What3Words Geo Encryption: An Alternative Approach to Location based Encryption

Julian Dreyer and Ralf Tönjes
Faculty of Engineering and Computer Science, University of Applied Sciences Osnabrück, Osnabrück, Germany

Keywords: Identity based Encryption, Location Based Encryption, What3Words, GPS.

Abstract: Identity Based Encryption (IBE) is a steadily emerging field of research in the cryptographical domain. A special flavor of IBE called Location Based Encryption (LBE) includes a given location attribute to add additional access control to the encrypted entity. The main goal is to allow an entity to decrypt the ciphertext only and only if the correct location information is provided. This allows to control the access based on the position of the data user. Existing solutions for LBE make use of the conventional Global Positioning System (GPS). Though, conventional GPS solutions are known to be influenced by an artificially added error, resulting in inaccuracy of the location data. This will consequently require the LBE scheme to include a level of tolerance, as the GPS coordinates may slightly diverge between different points in time. In order to mitigate this problem by design, an alternative approach to LBE is proposed to add additional tolerance. The approach presented in this paper makes use of the What3Words location system, which offers the required tolerance for the decryption and thereby mitigating the problem of GPS inaccuracy. A following study then evaluates the real-world performance of the new encryption algorithm.

1 INTRODUCTION

This paper proposes a new method of implementing a Location Based Encryption (LBE) scheme, which represents a special variation of Identity Based Encryption (IBE). In general, IBE can be used to add an additional layer of access control, onto a previously encrypted plaintext. With the introduction of LBE, the access control is implemented by using the location of a receiver. The main goal is to restrict the ability to decrypt a ciphertext only to receivers at a designated position. Existing proposals make use of the Global Positioning System (GPS) and use the mechanism within a Digital Rights Management (DRM) Engine. A film production studio for instance, could make use of LBE to protect the digital versions of the produced films, thus enabling only mobile users to consume the content only at a specific location. Other forms of access control, e.g. in the domain of the Internet of Things (IoT) are possible.One inherent problem of the GPS is the artificial inaccuracy of the signal. Therefore, approaches are need to cope with the varying coordinates by introducing a method to tolerate inaccurate coordinates. This work is proposing a novel way of approaching the previously described problem by utilizing the What3Words localization service.

The rest of the paper is organised as follows: section 2 introduces various existing approaches of mitigating the inaccuracy of the GPS and implementing a LBE scheme. It also shows a study, which indicates the accuracy of modern GPS receivers in order to later evaluate the real-world usage potential of the newly proposed LBE scheme. Said scheme is briefly described as well as the What3Words service in section 3. In section 4, an experimental study is conducted to evaluate the real-world tolerance of the newly proposed LBE scheme. Afterwards, the results are evaluated in section 5 and compared to the results of the GPS receiver accuracy study, mentioned in section 2, thus indicating the real-world application potential of the encryption scheme.

2 RELATED WORK

The Location Based Encryption (LBE) is a widely researched field within the Identity Based Encryption domain. Thus, there are already proposals for efficient algorithms for encryption like the Location-Dependent Data Encryption Algorithm (LDEA), first proposed by Liao and Chao (Liao and Chao, 2008).

Dreyer, J. and Tönjes, R.
What3Words Geo Encryption: An Alternative Approach to Location based Encryption.
DOI: 10.5220/0009791904450452
In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security (IoTBDS 2020)*, pages 445-452
ISBN: 978-989-758-426-8

The algorithm makes use of a GPS coordinate, consisting of the latitude and longitude of the receiver. It ensures confidentiality by including well known and tested symmetric encryption algorithms like the Data Encryption Standard (DES), the Advanced Encryption Standard (AES) or the triple-DES and hashing algorithms like the Message Digest Algorithm 5 (MD5). Though, due to advanced security research by Xie et.al. (Xie et al., 2013), the MD5 hash algorithm is considered as not reliable anymore and more sophisticated ones like the Secure Hashing Algorithm (SHA) should be used. Additionally, the DES algorithm in its plain form is vulnerable to known-ciphertext attacks. Due to its key size of only 56 Bits (NIST, 1999), an attacker is able to perform a brute force attack on the encryption keys with at most 2^{56} tries (Biham and Shamir, 1991). With further investigative cryptanalysis methods described by Matsui (Matsui, 1993), the conventional DES cipher can be broken in up to 2^{29} tries. With modern and specialized hardware at hand, an attacker is able to decrypt a DES encrypted ciphertext within a feasible amount of time. Thus, the proposed usage of DES in its plain form is not recommended for modern encryption schemes. Liao and Chao also showed potential future research on their algorithm, including the replacement of the aforementioned cipher suites and concepts, thereby stressing the modularity of the LDEA algorithm itself (Liao and Chao, 2008).

