Exploring the stochasticity of chemical processes in an automated robotic crystallization platform to generate random numbers

Edward C. Lee,[1] Juan. M. Parrilla-Gutierrez[1], Alon Henson,[1] Euan. K Brechin[2] and Leroy Cronin[1*]

[1]University of Glasgow, Joseph Black Building, University Avenue, Glasgow, UK, G12 8QQ.

[2]University of Edinburgh, Joseph Black Building, David Brewster Road, Edinburgh, UK, EH9 3FJ

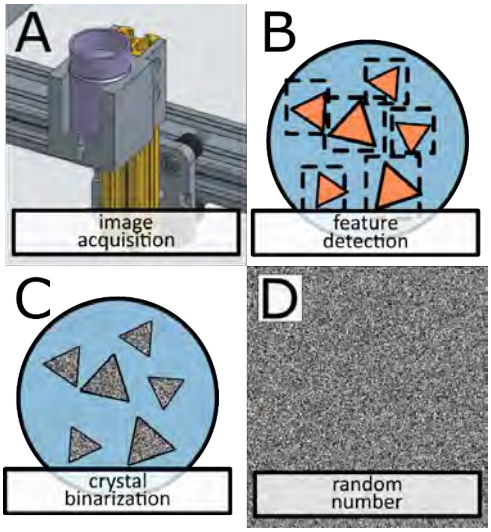
*Corresponding Author. E-mail: lee.cronin@glasgow.ac.uk

Abstract: Random number generators are important in fields which require non-deterministic input, such as cryptography. One example of a non-deterministic system is found in chemistry via the crystallization of chemical compounds, which occurs through stochastic processes. Herein, we present an automated platform capable of generating random numbers from observation of crystallizations resulting from multiple parallel one-pot chemical reactions. From the resulting images, crystals were identified using computer vision, and binary sequences were obtained by applying a binarization algorithm to these regions. An assessment of randomness of these sequences was undertaken by applying a barrage of tests for randomness described by the National Institute of Standards and Technology (NIST). We find that numbers generated through this method are able to pass each of the three levels for each of the NIST tests. We then compare the encryption strength of the random numbers generated from each of the crystallizing systems to that of a pseudo-random number generation algorithm (the Mersenne Twister). We find that messages encrypted using chemically derived random numbers take significantly longer to decrypt than the algorithmically generated number.

Random numbers are used extensively in many applications where their non-deterministic properties and unpredictability are essential, such as cryptography (*1*), scientific modelling (*2*) and lotteries (*3*). As such, both generation and validation of random numbers are important to ensure that all output is as desired and devoid of systematic determinism or predictability, which could affect the quality of the output (*4*). There are two approaches for generation of random numbers: computationally generated (pseudorandom) and generation via a non-deterministic physical process (true random) (*4*). Pseudorandom number generators (PRNGs), which create an arbitrarily long sequence of binary integers from an input seed using a mathematical algorithm, such as the Mersenne Twister (*5*), are deterministic and show long range correlations making them inappropriate for many applications. In contrast, the binary sequences produced by true random number generators (TRNGs) are inherently unpredictable and non-deterministic, increasing their quality but at the expense of their required resources. (*6*)

True random numbers cannot be generated computationally (*4*) and must instead be harvested and distilled from a physical entropy source which exhibits nondeterministic behavior. (*4*) For the purposes of random number generation, a bit sequence that results from any system can be considered random if it meets the following criteria: 1) it exhibits statistical properties of an ideal random number, 2) subsequent bits cannot be predicted from prior bits, and 3) it cannot be reliably reproduced (*7*). Many such true random number generators exist such as, cosmic background radiation, (*8*) lava lamps, (*9*) and radioactive decay (*10*) with recent developments including quantum entanglement (*11*), electronic noise (*12*) and social media (*13*). Many processes in chemistry are understood to be stochastic in nature. Phenomena controlling reaction kinetics such as molecular diffusion and collision rates, as well as thermodynamic phenomena such as chemical equilibria and phase changes can be modelled parsimoniously from stochastic assumptions (*14,15*). We hypothesized that by creating an automated device to observe a stochastic chemical process as a source of entropy, such as crystallization, it could be possible to generate sequences of truly random numbers which both pass standard tests for randomness and can thus securely encrypt data.

However, to the best of our knowledge, no one has ever attempted to generate true random numbers by observation of any chemical process. In this regard we developed an automated platform capable of performing chemical reactions and following the subsequent growth of crystals from the reaction products using a camera. It then converts these data into binary sequences that we assessed for randomness as shown schematically in Figure 1. We demonstrate the methodology used in, and



assess the results obtained from the generation of random numbers by using crystallization stochasticity as a primary entropy source.

Fig. 1. Schematic of procedure for generating random numbers using crystallization. Crystals, having been formed by reaction in a glass vial are imaged by a tracking camera. Pixels corresponding to crystals are detected by feature detection software and converted sequences of bits, here represented as an array of black and white pixel.

Robotic Platform. A robotic platform was designed to generate raw images for random number generation (Figure 2) from chemistry, based on a Computer Numerical Control (CNC) Machine. Using rapid prototyping techniques described previously, (*16*) an additional set of motorized linear axes were attached to the underside of the device to support a camera on a mobile gantry. The mobility in the main CNC framework and auxiliary framework were controlled using technology originally designed for the open source 'RepRap' 3D printer (*17*). Reagent stock solutions were

located adjacent to the platform, and could be transferred to vials in the crystallization array using a combination of tubing and pumps. Additional 3D printed components were incorporated to direct the reagent outlets, fix the positions of the vials in the array and support the camera (see Supplementary Figures S1-S7). Experiments consisted of pre-set volumes of stock solutions being pumped into each 14 ml vial in a vial array, sequentially. The subsequent growth of crystals in each of the vials was recorded by a mobile camera at regular 10 minute intervals at a resolution of 1280 x 800 pixels. Image analysis using a masked region based convolutional neural network (Mask R-CNN) (*18*) was employed to locate the crystals in the vial. The full methodology for platform construction and operation is described in the Supplementary Methods.

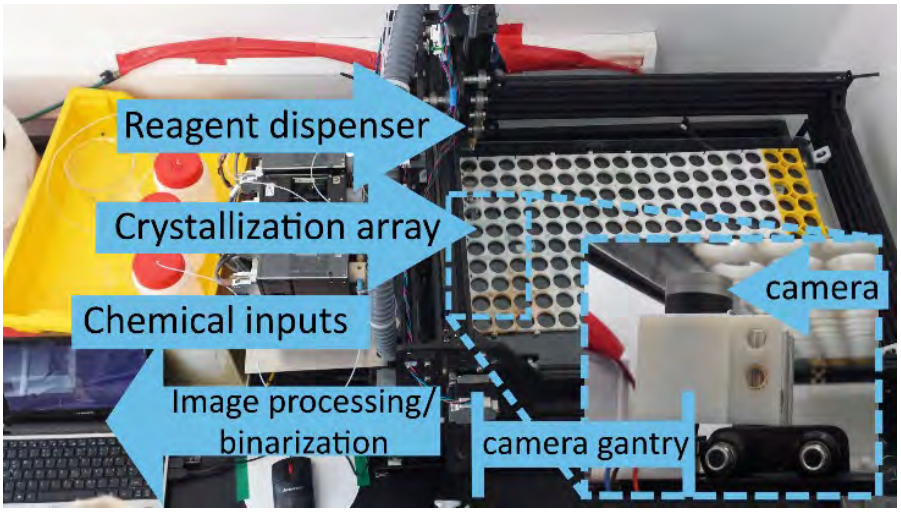
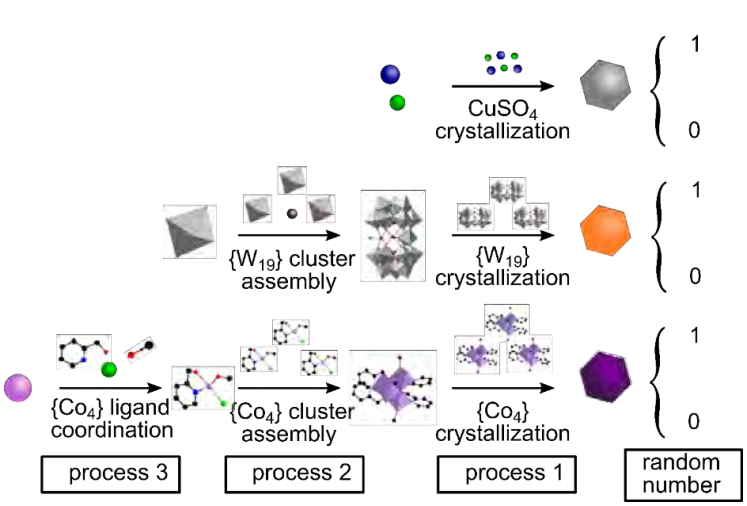
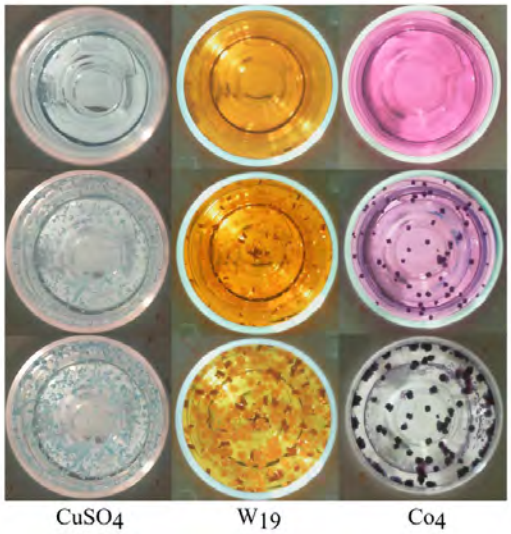


Fig. 2. Setup of the robotic system. Photograph showing the crystallization array inside the CNC framework and its relative position to the input stock solutions, pumps, camera and controlling computer.

