iMAC: Implicit Message Authentication Code for IoT Devices

Ikram Ullah
University of Twente
Enschede, The Netherlands
i.ullah@utwente.nl

Nirvana Meratnia
University of Twente
Enschede, The Netherlands
n.meratnia@utwente.nl

Paul J. M. Havinga
University of Twente
Enschede, The Netherlands
p.j.m.havinga@utwente.nl

Abstract—The role of cryptography is not only limited to enabling secure communication but also providing mechanisms for authentication, integrity, privacy, trust and non-repudiation. All these mechanisms play vital role in modern digital communications. Currently, number of interconnected IoT devices has exceeded over billions, which makes managing them not only challenging but also impractical. Central management of IoT devices also introduces serious security threats such as central point of failure and lack of transparency, confidentiality, and integrity. Lightweight and decentralized device-to-device schemes are important to cope with the ever increasing number of resource constrained IoT devices. Implicit cryptographic schemes, in which only devices in the same context can authenticate and communicate, is an efficient way to achieve decentralized and transparent IoT security. In this paper, we propose a multi-sensor-based lightweight implicit message authentication mechanism. The proposed scheme is efficient and secure.

Index Terms—Internet of things, Implicit authentication, Smart logistics, Decentralization.

I. INTRODUCTION

Smart logistics is a revolutionary vision to reshape traditional logistics architecture into a smarter, intelligent and renewable architecture by incorporating intelligent technologies such as Internet of Things (IoT) in the existing logistics procedures. These technologies create values in logistics via IoT enabled capabilities like monitoring, optimizing, learning, and automation. IoT devices measure, store, and share data which may contain personal or business related information. Preserving and securing information in this context is important [16] and should be taken seriously. Traditional authentication mechanisms are significantly complicated, vulnerable, and not scalable [8] [6] [9]. Distributed ledger technologies (DLTs) such as blockchain [1] and IoTA [2] can play crucial role in enabling decentralized, secure and immutable storage, sharing, and processing of verified data [1]. Nonetheless, IoT devices are resource constrained and lack computation capabilities for execution of DLT or conventional cryptographic mechanisms, therefore are vulnerable to active and passive attacks [6]. In contrast,

implicit authentication which is based on being in the same context, for instance having similar biometric, behaviour, or features, has several advantages over conventional explicit authentication mechanisms. Implicit mechanisms are hard to clone and generally do not require out-of-band channel for secret key sharing especially in symmetric schemes [7]. They are also scalable, suitable for distributed applications, and privacy-aware because they do not require third party interventions. In this paper, we propose an implicit multi-sensor-based device-to-device message authentication code (MAC). It is used to authenticate messages between two or more devices and to ensure the messages are not modified, deleted, or replayed during the transmission from the sender to the receiver. Our proposal is based on combination of a number of simple, yet robust cryptographic techniques, non-linear polygraphic substitution, and transposition. Our non-linear polygraphic mechanism is different from the standard Hill cipher [3]. The proposed polygraphic mechanism is non-linear and based on non-invertible matrices and no constraints on the entries of key matrix and the output is not modulo N . Cryptographic schemes based on invertible matrices are frequently proposed. However, the idea of using non-invertible matrices is superior, because the number of non-invertible matrices are higher than invertible matrices as shown in Figure 1 and non-invertibility feature of the key matrix makes it hard to inverse the key. The equations to calculate key space of polygraphic substitution cipher is given in [4]. The number of $(d \times d)$ invertible matrices over Z_p for a prime p is

$$|GL(d, Z_p)| = \prod_{i=0}^{d-1} (p^d - p^i)$$

The number of $(d \times d)$ invertible matrices over Z_{p^n} for a prime p and a natural number n is

$$|GL(d, Z_{p^n})| = p^{(n-1)d^2} \prod_{i=0}^{d-1} (p^d - p^i)$$

The number of $(d \times d)$ invertible matrices over Z_m for any integer $m \geq 2$ is

$$|GL(d, Z_m)| = \prod_i (p_i^{(n_i-1)d^2} \prod_{k=0}^{d-1} (p_i^d - p_i^k))$$

Our key space includes data from sensors and random bits from the business rules. Along with sensor data, business rules are used as additional source of randomization to make the mechanism more secure. In the proposed mechanism, only devices having similar sensor data and business rules can communicate and without requiring a central server. The mechanism is scalable and distributed.

