information by post. Alternatively, they give them directly to the patient to bring to the doctor. Both methods have security weaknesses as the images could be lost in transit. Our solution involves hospital staff encoding the scanned images using the OTP generator and uploading the encoded images to the Cloud directory assigned to that doctor. The hospital staff then saves the encoding OTP to a memory device and gives it to the

patient who then gives it to the doctor to decode the scans at his office. A similar legal application concerns the legal profession operating between office and court. Here, data is encoded locally before uploading to the Cloud and the OTP replaces all those bulky folders carried previously. There are many such applications where people operating between two locations could use the encoder system to prevent sensitive information being lost in transit.

IV. The Lorenz Chaotic Analogue oscillator

Claude Shannon's 1949 paper [24] outlined presciently how digital chaotic maps could encrypt data using symmetric key encryption. Since then chaos cryptography has grown considerably, and from 2000, many chaotic maps were used in multi-algorithmic systems for encrypting data on a randomised block-by-block basis [25]. Our prototype uses Lorenz and Chua chaotic analogue chaos oscillators to create cryptographically-strong encryptors because of their ergodic properties. Edward Lorenz, a meteorologist, modelled weather patterns in the sixties and discovered chaos theory and Sensitivity to Initial Conditions (SIC), one of the hallmarks of chaos systems, when he truncated places of decimal from five down to three in his model after one run and it produced different results. The following first-order coupled equations appeared in his 1963 [26] paper (largely ignored at the time):

$$\begin{aligned} x &= -P\int_{t_0}^{t}\{x-y\}dt \\ y &= -\int_{t_0}^{t}\{-Rx+y+10xz\}dt \\ z &= -\int_{t_0}^{t}\{Bz-10xy\}dt \end{aligned} \tag{1}$$

The equations in integral form allow for electronic integrator implementation and also include a scaling factor of ten to reduce signal amplitudes for electronic devices. The parameters Lorenz used were: B = 2.666, P = 10, R = 28, but these were changed to: B = 2.8, P = 11, and R = 27.5 to maximise the cryptographic strength or entropy, of the OTP.

A. Thresholding the chaos signal

A random binary OTP stream was produced by thresholding the x signal at two voltages corresponding to the values at the centres of the *(x-y)* attractor shown in Figure 4 (b). These centres are the Fixed Points (FP) of (1), where one centre could represent a '1' when in that region, and when the other centre is visited, a '0' is created. The FPs are determined by assuming the system is approximately linear at the origin, i.e., (x = y = z = 0), so the coordinates at each lobe centre are calculated as follows:

$$\frac{dx}{dt} = 10(y-x) = 0 \Rightarrow x = y \tag{2}$$

Hence, we may write:

$$\frac{dy}{dt} = Rx - x - xz = 28x - x - xz = 0 \tag{3}$$

Substituting z = 27, yields:

$$\frac{dz}{dt} = x^2 - Bz = 0 \Rightarrow = \pm\sqrt{B(R-1)} \tag{4}$$

The FPs are the coordinates of the lobe centres $C_{1,2}$, given by:

$$C_{1,2} = \{+\sqrt{B(R-1)}, -\sqrt{B(R-1)}, (R-1)\} \tag{5}$$

Substituting the Lorenz parameter values gives the locii of the attractor as:

$$C_{1,2} = \{+8.48V, -8.48V, 27V\} \tag{6}$$

Magnitude scaling by 10 yields FPs equal to ± 0.8485 V at 2.7 V. Adding a bias shifting voltage to the bipolar x-signal makes it polar in form and gives threshold levels of 3.15 V and 4.84 V. The upper and lower threshold voltages are superimposed on the biased x-signal as shown in Figure 4 (a), and the out-of-phase set and reset sequences from each comparator were converted to constant widths by two 74121 monostables and superimposed on the Lorenz strange attractor as shown in Figure 4 (b).

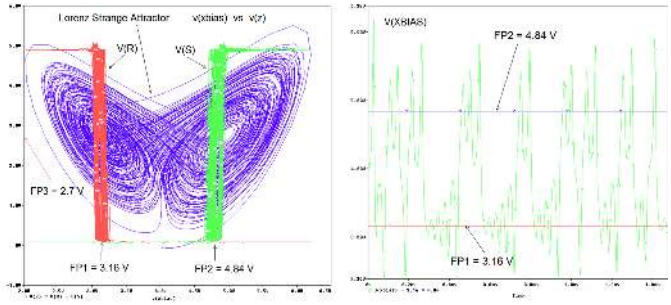

Figure 4. (a) FP thresholds (b) Butterfly attractor.