Chemical inputs. Chemical inputs were chosen primarily such that they would produce macroscopically observable crystals in a timescale of minutes to hours without the formation of precipitate. We also utilised different reactions which exhibit additional stochastic chemical processes, namely 1) crystallization, 2) cluster formation and 3) ligand attachment to cluster. We hypothesized that by increasing the number of random events prior to observation of crystallization, the likelihood of the final output being random would increase. As such, three reactions were

chosen: recrystallization of the inorganic salt $CuSO_4.5H_2O$, the synthesis and crystallization of the polyoxometalate $[W_{19}Mn_2O_{61}Cl(SeO_3)_2(H_2O)_2]^{9-}$, hereafter referred to as $\{W_{19}\}$ (*19*), and the synthesis and crystallization of the coordination cluster $[Co_4(2\text{-pyridinemethanol})_4(MeOH)_4Cl_4]$,





hereafter referred to as $\{Co_4\}$. (*20*)

Fig. 3. LEFT: Chemical schemes for process investigation. $CuSO_4$ requires the stochastic process of crystallization alone, whereas $\{W_{19}\}$ and $\{Co_4\}$ require cluster formation in addition, and $\{Co_4\}$ requires the further step of ligand attachment. **RIGHT**: Reactions to form crystals of $CuSO_4$, W_{19} and Co_4. Top – Initial reaction solutions (time = 0 minutes). Middle – partially complete crystallization (time = 40 minutes). Bottom – crystallizations at the end of the experiment (time = 150 minutes)

Snapshots of these crystallizations at different times are shown in Figure 3 and the chemical structures of these are shown in Supplementary Figures 8-10. These reactions involve the stochastic processes of (1), (1 and 2), and (1, 2 and 3), respectively. The synthetic procedure files are included in the online repository and were performed in a fully automated manner, and confirmation was obtained by performing single crystal X-ray diffraction. To test this hypothesis, we assessed each of the resultant binary sequences using the tests available in NIST SP-800 22a (*7*).

Binarization Procedure. For each crystallization experiment, an initial set of images was taken of the crystallization at 10 minute intervals for three hours. These were then retrospectively annotated by hand with the pixels corresponding to each crystal using Visual Geometry Group Image Annotation software (*21*). Each crystal dataset was then used as the basis to generate a model file using the Mask RCNN technique (18). Once models had been generated, they could be incorporated into the platform software to locate crystals in new samples as shown in Supplementary Figure S13. The rim of each vial was detected as the brightest circular object in each image (see Supplementary Figure S14 for details). Once the location of these features had been detected, a binary sequence was formed using the position and intensity properties of crystals within the vial in the following way (and shown in Figure 4):

1. Search each row of the image for a pixel inside the vial which has been classified as belonging to a crystal.
2. Count the number of adjacent crystal pixels in this row and this column (P_r, and P_c, respectively); calculate the distance of this pixel from the vial centre (P_d); and determine the greyscale value of the pixel (P_v).
3. Calculate an 8-bit location component for the pixel (P_L) by multiplying P_r, P_c and P_d modulus 256.
4. Apply the exclusive OR operator between P_L and P_v to obtain the final 8-bit sequence resulting from a crystal pixel.

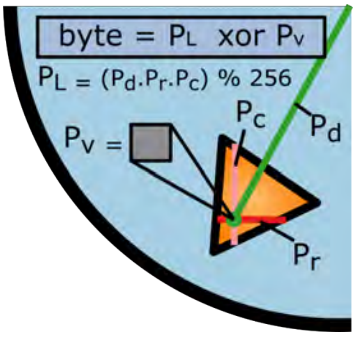


Fig. 4. Binarization method. For each crystal pixel (P), distance from center (P_d), crystal width (P_r) and crystal height (P_c) are multiplied together and the modulus of 256 is taken. This 8-bit number undergoes an exclusive OR operation with the greyscale pixel color value (P_v) to generate a random 8-bit sequence.

Randomness Evaluation. The output binary sequences were evaluated for randomness using the tests for randomness published in NIST SP-800-22a (*7*). These identify features of an ideal random binary sequence, such as an equal number of 0s and 1s, and assesses the probability that an input test binary sequence exhibits those features. A summary of each test is presented in Supplementary Table S1. For each test a feature (such as equal bit frequency) is identified and a null hypothesis (that the feature is present, H_0) and alternative hypothesis (that the feature is not present, H_a) are generated. Applying the test to the sequence generates a p-value representing the likelihood that the null hypothesis is true. Typically, the null hypothesis should only be rejected if the p-value is below a specified threshold, often set as 0.01.

Once a sequence has undertaken a test, two further levels of assessment must be passed to ensure that the generator indeed produces random sequences. The second level requires that each test is carried out many times, with the p-values for each test being recorded. Then a confidence interval can be assigned (using Supplementary Equation S1) for the expected pass rate when running each test.[7] The third level, also known as the p-value of p-values, assesses the uniformity of p-values obtained from running each test multiple times. In this case, the p-values are plotted in a histogram with 10 bins, a further p-value is generated (using Supplementary Equations S2 and S3) based on how likely it is that the histogram originated from a uniform distribution. Sequences can be considered uniformly distributed if the observed p-value if much greater than 0.0001 (*7*). In general, a binary sequence generated from one image of one reaction was insufficient in length to produce valid results for each level of test. Instead, sequences from images taken at the same time in one experiment were concatenated to produce new sequences containing several million bits. We find that numbers generated from each of the crystallization systems were able to generate binary

sequences which easily passed every NIST test at each level, with an example of the histogram generated by first order NIST testing on crystallization of W_{19} shown in Figure 5. First, second and third level results for each compound are shown in Supplementary Figures 5.1-5.3.

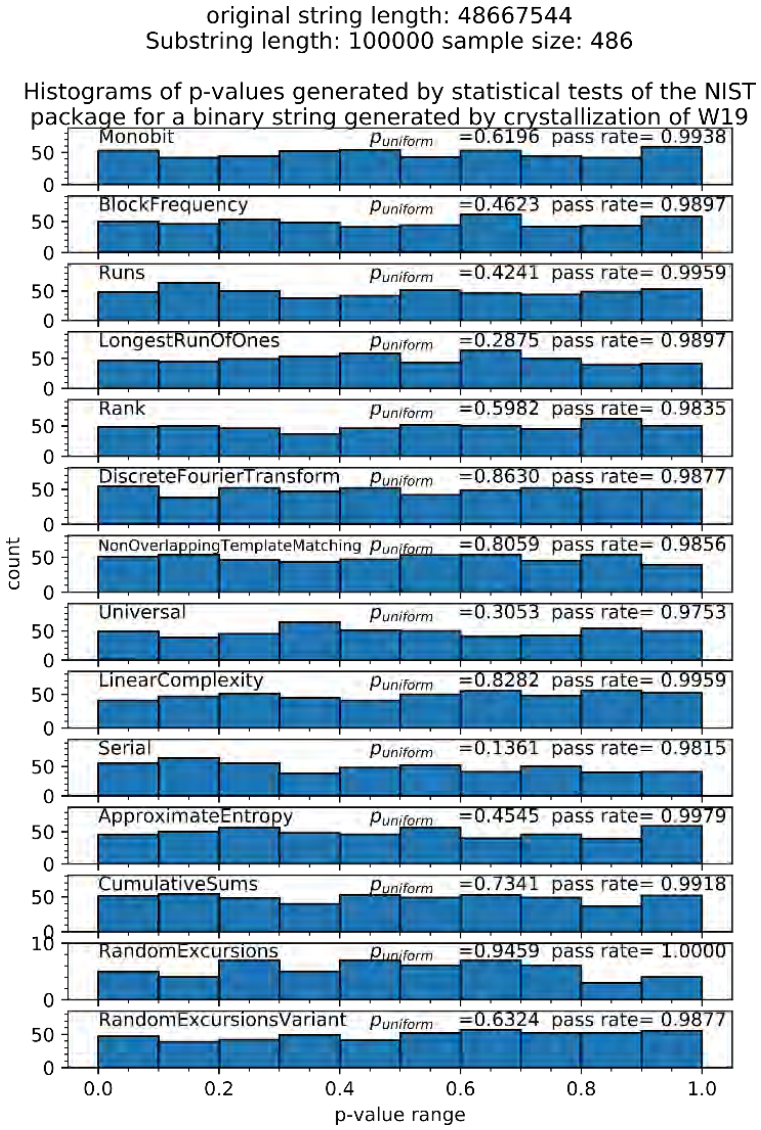


Fig. 5. Results of NIST testing for a sequence generated by {W_{19}} at a time of two hours. The histogram consists of p-values obtained by running the first level testing on a single sequence divided into 486 blocks of length 100,000.