Figure 1. The total number of invertible (5×5) and non-invertible (5×5) matrices. Compared with invertible matrices, the number of non-invertible matrices are much higher.

CONTRIBUTION

We propose an implicit multi-sensor-based device-to-device message authentication mechanism. The unique characteristics of the proposed mechanism are: lightweight, stateless, and distributed. Existing MAC schemes lack efficient shared key establishment mechanism. The proposed mechanism, allows establishing secret key more efficiently and in a distributed manner. Furthermore, non-linear polygraphic substitution based on non-invertible matrices is proposed.

II. RELATED WORK

Implicit cryptographic mechanisms are present in the literature specifically for smart phones. Lee & Lee [8] proposed a smart-phone based implicit authentication mechanism to authenticate current user. It uses data from multiple sensors of smartphone to learn the user behavior patterns and environmental characteristics. Limitations of this approach is that user behavior needs to be learnt, which might not be possible on resource constrained IoT devices. Shi et al. [9] proposed an implicit authentication based on user's daily routine activities to generate user profile. Authentication score was calculated per recent user activities. This mechanism is dedicated to smart phones and computers because these devices have the capabilities to collect and process data to generate profiles using machine learning techniques. Jakobsson et al. [7] proposed an implicit authentication protocol, which can be used

as second factor authentication in mobile devices, based on past behaviour. Mayrhofer & Gellersen [5] proposed implicit authentication protocols "shake well before use" based on accelerometer data. The proposed authentication is suitable for mobile devices and not suitable for a large network of IoT devices. Other form of implicit authentications mechanisms are based on biometric gait recognition [10], location-based [11], and device usage pattern such as typing [12]. To summarize, existing solutions are either unsuitable for constrained devices, centralized, or only used as second factor authentication requiring to learn user behaviors using computationally intensive machine learning algorithms.

III. METHODOLOGY

Generally, single cryptographic schemes are vulnerable. Combing different schemes may provide better abstraction and diffusion. Therefore, we integrate variant of non-linear polygraphic substitution, simple columnar transposition, and XoR to calculate message authentication code (MAC). Furthermore, we utilize sensors (pressure, temperature, RSSI) data and business rules for the purpose of generating secret keys, hence implicit MAC. Our proposed algorithm can be applied to message of any size to generate a fixed length of MAC. ASCII [14] decimal encoding is used for representation. The proposed MAC contains three steps: secret key generation, message signing and verification.

1) *Secret key generation*: Our key space is based on random bits from business rules (R_b) and sensor data (pressure, temperature, and RSSI). To generate the keys, the following steps are performed:

a) *Secret keys extraction from business rules*: Instead of using complex cryptographic functions, we increase the security by extracting randomness (R_b) from the business rules. Businesses often employ certain business rules. These rules define how to execute business processes and to make decision in order to achieve the business goals. In general these rules can be implemented at three different levels, i.e, centrally (server), gateway, or device level. It is common to implement business rules at a server or a gateway level. In our proposed scheme, the business rules are required to be private and secure. The technique used to extract random bits R_b from business rules is presented in Algorithm 1.1. In the first iteration, the first and the last characters from every line is selected. In the second iteration, the second and the second last characters from every line are selected and so on. In order to generate MAC for message M , $R_{b_1}, R_{b_2}, R_{b_3}, R_{b_4}, R_{b_5}, R_{b_6}$ are selected from R_b . Each of ($R_{b_1}, R_{b_2}, R_{b_3}, R_{b_4}, R_{b_5}, R_{b_6}$) is a block consisting of five characters. Each block is used as a secret key for non-linear polygraphic substitution and transposition which will be explained in the following sections. $R_{b_1}, R_{b_2}, R_{b_3}, R_{b_4}, R_{b_5}, R_{b_6}$ used to generate MAC for a message M are not reused. For a different M

different $R_{b_1}, R_{b_2}, R_{b_3}, R_{b_4}, R_{b_5}, R_{b_6}$ will be selected from (R_b).