The threshold components were calculated by assuming a total potentiometer of 1 MΩ and Vref = 1.24 V:

$$Vhigh = 4.84\ V = Vref\frac{R10+R11+R12}{R12} \tag{7}$$

Similarly for R11,

$$Vlow = 3.15\ V = Vref\frac{R10+R11+R12}{R11+R12} \tag{8}$$

Figure 5 is the Lorenz chaotic oscillator circuit to realise (1). The circuit was simulated using the latest v 17.2 Cadence® OrCAD PSpice V17.2 using Analogue Behavioural Model (ABM) parts to achieve multiplication and integration but subsequently were replaced with actual model parts [27] [28]. The four-quadrant AD633 device modelled the cross-product nonlinear terms, xy and xz, terms necessary for chaos production and the TL084 quad operational amplifier integrated circuit solved the equation using a summing inverting integrator configuration.

The set and reset pulses from the monostables were connected to a 7486 exclusive OR gate (XOR) which outputs a clock stream for controlling when the OTP ones and zeroes are written to the Arduino shield attached to the main Arduino. The 'ones' are written to the shield from the monostable reset output, and the clock signal determines when the 'zeroes' are written. This is a different procedure to that used for storing the OTP during simulation [20], where the OTP was written to a text file using vector parts and processed in a JavaScript application.

Chaos oscillator initial conditions were obtained from a detuned 433 MHz FM receiver integrated circuit.

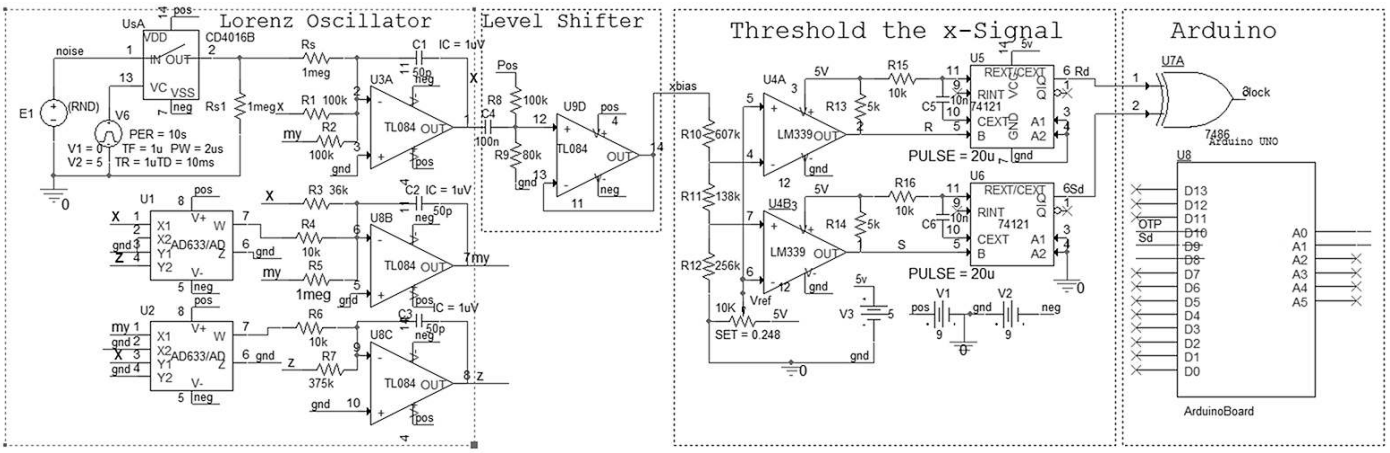

Figure 5. Generating the OTP.

The output level of the natural noise is random and ensures the chaos sources produce a random output that cannot be reproduced by an unwanted third party. The oscillator and threshold components are: $R1 = R2 = 100$ kΩ, $R3 = 36.3$ kΩ, $R4 = 10$ kΩ, $R5 = 1$ MΩ, $R6 = 10$ kΩ, $R7 = 357$ kΩ, and $C = 50$ pF. The potential divider components are: $R8$ and $R9$, bias the x signal by 4 V, $R10 = 607$ kΩ, $R11 = 138$ kΩ and $R12 = 256$ kΩ.