Another approach to Location Based Encryption is given by Scott and Denning (Scott and Denning, 2003), which is focusing on the application of Geo Encryption on digital film licensing and DRM protection. The main goal is to mitigate the problem of unauthorized users being able to capture a film signal, duplicating it and sharing it illegally. Geo Encryption would allow film producers to lock their films with a GeoLock Key, which they would only distribute to cinemas or authorized users in general. Only these key holders will then be able to decrypt the film in this scenario. The distribution of the key would not be required, because the location of the cinema requesting access might be known to the film producer. The film can then be encrypted for only the cinema's coordinates. Thus, a preceding key exchange is not necessary. Scott and Denning propose a new geo encryption algorithm called ”GeoCodes GeoEncryption Algorithm” (Scott and Denning, 2003), which involves the recipient's location coordinates, as well as the velocity and time of the recipient. The algorithm also includes a Position Velocity Time (PVT) → GeoLock Mapping Function, which attempts to accommodate for the inherent error and inaccuracy of the GPS consumer signal. The function works by establishing a three-dimensional mapping of latitude, longitude and time onto a hexadecimal value. The underlying matrices can be scaled to accommodate for an arbitrarily accurate receiver. By including more dimensions into the mapping function, more entropy will be generated. In order to hinder potential attackers to extract the GeoLock values from the matrices, the use of a cryptographic hash function is recommended.

The civil consumer GPS signal is known to be influenced by an artificially added inaccuracy. Therefore, Wing et al. (Wing et al., 2005) conduced a field study to evaluate the positional accuracy of consumer GPS devices. They have set up different testing environments and chose six different GPS receivers by well-known manufacturers, which were then be examined regarding their accuracy. The devices themselves make use of the differential GPS algorithm (Van Sickle, 2015), that involves the difference between two positions, to further decrease the error of the GPS Signal. The study shows that the average error of the GPS receivers lies within a 4.1m range with a standard deviation of 1.47m, thus concluding that even on modern consumer GPS receivers an error of four to five meters is to be expected.

3 CONCEPT AND METHODS

The intention of this paper is to propose an extended form of the LDEA algorithm (Liao and Chao, 2008), which not only involves an improved cryptographic cipher suite, but also makes use of a different approach to the problem of GPS inaccuracy. The ”What3Words” service provided by the what3words Ltd. (what3words Ltd., 2019b) offers an alternative localization approach to the GPS. By including this service into the LDEA algorithm (Liao and Chao, 2008), an intrinsic theoretical tolerance of nine square meters can be achieved, thus solving the GPS inaccuracy problem of the LDEA algorithm in its plain variant. The given theoretical tolerance lies below the average error of consumer GPS receivers of 4.1m (Wing et al., 2005). Therefore, an experimental study will evaluate the real-world tolerance of the newly introduced What3Words mechanism. The following subchapters will describe the What3Words service, introduce the newly proposed LDEA* algorithm in detail and give an evaluation of its real-world application potential.

3.1 What3Words Service

The What3Words service works by dividing the earth's surface into three by three meter squares and