b) *Secret keys extraction from sensors data:* Symmetric implicit schemes require all sensors to be in the same context and perfectly synchronized, so all sensors generate similar secret keys. Sensor data we used include pressure, temperature, and RSSI. Data of node should be similar between all the neighbours if they, are in the same context. To make the values similar so that all nodes have similar time series, various techniques such as probability density function, cross approximate entropy, correlation, regression, and binning algorithm can be used. Considering the resource constrained characteristics of IoT devices, we use the binning algorithm. The binning is performed in three steps. First, as shown in Algorithm 1.2, pressure, temperature, and RSSI data is extracted at the time interval of 2 milli-seconds (or ten consecutive data points). Second, mean value of extracted data points is calculated. Finally, mean value is rounded to the nearest multiple of 10. The same steps are performed for every data type (temperature, pressure, RSSI). The values extracted are used to generate S_k , which will be further explained below.

Algorithm 1.1: Extract random bits from business rules

```

1: Input: Business rules
2: Output:  $R_b = (R_{b_1}, R_{b_2}, R_{b_3}, R_{b_4})$ 
3: for  $length(BusinessRules)$  do
4:    $MaxLineSize \leftarrow length\{BusinessRules\}(Line)$ 
5: end for
6:  $BitFromStartOfLine = 0$ 
7:  $BitFromEndOfLine = 0$ 
8:  $counter = 0$ 
9:  $R_b = []$ 
10: for  $MaxLine = 1$  to  $MaxLineSize$  do
11:    $INCREMENT\ BitFromStartOfLine$ 
12:   for  $EachLine = 1$  to  $length(BusinessRules)$  do
13:      $StartingBit = BusinessRules\{EachLine\}(BitFromStartOfLine)$ 
14:      $EndingBit = BusinessRules\{EachLine\}length(BusinessRules\{EachLine\}) - BitFromEndOfLine$ 
15:     if  $BitFromStartOfLine \leq length(BusinessRules\{EachLine\})/2$  then
16:        $R_b(counter) \leftarrow StartingBit$ 
17:        $INCREMENT\ counter$ 
18:        $R_b(counter) = EndingBit$ 
19:        $INCREMENT\ counter$ 
20:     end if
21:   end for
22:    $INCREMENT\ BitFromEndOfLine$ 
23: end for
24: End

```

The key space looks like $K_{space} = \{P[], T[], RSSI[], R_b\}$. Overall, our key space to generate MAC is 400 bits for message M of 160 bits. 16 bits are assigned to sensor data,

Algorithm 1.2: Data processing

```

1: Input: Dataset containing data of temperature (T), pressure (P), RSSI sensors
2: Output: Processed sensors (T, P, RSSI) data
3: for  $Size(Dataset)$  do
4:   for  $i = 1$  to  $10$  do
5:      $P \leftarrow pressuredata$ 
6:      $T \leftarrow temperaturedata$ 
7:      $RSSI \leftarrow RSSIdata$ 
8:   end for
9:    $MeanOfP \leftarrow mean(P)$ 
10:   $MeanOfT \leftarrow mean(T)$ 
11:   $MeanOfRSSI \leftarrow mean(RSSI)$ 
12:   $PisRounded \leftarrow roundn(P, 1)$ 
13:   $TisRounded \leftarrow roundn(T, 1)$ 
14:   $RSSIisRounded \leftarrow roundn(RSSI, 1)$ 
15:   $P[] \leftarrow PisRounded$ 
16:   $T[] \leftarrow TisRounded$ 
17:   $RSSI[] \leftarrow RSSIisRounded$ 
18: end for

