Analysis of Randomness of GAEN keys

April Sheeran, MCS

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science

Supervisor: Stephen Farrell

April 2024

Analysis of Randomness of GAEN keys

April Sheeran, Master of Science in Computer Science
University of Dublin, Trinity College, 2024

Supervisor: Stephen Farrell

...ABSTRACT...

Acknowledgments

Thank you Mum & Dad.

April Sheeran

University of Dublin, Trinity College

April 2024

Contents

List of Tables

List of Figures

Chapter 1

State of the Art

Introduction to chapter

1.1 Background

1.1.1 What are GAEN keys

GAEN Apps

Google and Apple developed the Google/Apple Exposure Notification (GAEN) system to facilitate contact tracing in response to the Covid-19 Pandemic. (Mention conventional contact tracing). Nations across the world used this technology to create contract tracing apps, for example Covid Tracker in Ireland and SwissCovid in Switzerland (Leith and Farrell (2021)).

The way the contact tracing works, if a user enables it, is as follows:

- Every 10-20 minutes the user's device will generate a random 128-bit key, referred to as a Temporary Exposure Key (TEK).
- The user's device will broadcast these keys using Bluetooth Low Energy (BLE).
- The user's device will listen and store the TEKs being broadcasted from other devices within a certain radius. These TEKs are stored locally on the device.

- If a user tests positive for Covid, they can log this into the app. The app will send the user's recent TEKs (around the 14 days) to a central server managed by the local health authority.
- Every approx. 2 hours, the user's device will download the TEKs from the central server.
- The app compares these downloaded TEKs to the TEKs stored locally on the device.
- If there is a match, this means that the user has potentially been exposed to Covid and the app will notify them.
- CITES

How Keys are Generated

Not sure if i will include this or not yet.

Analysis on Apps

There have been numerous studies done on contact tracing apps that use GAEN technology. Google and Apple acknowledge that keeping users' information private and secure is essential to the success of the contact tracing app and claim to have designed their system with this central to the design (Google (2020)). Why is privacy so important for this app? Don't want to share covid status etc.

The major privacy concerns of GAEN apps according to (Nguyen et al. (2022)) are the identification of users, tracking users or extracting the social graph of users. Contact tracing should aim to identify encounters rather than actual users, by doing so they should not leak any information that could be used to identify the user. Similarly, the data collected should not be able to be used to create the social graph of the user, the social connections and relationships of a user. Having this information could potentially result in the user being identified.

Security is also a requirement for a contact tracing app according to (Nguyen et al. (2022)). The system should be resilient to large scale data pollution attacks. These could be fake exposure claims, where users may falsely claim they have been exposed in order to get out of work or another obligation or in an attempt to damage the reputation and credibility of the contact tracing apps. Fake exposure injection, a relay attack, may send users false notifications of potential exposures. The attacker could do this by capturing the TEKs of some user and broadcasting them in another location, leading to people being falsely notified of exposure. This could result in panic among users and the population, putting further strain on the healthcare system by creating a demand for unnecessary tests. It could also damage the trust in the contact tracing apps as their accuracy would be no longer trusted.

(Nguyen et al. (2022)) assess the GAEN apps on a variety of requirements. In terms of effectiveness, GAEN has been found to be imprecise at determining the distances between user devices (do i need to reference an internal reference?). Its use of BLE means scanning of the user's surroundings for other devices can only happen with frequent pauses to save battery life of the device. Many factors like positioning of the device's antenna, obstacles in the way and orientation of the device affect the computation of the distance between devices and the errors are significant. GAEN fails to account for 'superspreaders' of the virus, an individual who is very contagious and infects a number of other people. GAEN also does not have any mechanism for dealing with asymptomatic individuals, people that are infected with the virus and are contagious but do not show symptoms. Unknowingly, these people spread the virus. These individuals are unlikely to get tested and therefore won't log their infection in the app, meaning those that come in contact with them will not be notified of a potential exposure. This significantly impacts the effectiveness of the contact tracing apps.

An investigation into the data shared by Europe's contact tracing apps that use GAEN (Leith and Farrell (2021)) discovered that a significant amount of data was being sent to Google servers. The android implementations of the GAEN systems use Google Play

Services to facilitate GAEN-based contact tracing. The user must enable Google Play Services. It was found that Google Play Services connects to Google servers approximately every 20 minutes, sending requests that include the handset IP address, location data and persistent identifiers to link requests coming from the same device. The data sent to Google in other types of requests also include phone IMEI, device hardware serial number, SIM serial number and IMSI, phone number, WiFi MAC address, user email and Android ID. While sharing data to backend servers is not in itself an intrusion of privacy, the ability to link this data to a real-world user is problematic. Given that the user's IP address is being sent to Google very frequently, this could be used as location tracking. It is possible to de-anonymise this location data and potentially identify the user. Given that the user must enable Google Play Services, and therefore this data sharing, to do contact tracing, this does raise a concern to the privacy of the user.