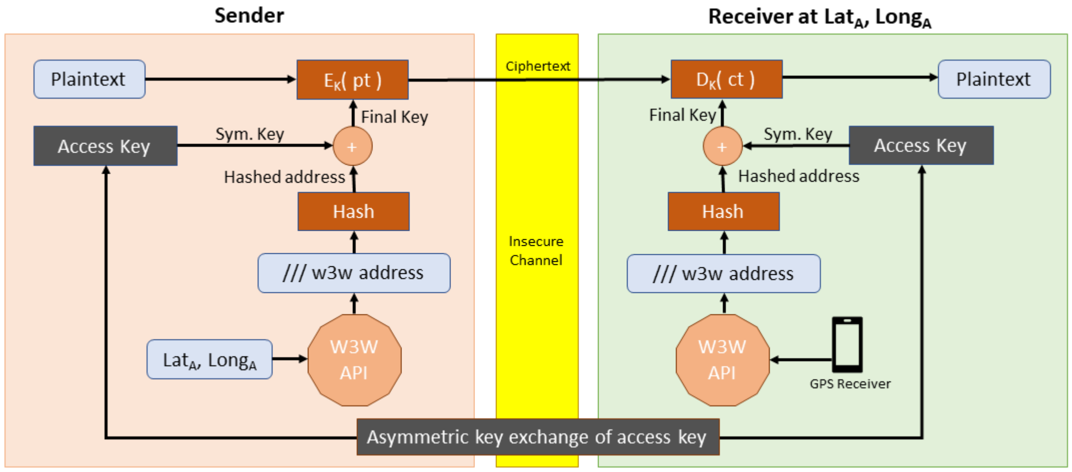

Figure 1: The LDEA* algorithm.

giving each of these squares a unique three word address. The address itself is separated by dots and involves human readable words from a dictionary (what3words, 2019), e.g. English. There exists a developer API capable of converting regular GPS coordinates into a unique What3Words address. A examplary What3Words address can look like the following:

/// filled.count.soap

In order for the user to easily identify an address in the What3Words format, three slashes are prepended to the address. The service is designed to simplify the use case of sharing a location with other people or with a navigation system. By making use of human readable and pronounceable words, unlike regular GPS coordinates, which make use of multi digit numbers, voice recognition systems can easily grasp the desired address. Additionally, the addresses are easier to remember as their structure is made easy to read for humans. Unlike the dot-separated words might suggest, the addresses themselves do not follow a hierarchical structure (Macgregor, 2019). This is due to the intended use case of automobile navigation systems and especially voice recognition systems. If the addresses would follow a hierarchical scheme rather than randomly assigned words like they do now, misinterpretations by wrong voice recognition could occur, leading to a wrong location being determined. This would impact the usability of the whole system. For this reason the developers decided on a random assignment of words for each address. Additionally, they focused on words that are easily pronounceable by humans (Day, 2019), which increases the overall usability of the system even more.

When using the service, a developer can access the freely available online API provided by what3words or even request an offline SDK. Though, the offline SDKs are only distributed to contractors who pay a monthly fee. The What3Words developers claim, that the offline SDK, which involves an offline version of the address database in conjunction with either a C++, Java or Mobile Library, only requires about 5 Megabytes of storage (what3words Ltd., 2019a). Thus, an offline version of the API would even be suitable for small IoT devices, which are not able to communicate with the online API directly and possess limited hardware resources.

Generally, the algorithm behind the addressing of individual squares is not transparent to the user. By the definition of the aforementioned concept, a theoretical tolerance of about nine square meters can be achieved (what3words, 2019). By introducing the What3Words service to the LDEA algorithm and thereby exchanging the GPS coordinates with a What3Words address, an intrinsic tolerance is established. Thus, the problem of the GPS inaccuracy is inherently resolved. Given an IoT device, which includes a low-cost GPS receiver and online capabilities, the What3Words service would be able to cope with the inaccuracy of the GPS receiver and offering a LBE encryption option.

3.2 LDEA* Algorithm

The LDEA* algorithm represents an encryption scheme, that allows arbitrary data to be symmetrically encrypted and location locked at the same time. Due to its underlying similarities with the LDEA algorithm first described in (Liao and Chao, 2008), an asterisk has been added, indicating its new functionality in combination with updated crypto algorithms.

The LDEA* algorithm, the process shown in Fig-

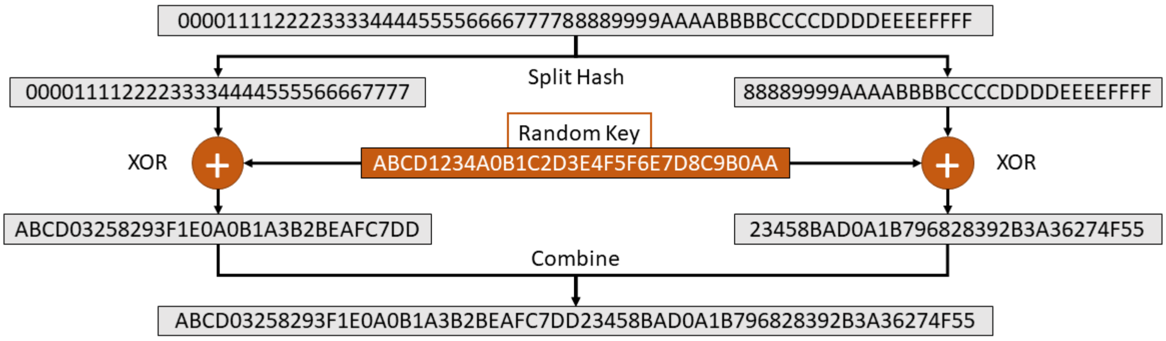
0000111122223333444455556666777788889999AAAABBBBCCCCDDDDEEEEFFFF
Split Hash
00001111222233334444555566667777
88889999AAAABBBBCCCCDDDDEEEEFFFF
Random Key
XOR
ABCD1234A0B1C2D3E4F5F6E7D8C9B0AA
XOR
ABCD03258293F1E0A0B1A3B2BEAFC7DD
23458BAD0A1B796828392B3A36274F55
Combine
ABCD03258293F1E0A0B1A3B2BEAFC7DD23458BAD0A1B796828392B3A36274F55