Encryption Efficacy. True random numbers have an advantage over pseudo-random numbers in that they are non-deterministic and therefore cannot be predicted with greater (or less) than 50%

certainty. We wanted to use this feature to compare the encryption strength of our numbers to that of a pseudo-random number generator. The Mersenne Twister (MT)[5] is a pseudo random number generator used as the default random number generator in several applications including Microsoft Excel, Python, R and MATLAB. It algorithmically generates a number based on its current state, which contains 624 32-bit numbers. Once this state is determined, its output can be determined with a high degree of accuracy.[5]

We compared the encryption strength of the MT with that of numbers generated from our system by generating a large set of 8-bit keys from each system, using each key to encode the word 'crystal' and observe the time taken to determine the original encryption key. Both systems used a 'brute-force' approach where every combination of bits was sequentially checked. However, in the case of the MT, the keys were retained in memory in order to determine the state of the MT. Once the state was determined, subsequent keys were predicted and tested preferentially in the 'brute-force' methodology. The results (Figure 6) show that the average time taken to decrypt 8-bit keys from a sample of 128000 keys generated by the crystal random number generator is approximately the same as that of the uncracked MT (4ms). However, when the 'cracking' methodology was applied, the average decryption time was reduced to 30μs.

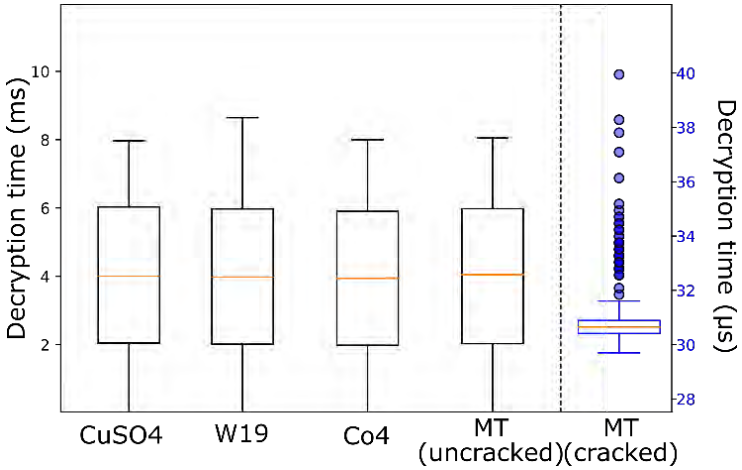


Fig. 6. Boxplots of times for the message 'crystal' to be decrypted after encryption using 8-bit keys generated by different methods. Keys generated using the crystallization random number generator on average take as long to crack as those produced using the uncracked Mersenne Twister

(MT), however, once the state of the MT is determined, the average decryption time for this method is substantially shorter than the crystallization method.

Discussion

We have shown that crystallization of chemical compounds can be used to create sequences of numbers which conform to statistical distributions expected from truly random bit sequences. These bit sequences can be generated using the stochasticity inherent in chemical transformations as an entropy source, followed by imaging, feature detection and binarization algorithms in an automated platform. However, we could not detect any relation between number of prior stochastic processes and degree of randomness produced between different crystallizations. Finally, we show that numbers produced in this manner are superior to those produced by the Mersenne Twister, a common pseudo random number generator in terms of encryption strength, as the latter can easily be cracked.

Unique advantages of this system over other generators of random numbers are that the generation may be done in private, with no outside observers, that it is relatively cheap and easy to construct, that it may be extended to allow more crystallizations and longer random number sequences and finally that it is applicable to many different chemical systems. Further work may include investigating the properties of bit sequences generated from other crystallization systems, such as crystals whose components are incommensurate in size. In conclusion, this article describes the first application of chemistry to the generation of random numbers. The methodology can be set up and applied inexpensively without any restriction on location to generate large sequences of random numbers.

Materials and Methods

Materials and Methods for this article are listed in the supplementary materials due to their length.

H2: Supplementary Materials

Materials and Methods S1-S2

Figs. S1-S18

Table S1

Equations S1-S3

References and Notes

1. Shannon, C. E. Communication theory of secrecy systems. 1945. *MD Comput* **15,** 57–64 (1998).

2. Metropolis, N. & Ulam, S. The Monte Carlo Method. *Am. Stat. Assoc. JOURNAL,* **44,** 335–341 (1949).

3. Genest, C., Lockhart, R. A. & Stephens, M. A. X^2 and the Lottery. *Stat.* **51,** 243–257 (2002).

4. Stipčević, M. & Koç, Ç. K. True Random Number Generators. *Open Probl. Math. Comput. Sci.* 275–315 (2014). doi:10.1007/978-3-319-10683-0_12

5. Matsumoto, M. Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator Dedicated to the Memory of Nobuo Yoneda. *ACM Trans. Model. Comput. Simul.* **8,** 3–30 (1998).

6. Park, S. K. & Miller, K. W. Random number generators: good ones are hard to find. *Commun. ACM* **31,** 1192–1201 (1988).

7. National Institute of Standards and Technology. A statistical test suite for random and pseudorandom number generators for cryptographic applications - special pub. 800-22 - Rev. 1. 1/1--G/1 (2010). doi:10.6028/NIST.SP.800-22r1a

8. Pimbblet, K. A. & Bulmer, M. Random numbers from astronomical Imaging. *Publ. Astron. Soc. Aust.* **22,** 1–5 (2005).

9. G.G. Liversidge J.F. Bishop, D.A. Czekai, K. C. C. United States Patent (19) 54. **96,** 62–66 (1980).

10. Martini, G. & Bruno, F. G. True Random Numbers Generation from stationary Stochastic Processes. *2017 Int. Conf. Noise Fluctuations, ICNF 2017* 1–4 (2017). doi:10.1109/ICNF.2017.7985997

11. Zhang, Q., Deng, X., Tian, C. & Su, X. Quantum random number generator based on twin beams. **42,** (2017).

12. Petrica, L. FPGA optimized cellular automaton random number generator. *J. Parallel Distrib. Comput.* **111,** 251–259 (2018).

13. De Virgilio, R. & Maccioni, A. Random Query Answering with the Crowd. *J. Data Semant.* **5,** 3–17 (2016).

14. Mcquarrie, D. A. & Gillespie, D. T. Stochastic Theory and Simulations of Chemical Kinetics. *J. Appl. Probab.* **478,** 413–478 (1967).

15. Van den Broeck, C. *Stochastic thermodynamics: A brief introduction*. *Procedings of the International School of Physics 'Enrico Fermi'* (2013). doi:10.3254/978-1-61499-278-3-155

16. Gutierrez, J. M. P., Hinkley, T., Taylor, J. W., Yanev, K. & Cronin, L. Evolution of oil droplets in a chemorobotic platform. *Nat. Commun.* **5,** 5571 (2014).

17. Jones, R. *et al.* Reprap - The replicating rapid prototyper. *Robotica* **29,** 177–191 (2011).

18. He, K., Gkioxari, G., Dollar, P. & Girshick, R. Mask R-CNN. *Proc. IEEE Int. Conf. Comput. Vis.* **2017–Octob,** 2980–2988 (2017).

19. Symes, M. D. *et al.* Integrated 3D-printed reactionware for chemical synthesis and analysis. *Nat. Chem.* **4,** 349–354 (2012).

20. Yang, E. C. *et al.* Cobalt single-molecule magnet. *J. Appl. Phys.* **91,** 7382–7384 (2002).

21. Dutta, A., Gupta, A. & Zisserman, A. *Visual Geometry Group Image Annotator* Available at: http://www.robots.ox.ac.uk/~vgg/software/via/.

Acknowledgments

We gratefully acknowledge financial support from the EPSRC (Grant Nos EP/P00153X/1, EP/J015156/1, EP/K021966/1, EP/K038885/1, EP/L015668/1, EP/L023652/1), ERC (project 670467 SMART-POM).

Author Contributions

The idea was conceived by LC and developed by ECL with help from JMP, AH and EKB. ECL built the robot, wrote the code, did the reactions and analyzed the data with help from JMP and AH. All the authors helped write the manuscript.

Competing interests: Authors declare no competing interests

Data and materials availability: All data are available in the main text or the supplementary materials, and the software can be found in the hub repository located at *http://datalore.chem.gla.ac.uk/ECL/RNGbot*

Supplementary Materials for

Exploring the stochasticity of chemical processes in an automated robotic crystallization platform to generate random numbers

Edward C. Lee,[1] Juan. M. Parrilla-Gutierrez[1], Alon Henson,[1] Euan. K Brechin[2] and Leroy Cronin[1*]

[1]University of Glasgow, Joseph Black Building, University Avenue, Glasgow, UK, G12 8QQ.

[2]University of Edinburgh, Joseph Black Building, David Brewster Road, Edinburgh, UK, EH9 3FJ

Correspondence to: lee.cronin@glasgow.ac.uk

This PDF file includes:

Supplementary Methods S1: Construction of robotic hardware

The robotic platform described in this document was built by combining prior developments of the RepRap 3D printer project17 with a commercially available Computer Numeric Control (CNC) platform. The open-source nature and large amount of available documentation facilitated prototyping, development and implementation of the platform. This section is intended to give a complete methodology for recreating the random number generator.

