LPWA ネットワークにおけるアドレス乱数化の提案

妹尾尚一郎 [1]　古谷彰教 [1]　中山裕之 [1]

概要: さまざまな LPWA (Low Power Wide Area)ネットワーク技術により，比較的広いエリアであっても IoT (Internet of Things)アプリケーションの各種データを安価かつ容易にやり取りできるようになっている．一方 LPWA の普及に伴うセキュリティ上の懸念も生じており，数十 km に達する LPWA の到達距離と通信事業者に限られない多様な LPWA ネットワークの存在，また低価格な汎用品として提供される LPWA デバイスに内蔵できるセキュリティ対策の限界から，それらの解決は困難が見込まれる．LPWA ネットワークプロトコルの一つである LoRaWAN は暗号化とメッセージ認証というセキュリティ対策を含むが，デバイスアドレスは暗号化されないためトラヒック解析には対抗できない．本稿は LPWA ネットワークの更なるセキュリティ対策として，データフレームに一時的なデバイスアドレスを付与するアドレス乱数化を提案する．アドレスを乱数化したデータフレームの送信者と受信者間のアドレス同期プロトコルを述べ， Arduino ベースの LoRa デバイスへの LoRaWAN オープンソースを利用したプロトタイプ実装について報告する．プロトタイプでは比較的小さなオーバヘッドでアドレス乱数化を追加できた．

A Proposal of Address Randomization for LPWA Networks

SHOICHIRO SENO[1]　AKINORI FURUYA[1]　HIROYUKI NAKAYAMA[1]

Abstract: A family of LPWA (Low Power Wide Area) network technologies allow low cost and easy exchange of sensor data and other information essential for IoT (Internet of Things) applications over a relatively wide area. Increased use of LPWA networks implies security concerns which are difficult to solve due to LPWA's long reachability up to tens of kilometers and existence of various LPWA networks not limited to network operators. Low cost, off-the-shelf LPWA devices come with limited security measures built in them. One of LPWA network protocols, LoRaWAN, includes encryption and message authentication as security measures. But it is still possible to apply traffic analysis against it because device addresses are not encrypted. This paper proposes an additional security measure for LPWA networks, i.e., address randomization which assigns a temporary device address for every data frame. It describes an address synchronization protocol between the sender and the receiver of a data frame with a randomized address and its prototype implementation on Arduino-based LoRa devices using open source LoRaWAN codes. The prototyping demonstrated relatively small overheads with address randomization.

1. はじめに

IoT (Internet of Things)の発展に伴い，IoT 機器から各種データをゲートウェイ(GW)を介してインターネットとやり取りする LPWA (Low Power Wide Area)ネットワークの研究開発が盛んである[1]．LPWA ネットワークにおいてセキュリティを確保するため，例えば LPWA の 1 つ LoRa のプロトコルスタック LoRaWAN では，暗号化と MIC (Message Integrity Code)認証を規定している[2].

本稿では LPWA の通信エリアが広く，通信事業者に限られない様々な組織が運用するという特性より，更なるセキュリティ向上策が望ましいと考え，端末アドレスをランダムに変更するアドレス乱数化に着目した．アドレス乱数化とは，フレーム内のアドレスをランダムに変更して端末の特定やトラヒック解析を困難化する手法である．レイヤ 2 で使用される MAC アドレスが無線 LAN などの普及に伴いモバイルコンピューティング端末にて広く使用されるようになったことから，後述するようにその使用者の所在についてプライバシーを保護するため PC やスマートフォンの OS に MAC アドレスの乱数化が取り入れられた．LPWA ネットワークが収容する IoT 端末の場合は，下に例示するセキュリティ上の懸念がアドレス乱数化によって解消されると期待できる．

(A) LPWA ネットワークの適用例として，児童や認知症者の見守り[3][4]，バスロケーションシステム[5]など移動 IoT 端末を使用するものが提案されている．移動 IoT 端末のアドレスが固定であれば，移動 IoT 端末からのフレームを傍受することでその所在の推定が可能と考えられる．児童や認知症者の所在の推定はこれらの人々に対する犯罪の誘発につながる可能性があるが，移動 IoT 端末のアドレスを乱数化すれば所在を追跡することは困難である．