```

in case features values are large (big integers). 8 bits are assigned to each R_b character of the block.

$$S_k = \begin{bmatrix} RSSI & RSSI & RSSI & RSSI & RSSI \\ P & P & P & P & P \\ T & T & T & T & T \\ RSSI & RSSI & RSSI & RSSI & RSSI \\ R_b & R_b & R_b & R_b & R_b \end{bmatrix}$$

The secret key (S_k), is a (5×5) matrix used for non-linear polygraphic substitution based on P , T , $RSSI$ and R_b . The two rows in S_k are the same to make the matrix dependent, thus non-invertible. Alternatively, it is also possible to select S_k entries having determinant zero. However, this may lead to computation overhead. Size of S_k is 280 bits. There are no restrictions on the size of S_k entries, except that they should be positive integers. Having no restrictions allows having an extensive key space.

2) *Message signing:* As shown in the Algorithm 1.3, in the signing step we generate tag τ , where $\tau = MAC_{K_{space}}(M)$. In order to generate τ for message M of 160 bits, M is split in 4 blocks $m_1|m_2|m_3|m_4$, where each block is 40 bits. In each block, a random special character is inserted. If necessary, the message M is padded with additional random special characters R_s to make it multiple of 160 bits. Different cryptographic schemes are used to sign each block of the message M using different secret keys, thus generating four cipher blocks (C_1, C_2, C_3, C_4) for four messages blocks. The generated τ is 320 bits. τ generation has the following steps.

a) *Non-linear polygraphic substitution (round 1):* S_k for this round consists of $P, T, RSSI$ and R_{b_1} . In this round of non-linear polygraphic substitution, C_1 is computed by multiplying square matrix (5×5) S_k with m_1

Algorithm 1.3: MAC generation

```

1: Input:  $RSSI, P, T, R_b$ , Message ( $M$ )
2: Output: = MAC
3: Message( $M$ ) is split into 4 blocks  $m_1, m_2, m_3, m_4$ 
4: for Secretkey ( $S_k$ ) = 1 to 5 do
5:    $S_k \leftarrow RSSI[]$ 
6:    $S_k \leftarrow P[]$ 
7:    $S_k \leftarrow T[]$ 
8:    $S_k \leftarrow RSSI[]$ 
9:    $S_k \leftarrow R_{b_1}[]$ 
10: end for
11:  $C_1 \leftarrow S_k.m_1 + R_{b_3}$ 
12:  $C_2 \leftarrow$  columnar transpose  $m_2$  using  $R_{b_2}$ 
13: for Secretkey ( $S_k$ ) = 1 to 5 do
14:    $S_k \leftarrow RSSI[]$ 
15:    $S_k \leftarrow P[]$ 
16:    $S_k \leftarrow T[]$ 
17:    $S_k \leftarrow RSSI[]$ 
18:    $S_k \leftarrow R_{b_3}[]$ 
19: end for
20:  $C_3 \leftarrow S_k.m_3 + R_{b_4}$ 
21:  $C_4 \leftarrow$  columnar transpose  $m_4$  using  $R_{b_2}$ 
22:  $C_{12} \leftarrow C_1 \text{ XoR } C_2$ 
23:  $C_{34} \leftarrow C_3 \text{ XoR } C_4$ 
24:  $i = 1$ 
25:  $j = \text{length}(C_{12})$  or  $j = \text{length}(C_{34})$ 
26: for  $j$  do
27:    $\text{Multiply}_p \leftarrow C_{12}[i].C_{34}[j-i]$ 
28:   INCREMENT  $i$ 
29: end for
30:  $i = 1$ 
31: for  $j$  do
32:    $\text{Multiply}_s \leftarrow C_{12}[i].C_{34}[i]$ 
33:   INCREMENT  $i$ 
34: end for
35:  $C_{1234} \leftarrow \text{Multiply}_p || \text{Multiply}_s$ 
36:  $MAC = C_{1234} \text{ mod } 64$ 
37: End