V. JavaScript interface application

The original MRI scan in Figure 6 shows where the patient information was located but removed for obvious reasons. The JavaScript application performs modulo two arithmetic between the OTP from the Arduino shield and the pixel array data from the bitmap medical image. In this example, the encoded image displays horizontal lines (see Figure 6), which would makes the encoded image susceptible to cryptanalysis because it now contains a bias and should be avoided at all costs. However, the bias was deliberately introduced by making the OTP purposely short because the application code repeated some of the random streams to make the OTP the same length as the image. In the middle pane, we observe no bias lines, even with no von Neumann correction applied. The OTP from the actual prototype should always be the same length as the plaintext, otherwise, the encryptor is weak.

The interface also applies the von Neumann (vN) algorithm to deskew, or unbias, the generated OTP bit stream. Whenever a '00' and '11' dibit pair occurs in the stream, they are rejected. Dibit '01' is converted to 0 and '10' to 1 [29]. However, the algorithm is inefficient because 75 percent of the data is lost. Another important requirement, often not applied when using this algorithm, is that the dibit streams should be from two uncorrelated chaos data streams. In the prototype, this alternate bit independence is achieved by using two independent chaotic data streams.

VI. Testing the One-Time-Pad

To resist cryptanalysis and to ensure an encryptor is truly random for correct certification, we considered the following tests:

- The autocorrelation test should display a single Kronecker delta auto-correlation function,
- The Power Spectral Density (PSD) should be uniform,
- The OTP must have maximum entropy by operating the chaos sources in a chaotic region to produce positive Lyapunov Exponents (LE) [30] [31].

Shannon entropy measures randomness but essentially is the Kolmogorov Complexity (KC), created simultaneously by Andrey Kolmogorov and Ray Solmonoff and specifies the minimum length to which a string of binary digits may be compressed (a truly random sequence is incompressible) [32]. According to Brudno's theorem and for certain phase space conditions, the Kolmogorov-Sinai Entropy (KSE) is the Algorithmic Complexity (AC) for all trajectories [33].

However, the cryptographic strength of random number sequences was also tested using the NIST suite of tests (revised in 2010). There are other test suites such as the ENT, TestU01, CryptX, Diehard, but the NIST suite of tests is universally accepted as the most comprehensive [34]. The NIST suite contains non-parameter tests for short OTP sequences and parameter tests for several million bit sequences. Table I shows the NIST results from simulation and prototype circuits, and from true noise sequences downloaded from [35].

VII. Conclusion

Poor Cloud security was addressed and we proposed a solution for a system which created an extra layer of security using a OTP random number generator. The generator used analogue chaos sources initialised by a natural noise source to generate unlimited amounts of unbreakable OTPs that passed the NIST statistical tests. Personalising encryption locally by the client, prior to uploading data to the Cloud, gave complete control provided a new encryptor is used each time. This makes it expensive in encryptor keys but memory is cheap and plentiful.

During testing, the JavaScript application [36] was used to investigate how parameter changes affected the OTP entropy. The application also applied a von Neumann algorithm to maximise the entropy of the OTP. The GDPR legislation in 2018, will see hefty fines being imposed on companies who fail to meet the 72-hour deadline and it could be argued that our prototype means data was never available to a third party. Refinements to this system are being investigated such as extracting the patient personal information in the DICOM's metadata and encoding it separately from the medical image.