(B) LPWA が適用されるセンサネットワーク端末は電池駆動が多いと想定されるが，端末のアドレスが固定であれば，その端末とゲートウェイとの通信シーケンスへ同じアドレ

1 徳島文理大学
Tokushima Bunri University

スを含むフレームを割り込ませることで，端末の電池を消耗させ動作不能に陥らせる攻撃が考えられる．例えばLoRaWANでは，LoRaデバイスがゲートウェイへデータを送信すると確認応答(Ack: Acknowledgement)を待ち，Ackを受信しなければデータを再送するので，データ送信のタイミングでAck妨害フレームを割り込ませる攻撃があり得る．LoRaWANの全てのフレームはMIC認証の対象なのでAck妨害フレームはMIC不一致として破棄されるものの，余分なフレーム受信とデータ再送によってLoRaデバイスの電池が消耗する．LoRaデバイスのアドレスが乱数化されていれば，デバイスを特定できないためこの攻撃は成立しない．

本稿では，LoRaネットワーク向けアドレス乱数化プロトコルとその実装について報告する．以下，2.において関連する研究を概観して本稿との関係を明らかにし，3.においてアドレス乱数化プロトコルを提案し，4.においてその実装を述べ，5.を締めくくりとする．

2. 関連研究

レイヤ2で使用されるMACアドレスは製造者が固定的に機器に割当てるLAN上のアドレスとして標準化されたが，無線LANの端末識別にもMACアドレスが使用される．PCやスマートフォンなどモバイルコンピューティング端末が送信する無線LANフレームを傍受すると，容易に端末，ひいてはその使用者の所在を特定でき，使用者の感知しないところで所在先の履歴を収集されるというプライバシー上の懸念がある．そこで，無線LANアクセスポイントへ接続するモバイルコンピューティング端末のMACアドレスの乱数化が提案された[6]．現在はPCやスマートフォンのOSにMACアドレスの乱数化が組み込まれており，MACアドレス以外のプロトコル要素や物理レイヤの情報を使って無線LAN接続端末を特定する手法や，端末の特定を防ぐ手法が研究されている[7]-[9]．またレイヤ3以上においても，IPアドレスやポート番号の乱数化によるインターネット上のトラヒック解析の困難化などが報告されている[10]-[12].

一方，LPWAネットワークのレイヤ2におけるアドレス乱数化については，著者の知る限り報告されていない．これについて本稿が貢献しようとするのは次の各点である．

i) LPWA，特にLoRaWANに即したアドレス乱数化プロトコルの提案

ii) i)における頻繁な乱数アドレス更新と乱数アドレス同期

iii) 上記の実装における暗号化・MIC認証の利用

従来の無線LAN接続端末のMACアドレス乱数化および端末不特定化の研究[6]-[9]においては，端末と無線LANアクセスポイントとの接続がMACアドレスによって識別されるため，乱数アドレスを更新するには端末と無線LANアクセスポイントとのアソシエーションを再設定する必要があり，通信中の乱数アドレスの頻繁な更新には限界がある．このため端末を特定できなくても無線LANを傍受することで，異なるMACアドレス毎に総トラヒック量や無線LAN接続時間を把握するトラヒック解析が可能である．一方，IPアドレスやポート番号の乱数化においては，頻繁な乱数の更新を提案する研究もある[11][12].

本稿は，LPWAネットワークにおいてデータ送受信と同期した頻繁な乱数アドレス更新を可能にする手法を新たに提案するものであり，これは従来研究されてこなかった．

3. LoRaネットワークのアドレス乱数化の提案

3.1 提案の背景

LoRaWANのフレームフォーマットを図1に示す．図中の()内は各フィールドのオクテット長である．エンドデバイス(Dev)とGWが共有する暗号鍵，およびフレーム毎のフレームカウンタ(FCnt)値やDevを識別する固有アドレス(DevAddr)などから成る初期ベクトル(IV)から計算されるMICが各フレームに付随し，受信側はMIC値を用いてフレーム認証を行う．またフレームペイロードは，暗号鍵および FCnt と DevAddr を含む IV を用いて暗号化される．DevAddr がフレームヘッダ(FHDR)に含まれ暗号化されないため，フレームを傍受するとエンドデバイスを特定できてトラヒック解析等が可能なことから，セキュリティリスクが潜在する．