Robot frame

The main body of the platform consisted of an OX CNC Mechanical Kit, with dimensions of 500x750mm, giving a large area for multiple automated reactions and control of reagent output in the X, Y and Z dimensions. Reaction vials were then located on a supported glass sheet, below which a mobile camera was implemented on a set additional linear axes. A single mobile camera was incorporated (as opposed to an array of multiple fixed cameras) in order to prevent excess costs, and the delay in imaging between separate vials was considered not significant for data acquisition. All v-slot beam connections were made using 90 Degree Angle Corners. Every mobile axis had an end stop attached at the end in order to define the zero position of the axis. The complete platform is shown in Supplementary Figure S1

Mechanical Design

The final platform design used for all experiments in this publication is shown in Supplementary Figure 1a/b (photograph/schematic). This consists of the i) CNC kit, which positions where reagents are to be dispensed; ii) a crystallization platform, which supports reaction vials and maintains their positions; iii) an imaging support system, which moves a camera to locations below vials in order to capture images; and iv) a fluid handling system which controls reagent transfer from stock solutions to reaction vial.

CNC kit

The CNC kit used was an OX CNC Mechanical Kit with dimensions 500 x 750mm, black anodise extrusion color, and four NEMA23 – 175 oz – 2.0A stepper motors. It was constructed as directed in the manual.

Crystallization platform

The crystallization platform consisted of a sheet of glass with dimensions 350 x 400 x 3 mm. To support this, four 20 x 20 x 100mm v-slot aluminium beams were attached to the underside of the CNC Kit Y-frame: 50mm from each end of both sides and directed in towards the device in parallel with the X-frame. The vials used were made of glass, with a 14ml capacity and 12mm radius. An array of these vials was created on the glass surface held in place using interlocking 3D printed vial holders. We used an array of 10 x 10 vials to create experiments with 100 vials in them, although there was space available for more. A schematic of the crystallization platform and CNC kit is shown in Supplementary Figure 1.2.

Imaging support system

The platform was raised using four vertical 20x20x300mm v-slot aluminium beams connected at each corner of the CNC kit, and secured at the base using two horizontal 20x20x500mm V-slot aluminium beams and two horizontal 20x20x750mm V-slot aluminium beams running parallel with the Y and X axes, respectively. Two further horizontal 20x20x500mm were attached to the vertical beams along the Y axis 200mm below the CNC kit. The camera mobility was enabled using a set of belt driven linear actuators, each containing small v-slot four wheeled gantries, a 20 x 40 mm extrusion profile, black anodise colour and NEMA23 – 175 oz – 2.0A stepper motors. Two of these of length 750mm were attached at either side of the platform to the horizontal beams 200mm below the main framework in parallel with the X axis of the platform, forming the cY axis. A third actuator of length 500mm was fixed at either end to the gantries of the two 750mm linear axes and ran in parallel to the Y axis. Vertically attached to the cY axis gantry was a 20x20x100mm v-slot aluminium beam which acted as a support for the camera holder. A webcam was placed in the holder pointing up towards the crystallization platform. This was connected to a laptop via USB. A schematic of the whole Imaging Support system is shown in Supplementary Figure S2, and a schematic of the mobile camera gantry alone is shown in Supplementary Figure S3.

Fluid handling system

The fluid handling system consisted of a simple aluminium support of dimensions 250 x 120 x 5 mm attached distally to the back of the Y-axis of the CNC kit. It was supported by two 20 x 20 x 100mm v-slot aluminium beams. The sheet contained drilled holes for attaching five Tricontinent C series syringe pumps. The reagent stock solutions were located in a secondary containment tray adjacent to the platform and were not fixed.

Pumps

Five Tricontinent C Series pumps were attached to aluminium sheet described earlier and fitted with 5ml syringes.

Tubing

To connect the stock solutions to the pumps, 1/16 inch Fluorinated Ethylene Propylene (FEP) tubing was cut to size (~300mm) and attached using 1/16 inch flangeless PEEK fitting nuts To prevent leakage. Further 1/16 inch FEP was cut to size (~1200mm) and directed to the base of the Z-axis via a hollow flexible polyethylene tube (~1000mm). The tube was attached to the back of the CNC Y-axis and top of the Z axis motor. A 3D printed support device (Supplementary Figure S8) was attached to the end of the tubing at the base of the Z axis in order to direct the outlets of the tubing.

Axis control

Axis positioning was controlled by an Arduino 2560 Mega attached to a RepRap Arduino Mega Pololu Shield v1.4 (RAMPS) and connected to a laptop. This was powered by an ATX 500W power supply and connected to the motors and end stops of 5 linear axes. Three of the axes (X, Y and Z) controlled the reagent output location via the X, Y and Z RAMPS pins, and the other two (cX, and cY) control the location of the camera via the E0 and E1 RAMPS pins. Both the X and

cX axes consisted of two rails at either side of the platform and were connected electronically that they were powered by the same source. Five of the six end stop pins were attached to each of the axes- one for each axis.

Pump control

The pumps mentioned in subsection 1.2.1 were daisy chained together and controlled via USB. Power was sourced from the same ATX 500W power supply mentioned in section 1.3.1.

Bill of Materials

- The device body was constructed from an Ooznest OX CNC mechanical kit of dimensions: 500x750mm.
- The platform was raised using four 300mm (20x20mm) V-slot linear rails. Further linear rails (2x 500mm (20x20mm) and 2x 750mm (20x20mm)) were used to secure the base of the frame using 90-degree angle corner brackets.
- Two 750mm Mini V Linear Actuator Kits were attached to the vertical V-slot-linear rails in parallel along the length of the platform. One 500mm Mini V Linear Actuator Kit was positioned perpendicularly to this and fixed to the gantries of the 750mm Linear Actuator kits.
-The syringe pumps used were “TriContinent C-Series”. Each of them used 5ml syringes connected to identical 3-way PEEK valves
- An Arduino Mega 2560 and was used to control arm motors
- A RepRap Arduino Mega Polulu Shield was used to house the stepper drivers for motor arm control
- The stepper driver used to power the arm motors were Pololu a4988.
- “IDEX Health Science FEP Ora 1/16 x 0.20” tubing was to connect the stock reagents to the syringe pumps, and from the pumps to the vials.
- Flangeless fitting nuts, 1/16" OD Tubing, PEEK, were used to connect these tubes to the syringe pumps and device, with corresponding cone shaped fitting.
- “Microsoft LifeCam Cinema Webcam (H5D-00014)” was used to record images of the crystallizations.
- 3D printed objects were composed of polylactic acid (PLA)

Software implementation

The robotic platform was controlled using code written in a combination of Python and C++ programming languages. Python was used to define the experiment controller, in which experimental procedures, experimental inputs and logic of the platform were defined. This was run in parallel to the binarization procedure, which performed image analysis using computer vision, the binarization routine and verification of binary strings using the NIST barrage. C++ was used solely in the Arduino firmware in order to convert instructions written in Python to physical output to the motors on the axes.

Experiment controller

The experiment controller was used to read an input file, specifying compound name, reagent volumes, reagent addition times, image acquisition period, number of images to collect and number of reactions to perform, and perform reactions and image acquisitions to meet these criteria. It adopted an object orientated approach, where the experiment, robotic platform, reaction routine and imaging routine were individual classes.

Firmware

The firmware was uploaded only on the Arduino board and was responsible for the motion of the axes. It was written in C++ as this is the native language for Arduino development. The code utilised two libraries, CommandHandler and CommandManager, to link output pins to specific motors and these are located on GitHub. The motors could then be controlled through Python using another library, Commanduino, also located on GitHub.

Image Analysis

Crystal detection relies on the use of models generated by Mask RCNN software. Initial datasets were labelled manually to create a model for each crystal type. Once models were created, they could be run on any image to identify crystals in that image. Each crystal was saved as a separate mask file, which contained pixels where the crystal was present, and black pixels elsewhere. Vial detection was performed by locating the brightest circular contour in the image from computer vision techniques. Vial centre and radius were then determined using 'minimum enclosing circle' technique. These data were saved as a separate text file.