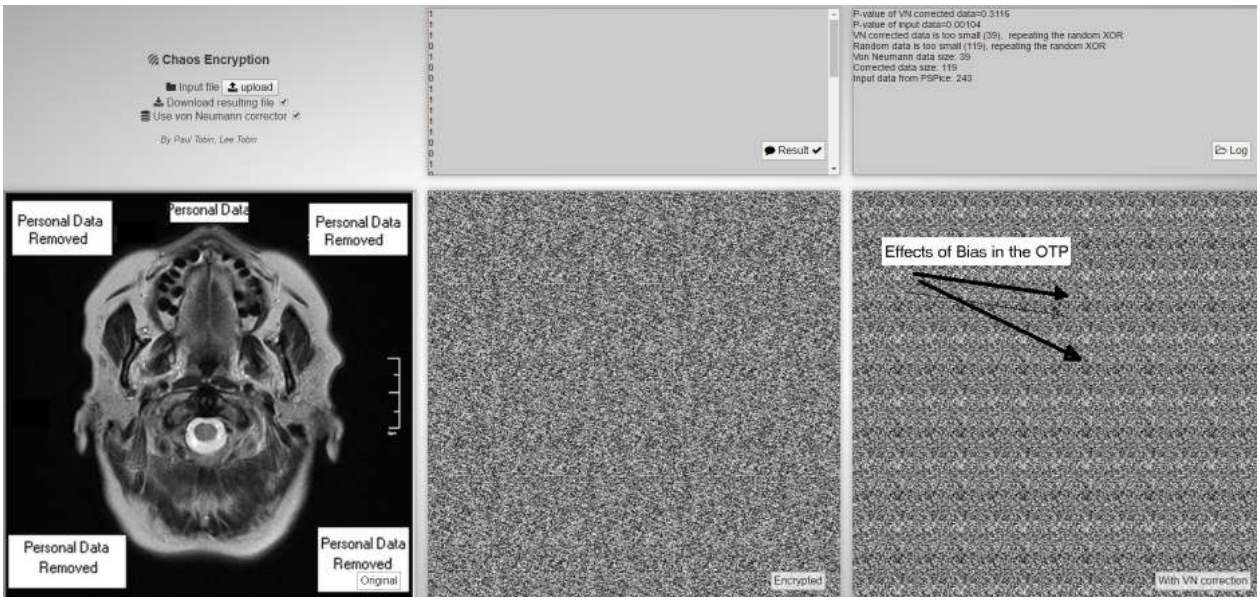

Figure 6. One-time pad JavaScript application.

TABLE I. NIST RESULTS for NOISE, SIMULATION and PROTOTYPE.

Statistical Test	*P-value natural noise*	*P-value simulation OTP*	*P-value prototype OTP*	*Passed/Fail*
Frequency test	P = 0.4122	P = 0.503	P = 0.403	Pass
Block frequency	P = 0.116	P = 0.216	P = 0.303	Pass
Runs	P = 0.7846	P = 0.508	P = 0.683	Pass
Block Longest Run Ones	P = 0.5388	P = 0.490	P = 0.553	Pass
Binary Matrix Rank	P = 0.7138	P = 0.333	P = 0.430	Pass
D Fourier Transform	P = 0.5206	P = 0.216	P = 0.420	Pass
Non-overlap Tp Match	P = NA	P = Na	P = NA	NA
Overlapping Tp Match	P = 0.7729	P = 0.002	P = 0.090	Pass
Universal	P = NA	P = NA	P = NA	NA
Linear Complexity	P = 0.9525	P = 0.263	P = 0.590	Pass
Serial	(P1 = 0.1971, P2 = 0.544)	P1 = 0.197, P2 = 0.544	P1 = 0.490, P2 = 0.509	Pass
Approximate Entropy	P = 0.1143	P = 0.201	P = 0.290	Pass
Cumulative Sums	P = 0.4444	P = 0.563	P = 0.490	Pass
Random Excursions	P = NA	P = 0.216	P = 0.230	Pass
Random Excursion Variant	P = NA	P = 0.216	P = 0.240	Pass

This would result in a much smaller OTP which could then be recombined with the image before uploading to the Cloud.

ACKNOWLEDGEMENT

The authors are grateful to Professor Michael Conlon and Dr Marek Rebow, Dublin Institute of Technology, for arranging the author's collaborative research programme.

REFERENCES

[1] [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral /service provider/global cloud index gci/Cloud Index White Paper.pdf, 2016, [retrieved: Mar 10, 2016].

[2] B. Duncan and M. Whittington,"Enhancing Cloud Security and Privacy: The Power and the Weakness of the Audit Trail", Conference on CLOUD COMPUTING, [retrieved: Sept 14, 2016], pp. 119-144.

[3] Verizon, "Verizon 2015 Data Breach Investigation Report", Tech. Rep., 2015, [retrieved: Sept 10, 2016]

[4] [Online]. Available: http://www.allenovery.com/SiteCollection Documents/Radical changes to European data protection legislation.pdf. http://www.allenovery.com/SiteCollection Documents/8 things you should be doing.pdf.

[5] [Online]. Available: https://en.wikipedia.org/wiki/Sony Pictures hack, [retrieved: May 10, 2016].

[6] D. Shumow and N. Ferguson, "On the possibility of a back door in the NIST SP800-90 Dual Ec Prng.", CRYPTO 2007 Rump Session, http://rump2007.cr.yp.to/15-shumow.pdf, August 2007.

[7] D. Hankerson, A.J. Menezes and S. Vanstone, "Guide to elliptic curve cryptography", Springer Science and Business Media, 2006.

[8] [Online]. Available: https://www.nist.gov/news-events/news/2014/04/ nist-removes-cryptography-algorithm-random-number-generator-recommendations.

[9] [Online]. Available: http://www.nytimes.com opinion aaron-sorkin journalists shouldn't help the Sony hackers.html, [retrieved: April 10, 2016], 2014.

[10] The New York Times, "Secret Documents Reveal N.S.A. Campaign Against Encryption", 2013.