Figure 2: Split-Hash-Combine method.

ure 1 will be executed. At first, the sender needs to determine the location of the receiver, who is intended be able to decrypt the message later on. Once the sender knows the receiver coordinates (Lat_A, $Long_A$), these will be sent to the What3Words API in order to convert them into a What3Words address. As there exist either the offline SDK or online API of the What3Words service, the implementation of the What3Words component may vary accordingly. Due to varying lengths of the addresses, a hash function with a defined output length (e.g. 64 Bytes using the SHA-512 hash algorithm) is being applied onto the What3Words address. The resulting hashed address is then XORed with a randomly generated symmetric key. Following a similar scheme like the original LDEA key generation by Liao and Chao (Liao and Chao, 2008), the randomly generated key is of half the size of the hashed address. In order to XOR both parts, the hashed address is split in half. Both resulting halfes of the hashed address then are XORed separately with the randomly generated key. The final key is then produced by combining the keys again to regain the original length of the hashed address. Notice that the length of the randomly generated key needs to be half of the hashed address length. This whole method is shown in Figure 2.

Once the final key is generated, it is supplied to the symmetric block cipher encryption algorithm. Here, the AES-256 in Galois/Counter Mode (GCM) mode was chosen due to its efficiency and it being the standard symmetric encryption algorithm standardized by the National Institute for Standards and Technology (NIST) (Farooq and Aslam, 2017). In its plain variant, the AES block cipher only ensures confidentiality but no authenticity. Therefore, a modern Authenticated Encryption with Associated Data (AEAD) mode like GCM is used. The resulting ciphertext does not only contain the encrypted plaintext of the sender, but also a set of unencrypted associated data items. The associated data ensures the authenticity of the encrypted data, because it is being included within a Message Authentication Code (MAC). Therefore, the integrity and authenticity of the received data can be directly evaluated. Thus, the overall security requirements for a secure and authentic data exchange are satisfied.

The most notable disadvantage of symmetric block ciphers is its intrinsic demand for both parties to share the same key for both de- and encryption. In general, the assumption of both parties knowing each other is often not valid. Thus, an asymmetric key exchange is needed beforehand. ”Hybrid Encryption” utilizes asymmetric encryption to share a symmetric block cipher key. The LDEA* algorithm recommends to share the randomly generated key either by Diffie Hellman Ephemeral (DHE) or RSA key exchange (Rountree, 2011). It is to be noted that this randomly generated key alone is not enough to decrypt the cipher-text. The receiver also needs to perform the key derivation process described in Figure 2.

For the decryption, the receiver needs to generate the access key with his own What3Words address.In order to generate the final key, the receiver makes use of his GPS coordinates, which are determined with a tamper-proof GPS receiver. The resulting coordinates (Lat_A, $Long_A$) are passed to the What3Words API, which returns a valid What3Words address. Then the address undergoes the Split-Hash-Combine method (s. Figure 2) resulting in the final key for the decryption. With the AES encryption algorithm in GCM mode, a receiver can also determine, whether a ciphertext was correctly deciphered by checking the hash of the resulting plaintext with the associated data. Thus, the receiver can immediately determine, whether the message was intended to be decrypted at his position, assuming the symmetric key was correctly shared beforehand. This allows a sender to broadcast data to multiple receivers, while only one of them being able to decrypt the data, assuming the all receivers hold the same symmetric key. The overall method of mapping the GPS coordinates into a predefined rectangle has also been described by Scott and Denning (Scott and Denning, 2003).