```

vector and adding column vector (5×1) R_{b_3} to the output. Therefore:

$$C_1 = S_k * m_1 + R_{b_3} \quad (1)$$

b) Columnar transposition (round 1): In this round of transposition, C_2 will be computed, using R_{b_5} as the key. In order to calculate C_2 , m_2 and R_{b_5} are represented as row vector (1×5) and columnar transposition will be performed.

c) Non-linear polygraphic substitution (round 2): In this round of non-linear polygraphic substitution, C_3 is computed. Here, S_k includes the same sensor data as the first round except that R_{b_2} is used in this round. R_{b_4} is added to the output of the second round of non-linear polygraphic substitution. Therefore:

$$C_3 = S_k * m_3 + R_{b_4} \quad (2)$$

d) Columnar transposition (round 2): In this round of transposition, C_4 will be computed similar to computing C_2 , except column vector (5×1) R_{b_6} will be used as key.

e) XoR operation: To further obfuscate, pairwise XoR operation will be performed. In order to perform the XoR, C_1 is transposed to represent in row vector (1×5) format and pairwise XoR is performed with C_2 . The output of the pair XoR operation is C_{12} . Similar to C_{12} XoR operation, C_3 is transposed to be represent in row vector (1×5) format and pairwise XoR with C_4 . C_{34} is the pairwise XoR of C_3 and C_4 . Moreover, to dissipate the relationship between the message and τ , two rounds of mutual multiplication will be performed. In the primary multiplication round, Multiply_p , the first block of C_{12} is multiplied with the last block of C_{34} and the second block of C_{12} is multiplied with the second last block of C_{34} and so on. In the secondary multiplication round Multiply_s , the first block of C_{12} is multiplied with the third block of C_{34} , second block of C_{12} is multiplied with the second block of C_{34} , third block of C_{12} with the first block of C_{34} , fourth block of C_{12} with the fifth block of C_{34} and finally fifth block of C_{12} with the fourth block of C_{34} . Each block of the primary multiplication and the secondary multiplication is further divided in four blocks of two decimal digits, and the extra digits are discarded. If it is not possible to divide into four blocks, then zeros are added. The output of both multiplication rounds is encoded in modulo 64. The modulo 64 is represented as: (A, B, ...Z) = (0, 1, ...25), (a, b, ...z) = (26, 27, ...51), (0, 1, ...9) = (52, 53, ...61), (+) = (62), (/) = (63).

3) Verification: In general, MAC is encrypted along with the message and sent to the receiver. The recipient receives the message M and the τ . Verifying algorithm (Ver) verifies M - τ pairs; $\text{Ver}(M, \tau)$. In order to check the validity of the τ , the receiver calculates the τ in a similar way as the sender. If the calculated τ and the received τ are similar, $\text{Ver}_{K_{space}}(M, \tau) = 1$ (τ is valid), otherwise $\text{Ver}_{K_{space}}(M, \tau) = 0$ (τ is malicious).

$$\text{Ver}_{K_{space}}(M, \tau) = \begin{cases} 1, & \text{if } \text{MAC}_{K_{space}}(M) = \tau \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

If MAC of the sender and the one computed by the receiver are not the same, then either the sender node is malicious or a malicious behavior (or anomaly) has emerged. If the two computed MACs are similar, this means that the message is sent by an authentic or the intended sender in the same context.

IV. EXAMPLE

As shown in Figure 2, we have developed an experimental setup which resembles smart pallets. The setup consists of eight Promove [13] nodes placed on a trolley.