1.1.2 What is Randomness

In order to test for randomness/non-randomness we must first define what randomness is. A random bit sequence could be explained as the result of flipping an unbiased coin, with two sides 1 and 0, which has an equal chance of 50 percent of landing on side 1 or side 0. Each flip of the coin does not affect any future coin flips which means the flips are independent of each other. This unbiased coin can therefore be considered a perfect random bit stream generator as the appearances of 1s and 0s will be randomly and uniformly distributed. All elements in the sequence are independent of each other and future elements in the sequence cannot be predicted using previous elements. CITE NIST Special Publication 800-107 Revision 1, Recommendation for Applications Using Approved Hash Algorithms This simple example gives us an understanding of what it means for a set of keys to be random.

The keys must exhibit certain properties in order to be accepted as random. They should be independent meaning no previously generated keys affect a new key. Equally likely meaning that the probability of a 0 or 1 appearing at any point in the key is equal

to $\frac{1}{2}$. Scalable meaning that if the key is random, then any extracted subsequence is also random. Any indication of a dependency or bias within the data would indicate nonrandomness.

1.2 Literature Review

1.2.1 How to Test for randomness/ Analyses done

It is important to note that you can not say for certain whether something is random or not, you can only find evidence against non-randomness. It is not possible to give theoretical proof of randomness of a sequence. CITE (55555) Various statistical tests can be performed on the data in an attempt to compare and evaluate the data against a truly random sequence since the outcome when a statistical test is applied to a truly random sequence is known. (**?**).

A challenge when testing for randomness is that there is no agreed upon complete set of statistical tests to deem a sequence random (**?**). There is an infinite number of tests that you could run in order to find the presence or absence of a pattern or bias within the data. The existence of a pattern or bias within the data would indicate that it is non randomness.

Hypothesis testing

Statistical testing is used to test against a defined null hypothesis (h0). The null hypothesis in this case is that the keys being tested are random. The alternative hypothesis (h1) is that the keys are not random. The challenge here is to determine which of these hypotheses can be accepted CITE 10.1145/3447773 . For each statistical test run on the data, the result accepts or rejects the null hypothesis.

The following table shows the possible results on a hypothesis test: ¡table¿ CITE 98765

The above situations are somewhat unknown but some control can be gained by knowing the probability of each of the error situations. The probability of Error Type 1 is defined as a, the level of significance. CITE 10.1145/3447773. This value is typically 0.01, 0.05 or 0.10. The probability of Error Type 2 is defined as B, referred to as contrast power and is usually used as 1-B. CITE 10.1145/3447773. If the data is truly random, rejecting the null hypothesis, determining that the data is non-random, will occur a small percentage of the time. For example if a is 0.01, it would be expected that 1 sequence in 100 sequences is rejected. CITE 98765

In practice, p-values are used to reject or accept the null hypothesis. In the context of this project, a p-value can be defined as the probability that a key produced is less random than the keys previously tested, given the kind of non-randomness the test is assessing CITE 98765. NEED TO FIX. A p-value equal to 1 indicates that the data is perfectly random while a p-value equal to 0 indicates that the data is completely non random. If the p-value is greater than or equal to a, the null hypothesis is accepted and the data appears to be random. If the p-value is less than a, the null hypothesis is rejected and the data is deemed non random.

Maybe K-S stuff here or later on TALK ABOUT test suites and what they are, not specific to one

1.2.2 Analyses done

Intro something like many examples of randomness testing being used on cryptographic techniques/applications.

Given that encryption is essential for maintaining data security in cloud computing, (Mohamed et al. (2012)) performed randomness testing on eight modern encryption techniques, including AES, MARS and DES. They tested on two different platforms, desktop computer and Amazon EC2 Micro Instance. They evaluated the encryption techniques implemented as Pseudo Random Number Generators (PRNGs). They used the NIST Test Suite to perform the randomness testing. With a significance level of 0.01, any p-value

less than 0.01 meant that sequence was rejected. They found no strong evidence of any statistical non randomness across the 8 encryption algorithms however some differences were found between them on the two different platforms.

Statistical analysis has been run on an enhanced SDEx encryption method based on the SHA-512 hash function (Hłobaż (2020)). Using various tests like frequency, cumulative sums and runs, with a significance level of 0.01, it was found that this encryption algorithm was sufficiently random and passed the tests.

1.2.3 Examples of Randomness Failures

Chapter 2

Design

2.1 Chosen Tests

2.1.1 Test Suites

The NIST and Dieharder test suites were chosen to evaluate the GAEN keys.