Though their method requires a more computationally intense workflow in order to generate the GeoLock Keys, making it infeasible for small scale devices with restricted hardware resources. This problem is mitigated by the unique mapping of the What3Words service intrinsically. Considering the results of Wing et al. (Wing et al., 2005), a tolerance level of three meters, which is theoretically given by the LDEA* algorithm, is below the expected accuracy range of consumer GPS equipment, making it suitable for small scale devices. As stated by What3Words, the size of the offline SDK version of the What3Words service only packs 5 Megabytes in size, thus making it feasible to store it on the local memory of the device. Even with limited hardware resources, this approach can be implemented efficiently.

3.3 Potential Issues

The LDEA* algorithm offers a theoretical tolerance of $\pm 1.5m$ for a consumer GPS receiver with expectable inaccuracies. This theoretical tolerance level is only fitting for the best-case assumption, which would resemble a GPS point, that is located in the center of the square on the What3Words world map. The GPS coordinates may shift arbitrarily in any direction with the same amount of tolerance. This can be seen in the first square of Figure 3.

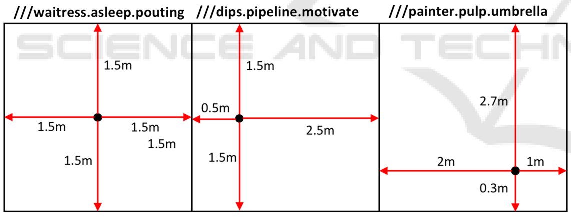

Figure 3: Varying tolerances due to different point locations.

In general, it cannot be assumed that the point lies within the center but rather at a uniformly distributed location within the square. Thus, different locations of the point within a given square are possible and most likely. In a real-world scenario, a potential receiver, who's location lies on the edge of a square, might be unable to decrypt the message due to a square change, if the GPS receiver has inaccuracies. Considering Figure 3 again, this would be the case if the receiver is on the marked point within the second square and owns a GPS receiver with an inaccuracy rating of up to one meter. If a sender would encrypt a message for the address ”///dips.pipeline.motivate” and the receiver would determine his location with one meter inaccuracy to the left, the What3Words API could return the address ”///waitress.asleep.pouting”, thus leading to the receiver being unable to decrypt the message. Though, the distribution of tolerances within a given square is always three meters, considering the designed dimensions of the squares within the What3Words world map. Regarding these assumptions, an experimental study will be conducted to identify the real-world tolerance of the LDEA* algorithm. With the design principles at hand, one can hypothesize that the LDEA* algorithm will not exceed the tolerance level of three meters.

3.4 Proof of Concept

In order to leverage an experimental study to evaluate the capabilities and the tolerances, the LDEA* algorithm is implemented in a high level programming language. The fundamental use case for the experimental study involves two parties. The sender produces arbitrary data which gets encrypted using the LDEA* algorithm. Consequently, the receiver's GPS coordinates are needed by the sender, to successfully encrypt the data for the desired receiver. These get provided on runtime. The data itself is being sent via MQTT as the underlying communication protocol. The receiver then attempts to decrypt the data, using the LDEA* algorithm with the employment of his own GPS coordinates. If the access key, as well as the location of the receiver are correct, the data is decrypted successfully by the receiver.

In order to facilitate the asymmetric key exchange between the two parties over MQTT, which itself does not encrypt any data (Singh et al., 2015), RSA was chosen. The receiver and the sender both own a private and public key pair. The sender generates a new, random symmetric key which will then be encrypted with RSA and the previously broadcasted public key of the receiver. The sender then transmits the encrypted symmetric key over the insecure network to the receiver who will be able to decrypt the symmetric key with his own private RSA key. The key exchange process is then finalized and the LDEA* algorithm itself can be executed.

4 EXPERIMENTAL STUDY

This chapter examines the real-world performance of the LDEA* algorithm. The hypothesized outcome of a tolerance level of three meters will therefore be validated.