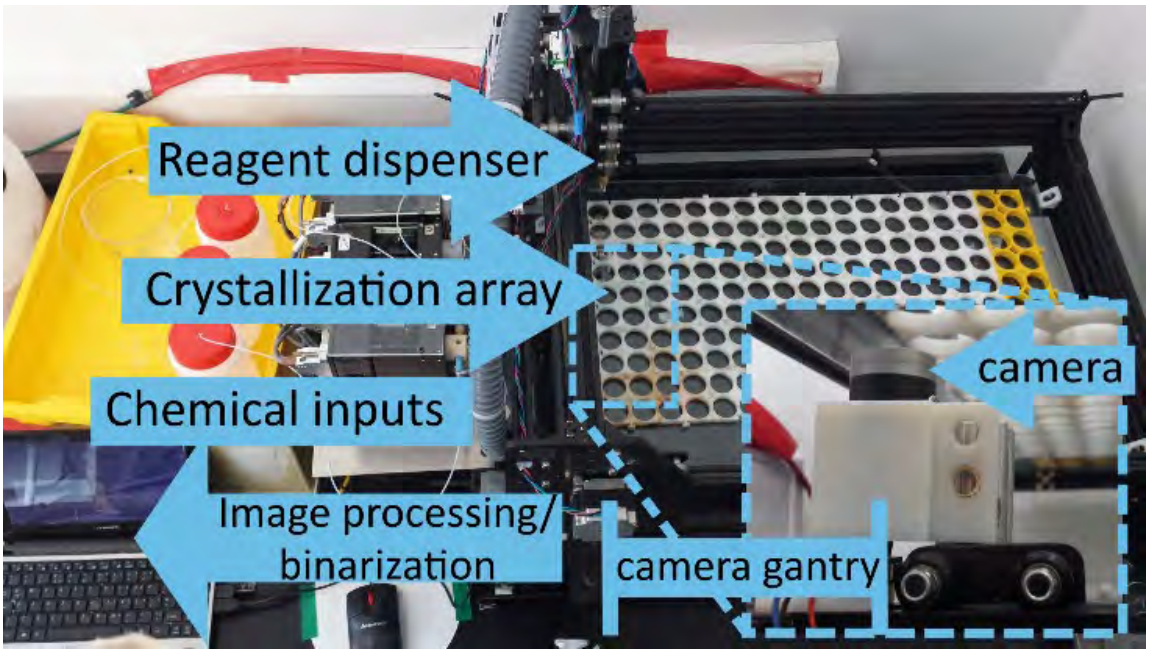
Binarization

Binarization was obtained by iterating through crystal masks to find crystal pixels that are within the radius of the vial circle. This was followed by a calculation of the crystal width and height at pixel, as well as the distance from vial centre to the pixel. These values are multiplied apply the modulo of 256 to this figure to get an 8-bit string at this pixel. The original image is then converted to greyscale, and the color intensity value of that pixel (which is in the range of 0-255) is taken as a second 8-bit string. Finally, an exclusive OR operation is applied between the two 8-bit strings and the output is taken as the random string for that pixel. Longer random strings were then built up by concatenating strings obtained from pixels of one crystal, crystals of one image, and images of each reaction at synchronous times in an experiment.

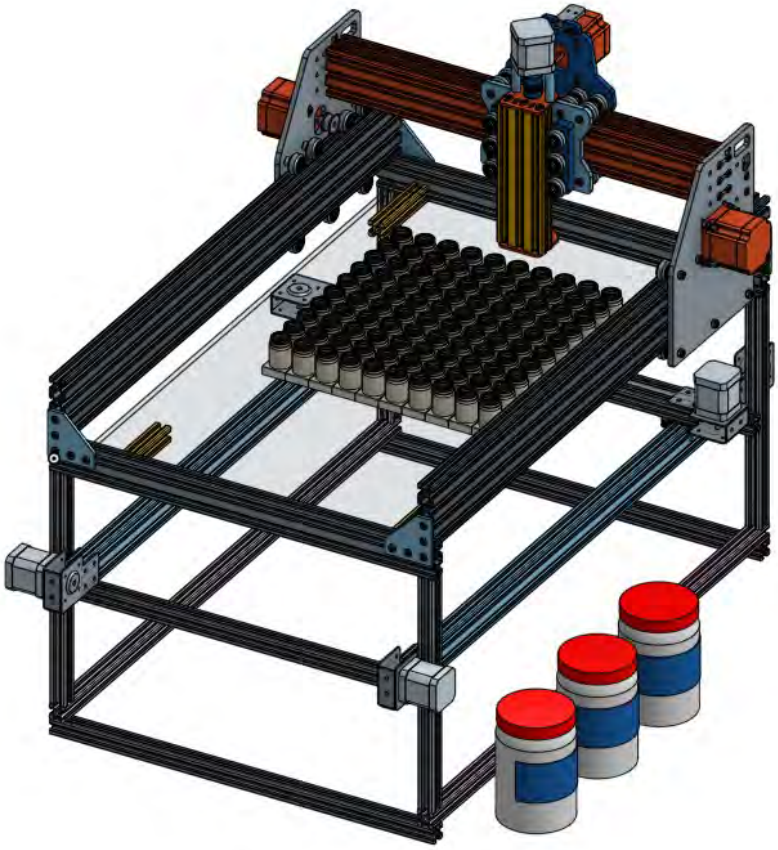
NIST barrage

The NIST barrage was incorporated into the analysis capability of the platform to allow continuous assessment of number strings being produced. Each of the 15 tests were written in Python and verified against the bit-strings provided in the package for the numbers e, pi, $\sqrt{2}$, and $\sqrt{3}$. These numbers are also provided in the analysis software of the platform. A brief description of each test can be seen in Supplementary Table 1.

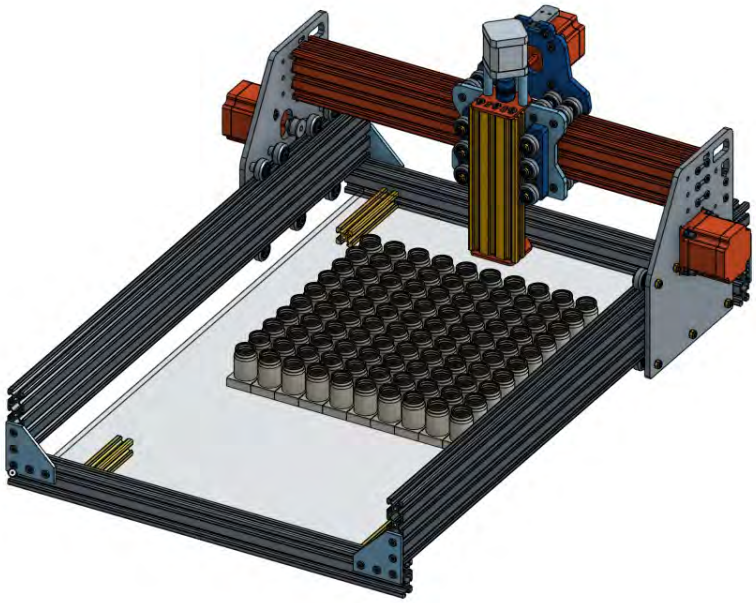
Supplementary Figures



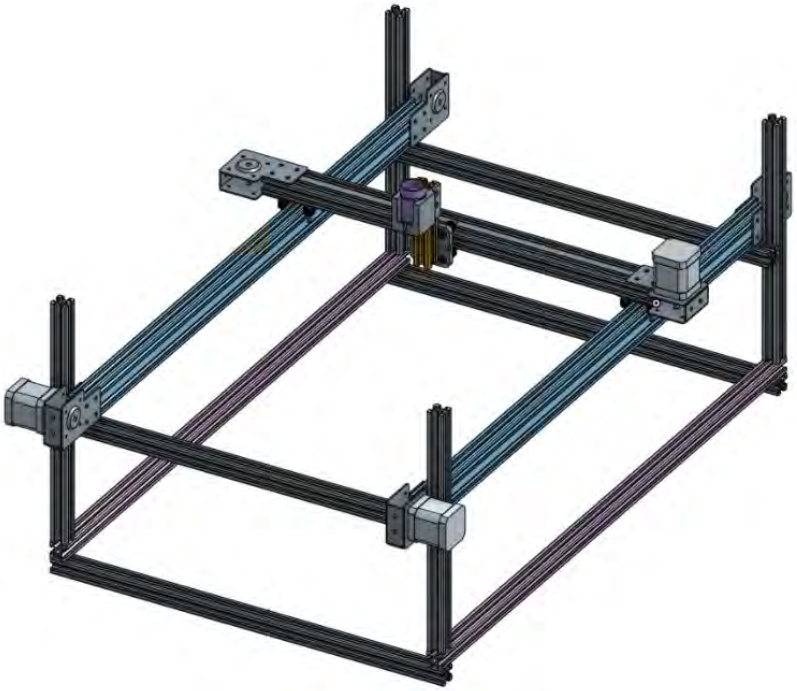
Supplementary Figure S1. Setup of the robotic system. Photograph showing the crystallization array inside the CNC framework and its relative position to the input stock solutions, pumps, camera and controlling computer, with 3D printed vial holders (Supplementary Figure S2) shown in the crystallization array.



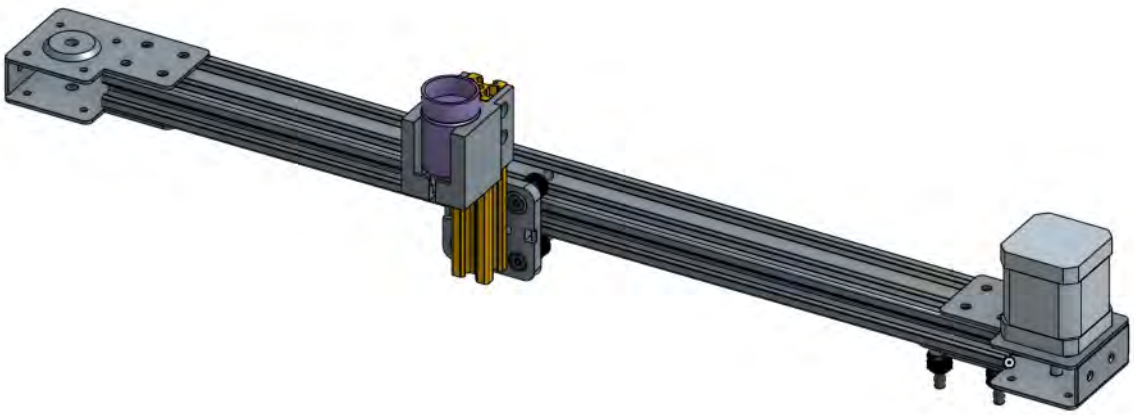
Supplementary Figure S3. Overall schematic of the robot. The CNC device was raised with fitted with a glass sheet to support an array of glass vials. Stock solutions were connected to vials via pumps and tubes which were connected to the main arm of the CNC device. Struts were added to allow the positioning of a camera using linear axes.