(May not be used because it wants streams of random numbers as input) NIST is widely used in industry (Hłobaż (2020)) (Brosas et al. (2020)) (Mohamed et al. (2012)) and is accepted as a standard. It contains tests that are recommended by NIST for the evaluation of PRNGs used in cryptography.

Dieharder is another widely used battery of tests for randomness created by Robert G. Brown. It is an extension of the Diehard suite of tests created by George Marsgalia.

2.1.2 Other Tests

Some tests outside of the test suites were performed. These include chi-squared test, lagplot and hilbert curve. The counts of the numbers of 1s and 0s in each bit position were also recorded and this data was plotted to allow for quick inspection.

This is a non-exhaustive list of possible tests. There are endless tests that can be ran to build confidence that the keys are random, however it is never certain if the data is

Dieharder Test	**Descriptions**
Birthdays Test	Item 1
OPERM5 Test	Item 1
32x32 Binary Rank Test	Item 1
6x8 Binary Rank Test	Item 1
Bitstream Test	Item 1
OPSO	Item 1
DNA Test	Item 1
Count the 1s stream Test	Item 1
Count the 1s byte Test	Item 1
Parking Lot Test	Item 1
Minimum Distance 2d circle Test	Item 1
Minimum Distance 3d sphere Test	Item 1
Squeeze Test	Item 1
Runs Test	Item 1
Craps Test	Item 1
Tang and Marsgalia GCD Test	Item 1
STS Monobit Test	Item 1
STS runs Test	Item 1
STS serial Test	Item 1
RGB Bit Distribution Test	Item 1
RGB Generalised Minimum Distance Test	Item 1
RGB Permutations Test	Item 1
RGB Lagged Sum Test	Item 1
RGB Kolmogorov-Smirnov Test	Item 1
Byte Distribution	Item 1
DAB DCT	Item 1
DAB Fill Tree Test	Item 1
DAB Fill Tree 2 Test	Item 1
DAB Monobit 2 Test	Item 1

Table 2.1: Tests in Dieharder test suite

truly random or not. The tests done here detect deviations from randomness rather than prove randomness.

Lagplot "A lag plot checks whether a data set or time series is random or not. Random data should not exhibit any identifiable structure in the lag plot. Non-random structure in the lag plot indicates that the underlying data are not random" (NIST (2010))

Hilbert Curve why ??

Spectral Test (Discrete FFT) Note that this description is taken from the NIST

documentation http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf . The focus of this test is the peak heights in the Discrete Fourier Transform of the sequence. The purpose of this test is to detect periodic features (i.e., repetitive patterns that are near each other) in the tested sequence that would indicate a deviation from the assumption of randomness. The intention is to detect whether the number of peaks exceeding the 95per cent threshold is significantly different than 5per cent.

Chi-squared A chi-square test checks how many items you observed in a bin vs how many you expected to have in that bin. It does so by summing the squared deviations between observed and expected across all bins (expand)

Plot of Counts Counts number of 0s and 1s in each bit position. Should be 50-50 1s and 0s. A plot of the data quickly shows any potential biases.

2.2 Closely-Related Work

2.2.1 Aspect #1

2.2.2 Aspect #2

2.3 Summary

Summarize the chapter and present a comparison of the projects that you reviewed.

	Aspect #1	**Aspect #2**
Row 1	Item 1	Item 2
Row 2	Item 1	Item 2
Row 3	Item 1	Item 2
Row 4	Item 1	Item 2

Table 2.2: Caption that explains the table to the reader

Bibliography

Brosas, D. G., Sison, A. M., Hernandez, A. A., and Medina, R. P. (2020). Analysis of the randomness performance of the proposed stream cipher based cryptographic algorithm. In 2020 11th IEEE Control and System Graduate Research Colloquium (ICSGRC), pages 76–81.

Google, A. (2020).

Hłobaż, A. (2020). Statistical analysis of enhanced sdex encryption method based on sha-512 hash function. In 2020 29th International Conference on Computer Communications and Networks (ICCCN), pages 1–6.

Leith, D. J. and Farrell, S. (2021). Contact tracing app privacy: What data is shared by europe's gaen contact tracing apps. In IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, pages 1–10.

Mohamed, E. M., El-Etriby, S., and Abdul-kader, H. S. (2012). Randomness testing of modern encryption techniques in cloud environment. In 2012 8th International Conference on Informatics and Systems (INFOS), pages CC–1–CC–6.

Nguyen, T. D., Miettinen, M., Dmitrienko, A., Sadeghi, A.-R., and Visconti, I. (2022). Digital contact tracing solutions: Promises, pitfalls and challenges. IEEE Transactions on Emerging Topics in Computing, pages 1–12.

NIST (2010). Lagplot.