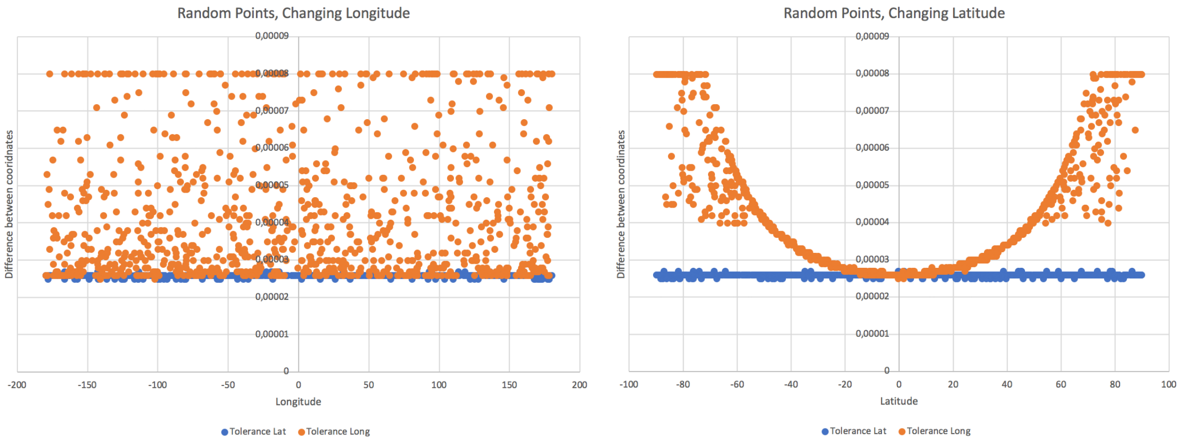

Figure 4: The resulting tolerance in latitude and longitude.

4.1 Methodology

In order to get a measurement for the tolerance level of a given point, a custom testing method has been implemented. The tolerance in this case is defined as the distance between the opposite edges within a square, either in vertical or horizontal direction. Therefore, randomly chosen points on the world map undergo a custom testing methodology. Each point $\mathrm{P_x}$ possesses a unique GPS coordinate pair ($\mathrm{Lat_x}$, $\mathrm{Long_x}$). For most consumer GPS receivers, the accuracy of the floating point numbers do not exceed $1.0 * 10^{-7}$ (Wing et al., 2005). Therefore, this scale can be set as the step size for the following algorithm:

1. A new random point $\mathrm{P_x}$ at ($\mathrm{Lat_x}$, $\mathrm{Long_x}$) is being generated from a uniform distribution.
2. The point is shared amongst sender and receiver. Both set the point as their own location for decryption and encryption.
3. The sender encrypts the points latitude and longitude with the LDEA* algorithm and sends them to the receiver. A decryption should be successful, as the receiver knows the position of the point.
4. The sender subtracts an amount of $1.0 * 10^{-7}$ from the coordinate and encrypts the new data for the resulting coordinate. He then sends the message to the receiver for decryption.
5. The sender repeats the previous step for each direction until a $\mathrm{min_{Lat}}$, $\mathrm{min_{Long}}$, $\mathrm{max_{Lat}}$ and $\mathrm{max_{Long}}$ value are reached. At some point, the receiver will not be able to decrypt the ciphertext anymore, thus marking the minimum or maximum value of the respective coordinate.
6. The distance in meters between the minimum and maximum coordinates of the latitude and longitude represent the tolerance and can be determined by utilizing the following formula (Sheppard and Soule, 1922):

$$\begin{aligned} \Delta Lat &= Lat_y - Lat_x \\ \Delta Long &= Long_y - Long_x \\ tmp_{lat} &= \frac{(Lat_x + Lat_y)}{2} * \frac{\pi}{180} \\ dx &= 111.3 * cos(tmp_{lat}) * (\Delta Long) \\ dy &= 111.3 * (\Delta Lat) \\ distance &= \sqrt{dx^2 + dy^2} * 1000[m] \end{aligned} \tag{1}$$

This testing algorithm yields a definitive metric for the tolerance within a given square in meters. Thus, this algorithm assigns any given random point a tolerance value which can be evaluated afterwards. The value itself represents the mean tolerance within the square surrounding the given point. Given the previous hypothesis, the expected value is approximately three meters.

For this particular evaluation, a dataset of 2000 sample points was generated with $lat_x \in]-90,90]$ and $long_x \in]-180,180]$ using a uniform distribution. Thus, a large scale of the earth surface is evaluated. The resulting latitude and longitude tolerances are shown in Figure 4.

4.2 Results

The study yields results which exceed the previously hypothesized tolerance of three meters by about a meter difference. The resulting mean tolerance of the study is 3.9302m. To further investigate the results of the study, the relation between the change in latitude and longitude of the examined point is of particular interest. The first diagram of Figure 4 assumes a fixed latitude and a changing longitude value. A change in longitude will, apart from error noise, tend to a mean longitude difference value of $3*10^{-4}$. No regular pattern can be extracted from the the first Diagram of 4. Though, the difference in latitude remains steadily at $2.6*10^{-4}$.