Each sensor is capable of measuring pressure, temperature and RSSI. The trolley is locomoted to collect data. We collected the data three times, for approximately 10 minutes each time. To analyze different patterns, for comparison and evaluation purposes. The data is collected with sampling rate of 500 Hz. An example of business rules is shown below.

```

if:
  all the following conditions are true
    ◇ driver is "XYZ van t Veer"
    ◇ delivers Product "Ice Cream" on "17th June" 2019
    ◇ destination(s) "XYZstraat xxx, 7545 Enschede"
    ◇ customer "XYZ Logistics B.V"
    ◇ following contract requirements are fulfilled
  ...
then:
  ◇ customer accepts the delivery
else:
  ◇ customer rejects the delivery
end

```

In order to demonstrate the applicability of the proposed algorithm, we generate tag τ for two messages (which are slightly different from each other) using the same key. For $M = \text{TheSecureSchemes}$, the $\text{MAC}_{K_{space}}(M)$ generated is `esRbj8m8cDr2jvzLkcX4OSA5rOHA0AKSerGYkv92` and for $M = \text{THESECURESECHMES}$ the $\text{MAC}_{K_{space}}(M)$ is `UHGLXyTeScLjWac-qXUP1cCwUL4JdcxMVUGTAXUye`. By comparing, it is obvious that even if the key is reused still the adversary will have nought advantage.

Figure 2. The experimental setup mimics smart pallet, is used to collect data.

V. RESULTS

A. MAC generation

All nodes in our experimental setup are in the same context, therefore we expect all nodes generate the same keys. However, RSSI data sometimes differ among the nodes. Therefore, the τ calculated for message M by different nodes differ and consequently nodes are not able to communicate. Figure 3 represents the similarity and dissimilarity of τ generation among the nodes using different rounding. By rounding the sensor data to the nearest multiple of 10, dissimilarity among the nodes is

high. However, by rounding the sensors data to the nearest multiple of 30, the data among all nodes is perfectly similar.

B. Cryptanalysis

Regarding cryptanalysis, given the sources of randomness in our methodology, an adversary would require $\min(2^k, 2^n)$ cost to win, where $k(400)$ is the size of the key and $n(320)$ is the size of the output. Resilient security feature of our proposed algorithm is that as the sensor data changes, new keys will be generated. So ideally for key recovery attack [15], the attacker would require about 2^{400} operations for each message M .

C. Complexity

We analyze the order of complexity of our mechanism using Big-O notation. For matrix multiplication, the order of complexity is $O(n^3)$, for matrix addition is $O(n^2)$ and for vector addition is $O(n)$.

VI. LIMITATIONS

The usage of sensors data for security purposes can evolve into many challenges, such as, an attacker might get physical access to the sensors and thus control the sensors, and mostly collinearity exists among sensors, so an attacker might be able to predict sensors data. These challenges can most probably imperil our proposed mechanism. Therefore, we presume that the sensors data to be secured. After discussion with few companies and experts in business rules, we came to know that business rules are highly secretive and are never made public. However, if sensors data or business rules are not secured, and if the adversary has physical access to smart logistic environment, it might pose threats to the proposed mechanism.

VII. FUTURE WORK

In the future, we will examine the usage of sensor data for asymmetric mechanisms and utilize additional data sources such as gyroscope, and compass.

VIII. CONCLUSION

The convergence of multiple sensors data to form a distributed message authentication mechanism demonstrates resilience against attacks and remarkable convenience in implementation. Different cryptographic primitives are used to ensure maximum level of security. The distinctive characteristic of the proposed scheme is the reusability of sensors data and business rules for security reasons. Along with its usage as MAC, it can also be used for anomaly detection. Given the extensive key space (sensors data and business rules), it is possible to use our proposed mechanism as one-time pad, if new S_k and R_b are generated for each message M . One possible limitation of our proposed mechanism is that if the nodes do not have similar data, the nodes will not be able to communicate. Further increasing the rounding to the nearest multiple of 30, can circumvent this limitation.