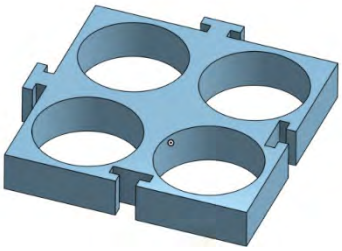
Supplementary Figure S4. Schematic of the upper half of the robot CNC kit and crystallization platform



Supplementary Figure S5. Schematic of the lower half of the robot. The camera positioned on a mobile gantry (Supplementary Figure S6) is visible.

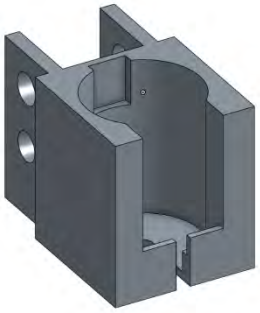


Supplementary Figure S6. Mobile camera gantry. The camera is positioned in a 3d printed holder (Supplementary Figure S9) and attached to the gantry of a linear axis via an aluminium strut. Both ends of the linear axis are attached to the gantries of two further linear axes which are themselves attached to the main robot frame (not shown).

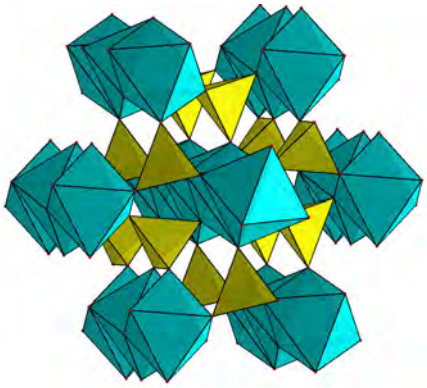


Supplementary Figure S7. 3d printed vial holder. Design allows individual pieces to be clipped together permitting an array to be created.

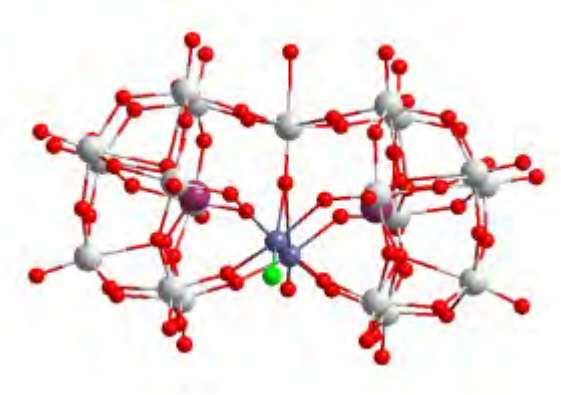
Supplementary Figure S8. 3d printed part to direct tubing output into vial.



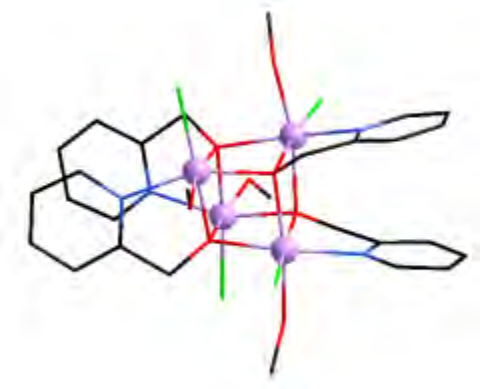
Supplementary Figure S9. 3d printed camera holder, with attachment points to an aluminium strut



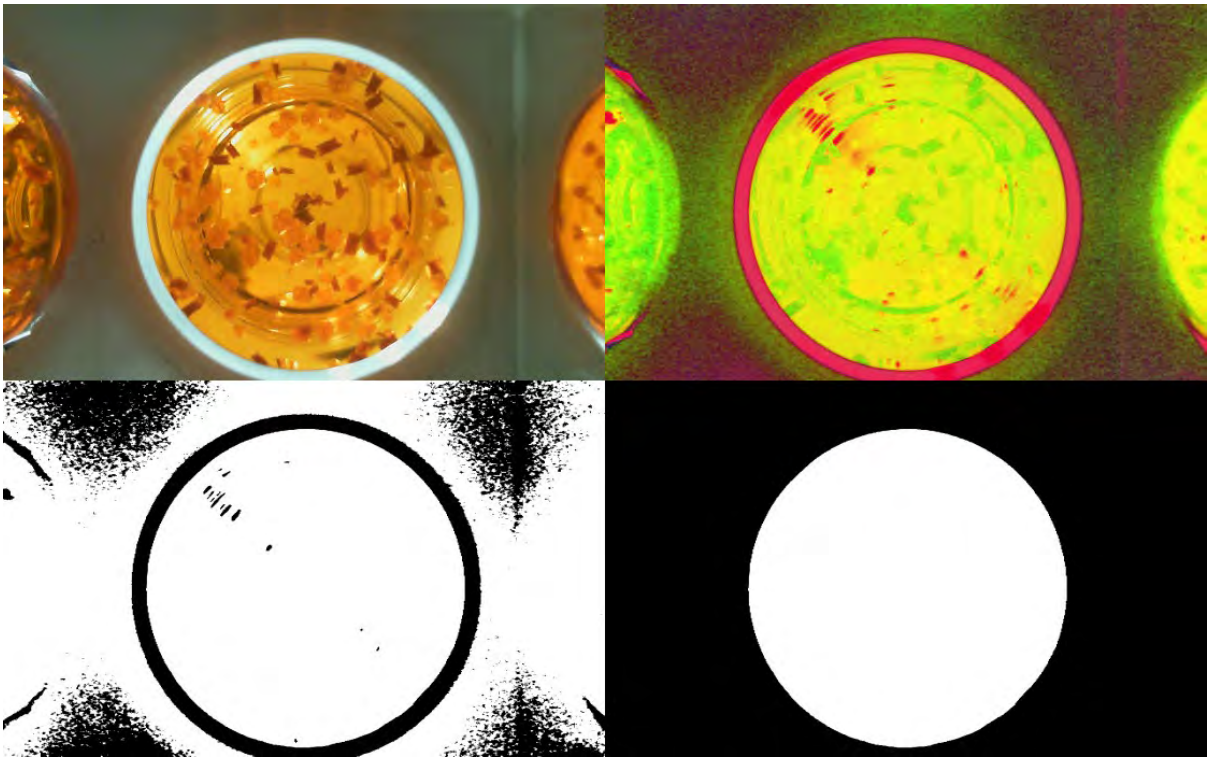
Supplementary Figure S10. Packing Structure of $CuSO_4$.. Turquoise polyhedra = CuO_6, Yellow polyhedra = SO_4



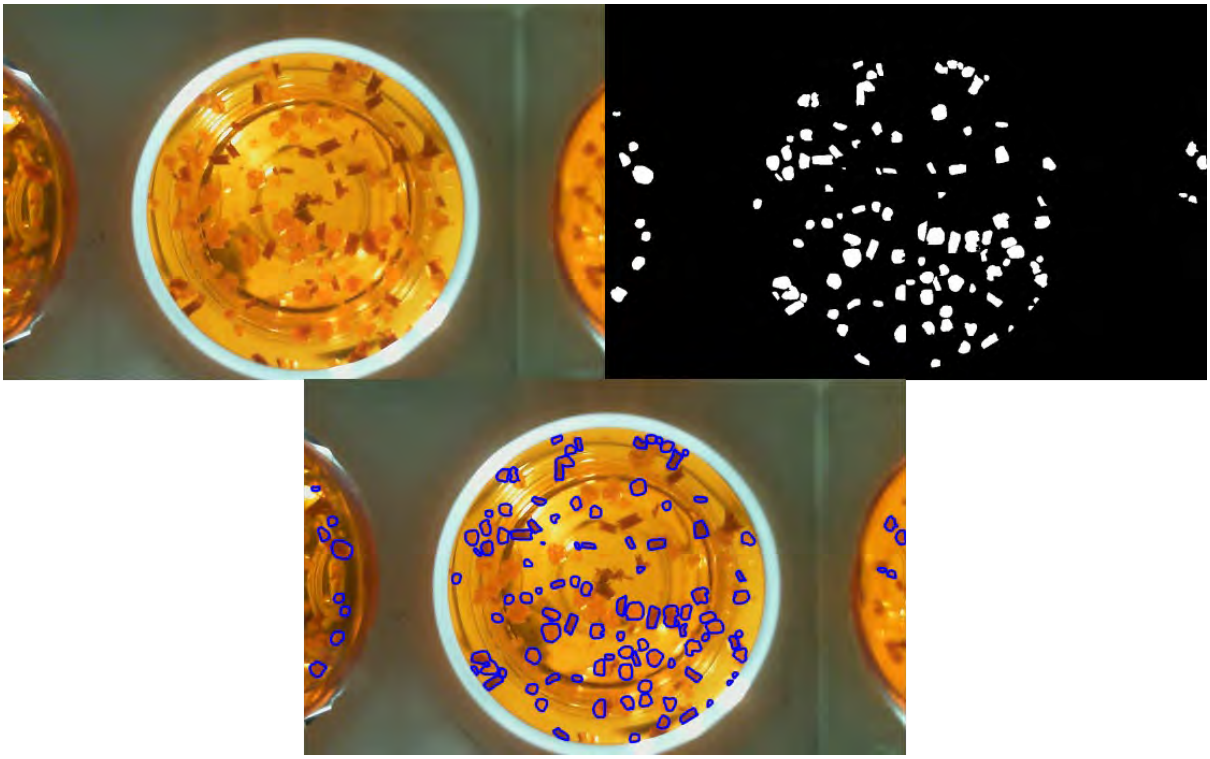
Supplementary Figure S11. Structure of $\{W_{19}\}$ (19)