Regarding the second diagram of Figure 4, a significant parabola shaped pattern results. When altering the latitude of the random points and fixing the longitude, the resulting longitude difference describes a parabola pattern, while the latitude difference stays at $2.6*10^{-4}$. This relation can be shown in a three-dimensional plot to further interpret the results. By utilizing formula 1, which gives a distance between two points, the effective tolerance in meters can be visualized.

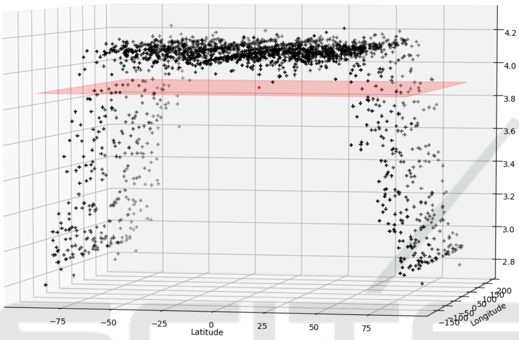

Figure 5: A 3D plot of the tolerance.

The trapezoid shaped figure in Figure 5 shows the relation between the change in the point's coordinates and the resulting tolerance in meters. A significant decrease in tolerance for latitude values with an increasing absolute value exists within the data. For longitude values $\in [-71,71]$ degrees, the tolerance is tending towards the previously mentioned mean value of 3.9302m and even exceeding it. This behavior is contradictory to the statement of the What3Words service, which ensures a tolerance of theoretically three meters. As for latitude values $\notin [-71,71]$, the tolerance falls below three meters, thus contradicting the statement. The expected behavior for the outcome in regards to the hypothesis would have been a steady plane in the 3D space on the three meter level. Though the three-dimensional trapezoid hints towards a different aspect, which has not been considered yet.

On the one hand, the upper and lower latitudes of the earth are significantly smaller concentric circles than the latitudes further towards the equator, thus resulting in different circumferences. On the other hand, the longitudes all possess the same length, reaching from the north to the south pole of the earth.

In order to fill the surface of the globe with squares, like the What3Words service does, a uniform distribution of the squares is required. The claims of the service, in regards to the three-by-three meter squares are thereby not feasible. A grid of equally sized squares cannot be fitted onto a spherical surface like the earth. So, in order to keep the organized grid form, the shape of the squares needs to be morphed. Thus, the upper and lower lengths or the square get shrunk down to adapt to the shrinking circumference of the latitudes. Therefore, the "squares" should not be considered as squares at that point but rather as trapezoids. Therefore, the results of Figure 5 are plausible. With rising absolute value of the latitudes towards the poles, the side lengths of the trapezoid consequently shrink. The tolerances behave accordingly, as there are lower tolerance bounds for the trapezoid itself. When lowering the absolute value of the latitude, thereby moving towards the equator, the tolerances get bigger as the trapezoid's side lengths and consequently its area increase.

Within the interval of [-71,71] degrees of latitude, no significant increase in circumference is appearing as the curvature of the sphere within this range is not as steep in comparison to the outer range of the interval. Thus, the tolerance almost stagnates within this interval, as can be seen in Figure 4. Nevertheless, the average value amongst all the measured tolerances is well above the hypothesized three meters, thereby contradicting the previously mentioned hypothesis. The LDEA* algorithm therefore allows a larger tolerance of 3.9302m. With more samples at hand, the mean value may tend towards four meters, as the interval of $\in [-71,71]$ degrees includes more values above four meters than the outer intervals which mainly consist of tolerance values in the interval $]2.7,3.5]$ m.

4.3 Security Considerations

Regarding the security of the LDEA* algorithm, a potential attacker only needs to gather information about the location of the receiver with a tolerance of 1.9651m. Due to the XOR operation of the symmetric access key and the What3Words hash (s. Figure 2), an attacker might be able to get hold of the secret access key. When conducting a ciphertext only attack on the algorithm, the computational effort would be analogous to the effort of breaking the AES-256 algorithm, thus making it infeasible.