Figure 3. Pairwise MAC calculation between the eight nodes in our experimental setup. "Connected" (represented by Blue) means that pairwise nodes have similar data and thus generated MAC is similar and nodes may communicate. "Disconnected" (represented by Red) means not similar data and nodes are not able to communicate.

ACKNOWLEDGMENT

This work has been partially supported by the EFRO, OP Oost program in the context of Countdown project.

REFERENCES

- [1] N. Hackius, and M. Petersen. Blockchain in Logistics and supply chain: Trick or Treat? (2017). Hamburg University of Technology Kühne Logistics University
- [2] S. Popov The Tangle. Version 1.4.3. (2018)
- [3] L. S. Hill. Cryptography in an algebraic alphabet. (1929). The American Mathematical Monthly, Vol. 36, No. 6, pp. 306-312
- [4] J. Overbey, T. William and W. Jerzy. "On the Keyspace of the Hill Cipher." Cryptologia 29 (2005): 59-72
- [5] R. Mayrhofer, H. Gellersen, Shake Well Before Use: Authentication Based on Accelerometer Data. (2007). In: LaMarca A., Langheinrich M. Truong K.N. (eds) Pervasive Computing. Pervasive. Lecture Notes in Computer Science, vol 4480
- [6] T. Azzabi, H. Farhat and N. Sahli, "A survey on wireless sensor networks security issues and military specificities,". (2017). International Conference on Advanced Systems and Electric Technologies (IC_ASET), Hammamet, 2017, pp. 66-72. DOI: 10.1109/ASET.2017.793668
- [7] M. Jakobsson, E. Shi, P. Golle, and R. Chow. (2009). Implicit authentication for mobile devices. In Proceedings of the 4th USENIX conference on Hot topics in security (HotSec'09). USENIX Association, Berkeley, CA, USA, 9-9.
- [8] W. Lee and R. B. Lee, "Multi-sensor authentication to improve smartphone security,". (2015) International Conference on Information Systems Security and Privacy (ICISSP), Angers, pp. 1-11
- [9] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. (2010) Implicit Authentication through Learning User Behavior. In: Burmester M., Tsudik G., Magliveras S., Ilić I. (eds) Information Security. ISC. Lecture Notes in Computer Science, vol 6531.
- [10] G. Davrondzhon, H. Kirsi and T. Söndrol. (2006). Biometric Gait Authentication Using Accelerometer Sensor. Journal of Computers. 1. 10.4304/jcp.1.7.51-59.
- [11] N. Sastry, R. Shankar, and David Wagner. (2003). Secure verification of location claims. In Proceedings of the 2nd ACM workshop on Wireless security. ACM, New York, NY, USA, 1-10. DOI: <https://doi.org/10.1145/941311.941313>
- [12] F. Monrose and A. Rubin. (1997). Authentication via keystroke dynamics. In Proceedings of the 4th ACM conference on Computer and communications security. ACM, New York, NY, USA, 48-56. DOI=<http://dx.doi.org/10.1145/266420.266434>
- [13] ProMove. [Online]. Available: <http://inertia-technology.com/>. [Accessed: 10- July - 2018]
- [14] ASCII table and description. [Online]. Available: <http://www.asciitable.com/>. [Accessed: 23 - Jan - 2018]
- [15] B. Preneel. Cryptanalysis of Message Authentication Codes. (2004) Katholieke Universiteit Leuven. Department Electrical Engineering-ESAT/COSIC September 15
- [16] Z. Zhang, M. C. Y. Cho, C. Wang, C. Hsu, C. Chen and S. Shieh, "IoT Security: Ongoing Challenges and Research Opportunities," 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications, Matsue, 2014, pp. 230-234. doi: 10.1109/SOCA.2014.58
- [17] S. L. Hong and C. Liu, "Sensor-Based Random Number Generator Seeding," in IEEE Access, vol. 3, pp. 562-568, 2015. doi: 10.1109/ACCESS.2015.2432140