[11] W. R. Bennett, "SIGSALY', IEEE Transactions on Communications", 31.1, 1983.

[12] P. M. Binder and R.V Jensen, "Simulating chaotic behavior with finite-state machines", Physical Review A 34(5), 1986, pp. 44604463.

[13] G. lvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems", Int. J. Bifurcat. Chaos 16(8), 2006, pp. 21292151.

[14] S. Li et al, "On the security of a chaotic encryption scheme: problems with computerized chaos", Comput. Phys. Commun. 153(1), 2003, pp. 5258.

[15] E. Salih, "Security analysis of a chaos-based random number generator for applications in cryptography", 15th International Symposium on

Communications and Information Technologies (ISCIT), IEEE, 2015, pp. 319-322.

[16] S. Ergn, S. Gler and U. Asada, “IC Truly Random Number Generators Based on Regular and Chaotic Sampling of Chaotic Waveforms”, Nonlinear Theory and its Applications, IEICE transactions, Vol. 2, 2011, pp. 246-261.

[17] E. K. Barker, Kelsey “Recommendation for the Entropy Sources Used for Random Bit Generation (draft)”, NIST SP800-90B, August, 2016.

[18] S. Ergn, U. Gler and K. Asada, “A High Speed IC Truly Random Number Generator Based on Chaotic Sampling of Regular Waveform”, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E94 A, 1, 2011, pp. 180-190.

[19] B. Schneier, “Applied Cryptography second edition”, John Wiley and Sons 1996.

[20] P. Tobin, L. Tobin, M. McKeever and J. Blackledge, “ Chaos-based Cryptography for Cloud Computing”, 27th ISSC conference Ulster University, Londonderry, June 21-22, doi: 10.1109, 2016, pp. 1-6.

[21] P. Kennedy, ”Genealogy of Chuas Circuit.”, In Chaos, CNN, Memristors and Beyond: A Festschrift for Leon Chua With DVD-ROM, composed by Eleonora Bilotta, 2013, pp. 3-24.

[22] J. Blackledge, A. Al-Rawi, and P. Tobin, “Stegacryption of DICOM Metadata”. In Irish Signals and Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014). 25th IET June, pp. 304-309, 2013. Chicago.

[23] P. Jees, and T. Diya, “Medical Image Protection in Cloud System”, matrix, V2, 2016, pp. 3.

[24] C.E. Shannon, “Communication Theory of Secrecy Systems”, Bell Technical Journal, vol.28-4, 1949, pp. 656715.

[25] J. Blackledge, “Cryptography and Steganography”: New Algorithms and Applications, Centre for Advanced Studies Text-books, Warsaw University of Technology, ISBN: 978-83-61993-05-6, 2012.

[26] E. Lorenz, “Chaos and Strange Attractors: The Lorenz Equations”, 1963, pp. 532-538.

[27] P. Tobin, “PSpice for Circuit Theory and Electronic Devices”, www.morganclaypool.com, ISBN:1598291564, pp. 127, 2007.

[28] P. Tobin, “PSpice for Digital Communications Engineering”, Synthesis Lectures on Digital Circuits and Systems, www.morganclaypool.com, ISBN:1598291629, 2007, pp. 97.

[29] J. von Neumann, “Various techniques used in connection with random digits”, Applied Math Series, 12, 1951, pp. 3638.

[30] J. Blackledge and N. Ptitsyn, “On the Applications of Deterministic Chaos for Encrypting Data on the Cloud”, Third International Conference on the Evolving Internet IARIA Luxembourg, (ISBN: 978-1-61208-008-6), 2011, pp. 78-87.

[31] J. Blackledge, S. Bezobrazov, P. Tobin and F. Zamora, “Cryptography using Evolutionary Computing”, (IET ISSC13 LYIT Letterkenny), 2013, pp. 1-6.

[32] P. Tobin, J. Blackledge, “Entropy, Information, Landauer’s Limit and Moore’s Law”, (IET ISSC14 UL, Limerick), 2014, pp. 1-6.

[33] R. Frigg, “In what sense is the KSE a measure for chaotic behaviour?”, (London School of Economics May), 2003.

[34] A. Ruk et al, “A statistical test suite for the validation of random number generators and pseudo-random number generators for cryptographic applications”, NIST http://csrc.nist.gov/rng/rng2.html, 2001.

[35] Random.org, “True Random Number Service”, [Online]. Available: http://www.random.org [retrieved: Aug 10, 2016]. 2013.

[36] [Online]. Available: http://jork.byethost7.com/chaosencrypt/