If an attacker would get hold of the randomly generated key, either by intercepting the key exchange or rather by extracting the key from a legitimate sensor node within the network, he would only need to break the What3Words hash. Theoretically, the attacker needs to hash all $57*10^{12}$ possible addresses and XOR them with the symmetric key. Though, the search space can be reduced by a large factor with additional knowledge about the location of the receiver, e.g. the country or state. Also, an attacker might know even more about the node, e.g. if it is located within a building thereby excluding rural areas. So, by educated guessing, a large scale of irrelevant addresses can be extracted from the set of potential addresses. This is all assuming that the attacker already knows the randomly generated key. To hinder an attacker from getting the symmetric key for all the participants in the network, it is recommended that every partner in the network exchanges a unique key with the sender. This leads to a more complex communication scheme but benefits the overall security of the algorithm.

5 CONCLUSION

The experiments conducted in this paper showed, that the overall mean tolerance of the LDEA* algorithm lies well above the hypothesized three meters of tolerance. This is due to the underlying grid structure of the What3Words world map. The varying sizes of the ”squares” lead to a divergence of tolerance within the most northern or southern regions of the earth.

With regards to the LDEA* algorithm, the tolerance leads to a larger area of decryption. A potential sender may very well assume, that a receiver will be able to decrypt the ciphertext, if encrypted with the LDEA* algorithm. This assumption only holds, when the GPS measurement equipment on the receiver side is accurate enough to fall within the region of $\frac{3.9302m}{2} = 1.9651$m, as the tolerance applies to the whole ”square”. Modern GPS receivers are well capable of achieving this level of accuracy by utilizing the differential GPS technology, described in (Van Sickle, 2015). This allows a GPS receiver to calculate its position in difference to another fixed point, thus eliminating some of the artificial inaccuracies of the receivers.

With the LDEA* algorithm at hand, e.g. an IoT sensor network is able to efficiently encrypt the data shared between the communication partners, who operate in a publish-subscribe scheme. Future work may focus on the concrete implementation of a tamper-proof GPS receiver, since this aspect is assumed as given in this paper. Existing proposals by Pozzobon et al.(Pozzobon et al., 2011) already give a theoretical approach but lack an implementation of the system itself.

REFERENCES

Biham, E. and Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72.

Day, C. (2019). What about swear words? Available: https://intercom.help/what3words/en/articles/2212867-what-about-swear-words.

Farooq, U. and Aslam, M. F. (2017). Comparative analysis of different AES implementation techniques for efficient resource usage and better performance of an FPGA. *Journal of King Saud University-Computer and Information Sciences*, 29(3):295–302.

Liao, H.-C. and Chao, Y.-H. (2008). A new data encryption algorithm based on the location of mobile users. *Information Technology Journal*, 7(1):63–69.

Macgregor, M. (2019). How are the words assigned? Available: https://support.what3words.com/en/articles/3577589-how-are-the-words-assigned.

Matsui, M. (1993). Linear cryptanalysis method for DES cipher. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 386–397. Springer.

NIST (1999). Data encryption standard. *Federal Information Processing Standards Publication*.

Pozzobon, O., Wullems, C., and Detratti, M. (2011). Tamper resistance. *GPS World*. Available: https://www.gpsworld.com/transportationtamper-resistance-11403/.

Rountree, D. (2011). 2 - cryptography. In Rountree, D., editor, *Security for Microsoft Windows System Administrators*, pages 29 – 69. Syngress, Boston.

Scott, L. and Denning, D. E. (2003). Location based encryption & its role in digital cinema distribution. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA.

Sheppard, W. W. and Soule, C. C. (1922). *Practical navigation*. World Technical Institute.

Singh, M., Rajan, M., Shivraj, V., and Balamuralidhar, P. (2015). Secure MQTT for Internet of Things (IoT). In *2015 Fifth International Conference on Communication Systems and Network Technologies.*, pages 746–751. IEEE.

Van Sickle, J. (2015). *GPS for land surveyors*. CRC Press, Boca Raton, FL, USA.

what3words, A. (2019). Available: https://what3words.com/about-us/.

what3words Ltd. (2019a). What3Words Enterprise Suite. Available: https://what3words.com/products/enterprise-suite/.

what3words Ltd. (2019b). What3Words Homepage. Available: https://what3words.com.

Wing, M. G., Eklund, A., and Kellogg, L. D. (2005). Consumer-grade global positioning system (GPS) accuracy and reliability. *Journal of forestry*, 103(4):169–173.

Xie, T., Liu, F., and Feng, D. (2013). Fast Collision Attack on MD5. *IACR Cryptology ePrint Archive*, 2013:170.