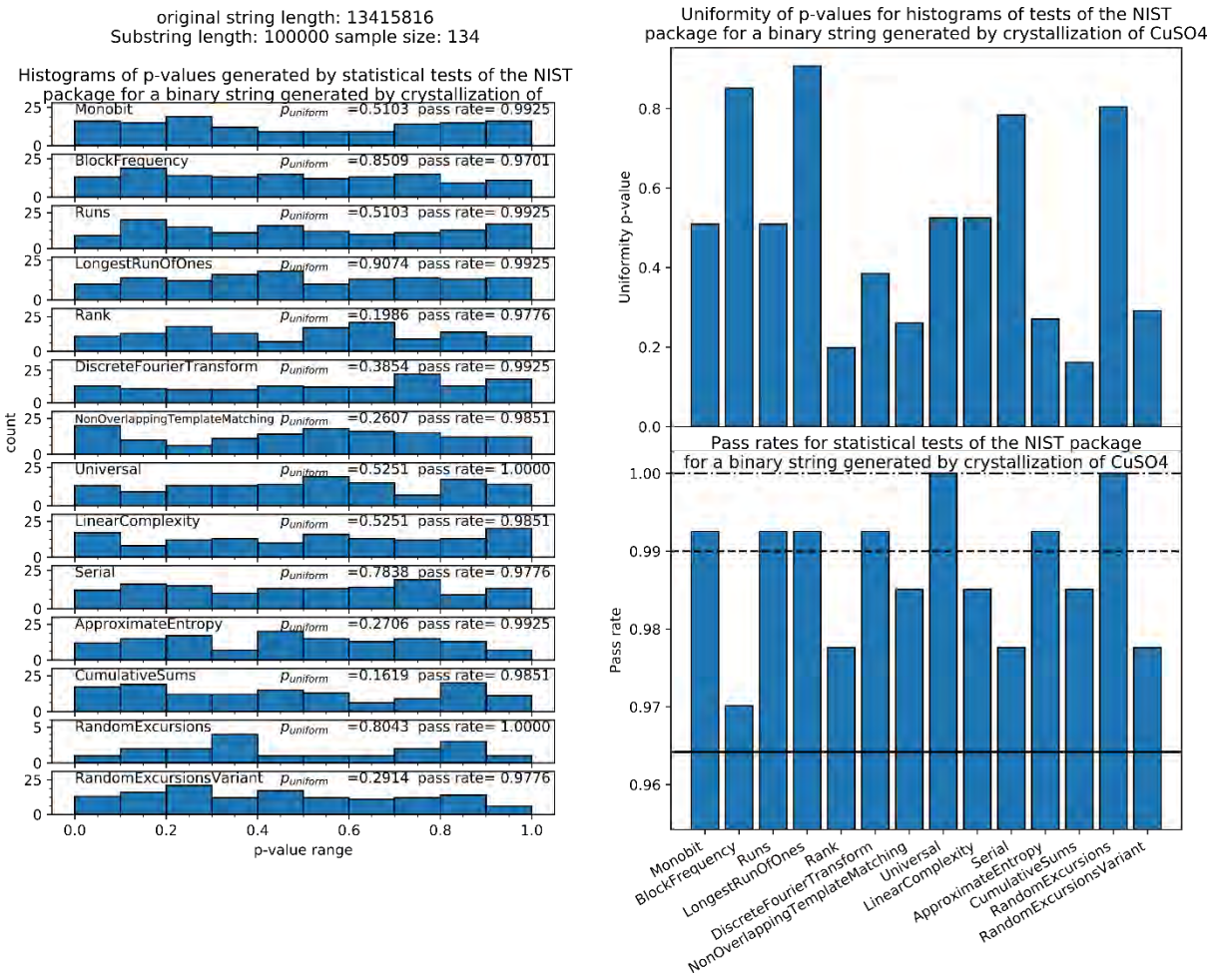
Supplementary Figure S12. Structure of $\{Co_4\}$ (*20*)



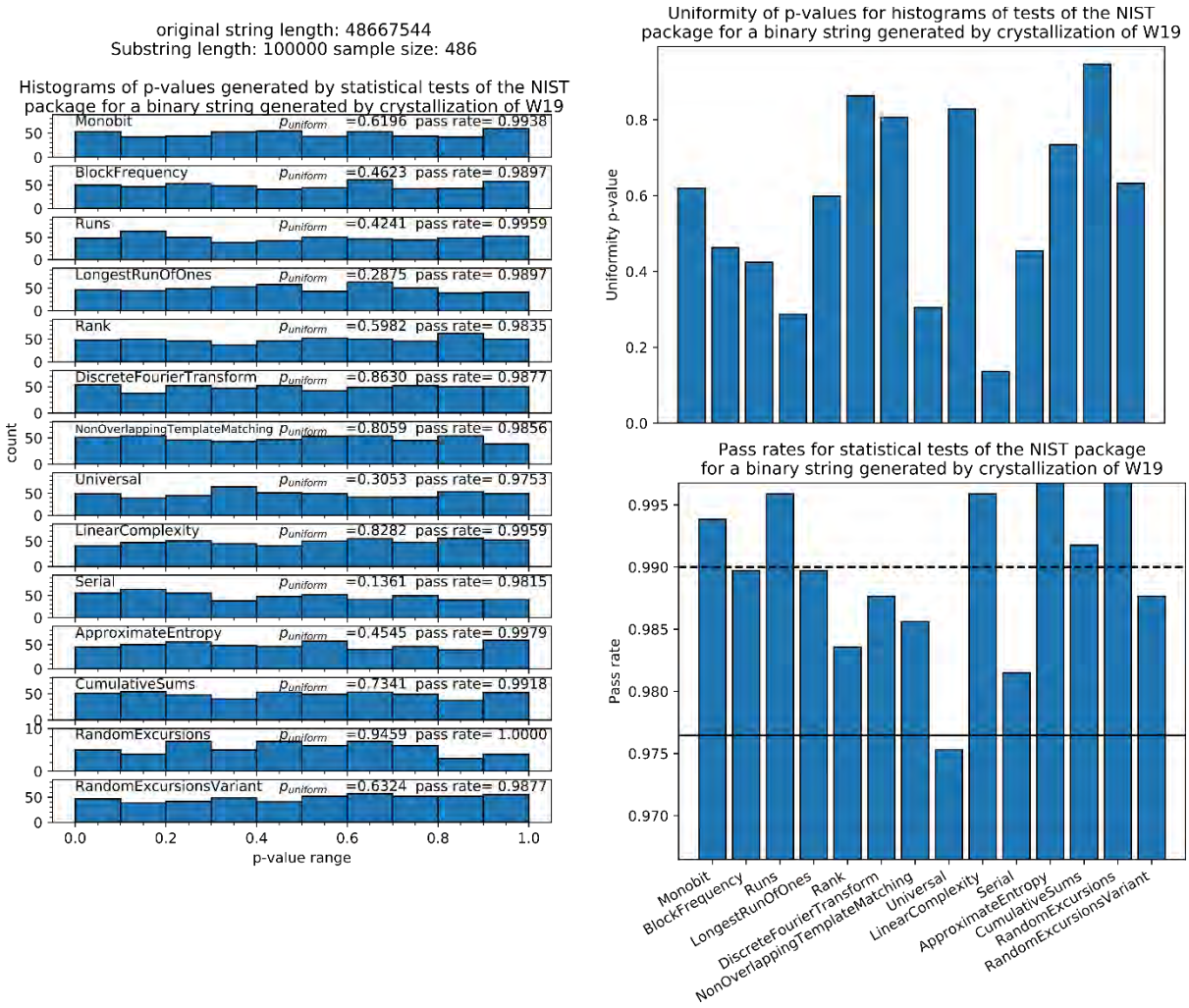
Supplementary Figure S13. Vial detection image processing method. Top left: original frame; Top right: colorspace conversion from bgr to hsv; Bottom left: separation of crystals from background by applying a pixel color threshold to the hsv image. Bottom right: Identification of large, central, circular contour as corresponding to the vial rim.



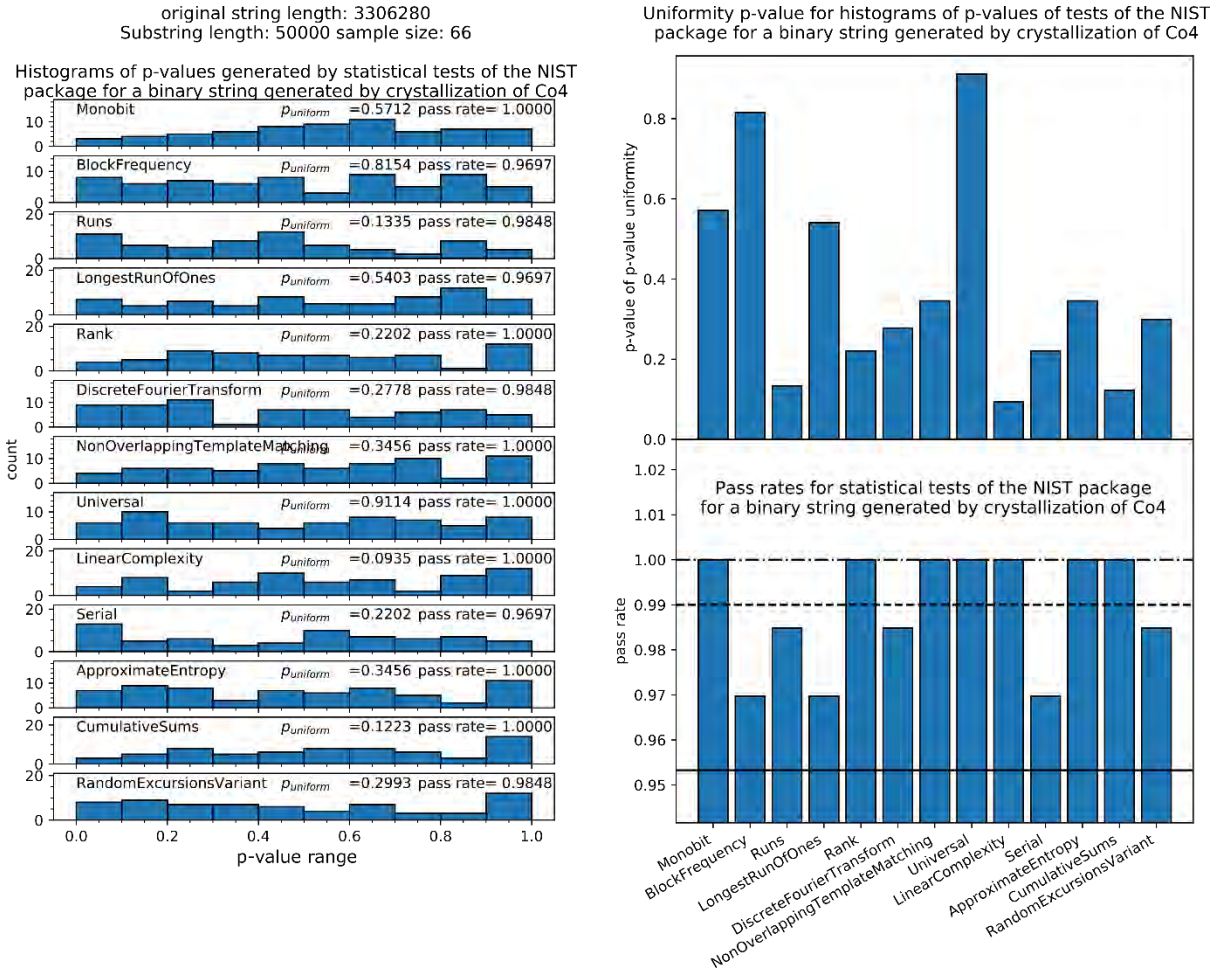
Supplementary Figure S14. Crystal detection image processing method. Top left: original image; top right: pixels detected as corresponding to crystal features by a model generated using Mask RCNN (*18*). Bottom: outline of the detect features superimposed on the original image. Although not all crystals were detected, the false positive rate was sufficiently low that it did not affect the randomness of the string produced.



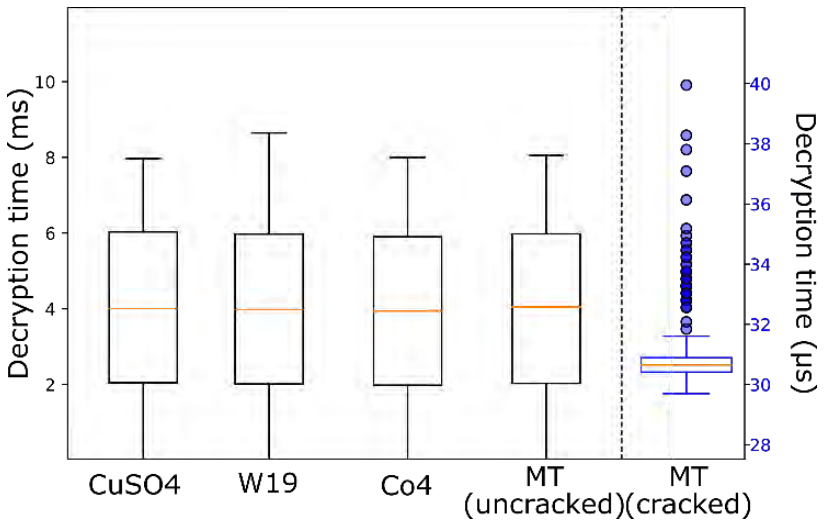
Supplementary Figure S15. Data for three levels of NIST random number testing (*7*) for crystallization of $CuSO_4$ after 2 hours. Left: histograms of p-values for each test; top right: uniformity p-value for each histogram. Each of these are well above the threshold of 0.0001 indicating a uniform distribution; bottom right: pass rate for each test. The dashed line indicates the expected pass rate with a significance level set at 0.01. The lower solid line indicates three standard deviations from the expected mean.



Supplementary Figure S16. Data for three levels of NIST random number testing (*7*) for crystallization of {W_{19}} after 2 hours. Left: histograms of p-values for each test; top right: uniformity p-value for each histogram. Each of these are well above the threshold of 0.0001 indicating a uniform distribution; bottom right: pass rate for each test. The dashed line indicates the expected pass rate with a significance level set at 0.01. The lower solid line indicates three standard deviations from the expected mean.



Supplementary Figure S17. Data for three levels of NIST random number testing (*7*) for crystallization of {Co_4} after 2 hours. Left: histograms of p-values for each test; top right: uniformity p-value for each histogram. Each of these are well above the threshold of 0.0001 indicating a uniform distribution; bottom right: pass rate for each test. The dashed line indicates the expected pass rate with a significance level set at 0.01. The lower solid line indicates three standard deviations from the expected mean.



Supplementary Figure S18. Boxplots of times for the message 'crystal' to be decrypted after encryption using 8-bit keys generated by different methods. Keys generated using the crystallization random number generator on average take as long to crack as those produced using the uncracked Mersenne Twister (MT). However, once the state of the MT is determined, the average decryption time for this method is substantially shorter than the crystallization method. (*5*)

Supplementary tables

Test name	Purpose
Monobit	**To assess whether overall ratio of 0s and 1s is as expected for a random bitstring.**
BlockFrequency	**To assess whether ratio of 0s and 1s in multiple substrings is as expected for a random bitstring.**
Runs	**To assess whether number of uninterrupted sequences of identical bits ('runs') is as expected for a random bitstring**
LongestRunOfOnes	**To assess whether longest uninterrupted sequence of 1s is as expected for a random bitstring**
Rank	**To assess whether the number of long-range repetitive patterns throughout its sequence is as expected for a random bitstring.**
DFT	**To assess whether the number of short-range repetitive patterns throughout its sequence is as expected for a random bitstring.**
NonOverlapping TemplateMatching	**To assess whether the number of sequences without showing repetition is as expected for a random bitstring**
Universal	**To assess whether the compressibility of the bitstring is as expected for a random bitstring**
LinearComplexity	**To assess whether the complexity of the bitstring is as expected for a random bitstring**
Serial	**To assess whether the frequency of different bit patterns of different lengths is as expected for a random bitstring**
Approximate Entropy	**To assess whether the frequency of different overlapping bit patterns of different lengths is as expected for a random bitstring**
Cusums	**To assess whether any maximum extent of the cumulative sum of the bitstring (treating 1 as +1 and 0 as -1) is as expected for a random bitstring**
RandomExcursions	**To assess whether the number of a particular cusum values is as expected for a random bitstring. Note that substrings in which the cusum crosses 0 less than 500 times are rejected.**
RandomExcursions Variant	**To assess whether the number of deviations from a particular cusum value is as expected for a random bitstring**

Supplementary Table S1. Summary of statistical tests applied to binary strings generated by the platform (*7*). Note that Overlapping Template Matching was omitted from the barrage due to insufficient bit-stream length in the Co_4 test to obtain significant results.

Supplementary Equations

$$\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}$$

Supplementary Equation 1. Determination of pass rate confidence interval where $\hat{p} = 1-\alpha$, α is the significance level and m is the sample size. (*7*)

$$\chi^2 = \sum_{i=1}^{10} \frac{(F_i - s/10)^2}{s/10}$$

Supplementary Equation 2. Calculation of histogram uniformity χ^2 value, where F_i is the number of p-values in sub-interval I and s is the sample size. (*7*)

$$p\ value_T = igamc(9/2, \chi^2/2)$$

Supplementary Equation 3. Calculation of histogram p-value where igamc is the complementary imcomplete gamma function. (*7*)