LoRaWANにおけるデータ送受信の基本シーケンスを図2に示す．DevはLoRaWANネットワークとのアクティベーション(activation)と呼ぶ手順によって接続関係を確立し，DevAddrおよび共有暗号鍵を共有する．Devからのデータ

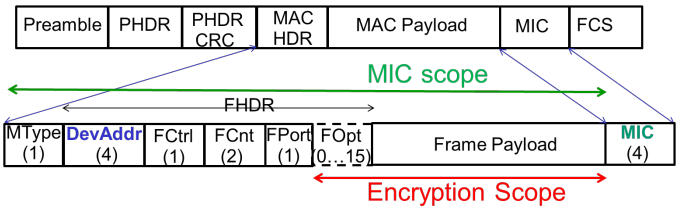


図 1　LoRaWAN フレームフォーマット

Figure 1　LoRaWAN Frame Format.

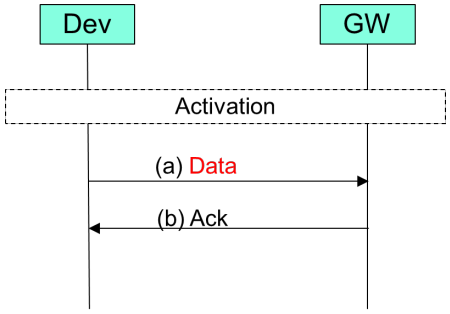


図 2　LoRaWAN データ転送シーケンス

Figure 2　LoRaWAN Data Transmission Sequence.

フレームを GW が受信すると，その FCnt 値を含む Ack が返送される．データフレーム送信後一定時間内に Ack を受信しないとき，Dev はデータフレームを再送する．

3.2 アドレス乱数化プロトコル

筆者は従来よりレイヤ 2 ネットワークのセキュリティについて研究しており[13][14]，上記の背景に基づいて LoRaWAN のデータ送受信シーケンスに範を取り，LoRaWAN の暗号化・MIC 認証機能を補完するものとして，下記の特徴を持つアドレス乱数化プロトコルを提案する．

① Dev からの最初のデータを契機とする乱数化の同期

GW は Dev からの最初のデータフレームをフレームカウンタの初期値で識別し，これを契機として該 Dev 対応のアドレス乱数列の生成を開始すると共に，該 Dev への Ack に相乗りさせてアドレス乱数同期コマンドを通知する．Dev は同期コマンドを受信してアドレス乱数列の生成を開始し，生成した乱数アドレスを含む同期コマンド応答を GW へ送信する．以後，GW におけるデータフレーム受信と Dev における対応 Ack の受信を契機に乱数を生成することで，GW と Dev は同期して乱数アドレスを更新する．データフレームないし Ack が不達の場合，Dev はデータフレーム送信後一定時間内に Ack を受信しなければデータフレームを再送するので，後述するように再送データフレームに対する Ack が受信されればアドレス乱数列の同期が維持される．

② LoRaWAN の暗号化・MIC 認証による同期の保護

LoRaWAN は図 1 に示したようにフレームペイロードの暗号化範囲および MIC によるフレーム認証の範囲を規定している．暗号化と MIC 認証を GW から Dev へのアドレス乱数同期コマンドおよびその応答に適用することで，コマンドと応答の秘匿と改ざん防止を図り，乱数アドレスの同期を保護する．

③ LoRa Alliance が割当てる DevAddr との衝突回避

LoRaWAN における DevAddr の構成を図 3 に示す．32 ビットの DevAddr は LoRa の標準化・普及を目指す団体 LoRa Alliance が割り当てており，LoRa Alliance の各メンバへ図中の Prefix から NwkID までの上位アドレスが割り当てられ，各メンバは NwkAddr 部分の下位アドレスを自由に Dev へ割り当てることができる．アドレス乱数化に当たっては，非 LoRa Alliance メンバ用に確保されている Prefix 値 0b000000 および 0b000001 のアドレス空間（2^{26} 個のアドレスを含む）からランダムにアドレスを選択することで，LoRa Alliance がそのメンバへ割り当てるアドレスとの衝突を回避する．

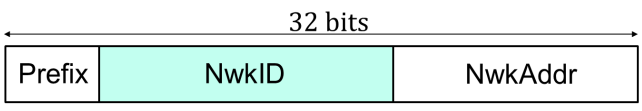


図 3　DevAddr の構成
Figure 3　DevAddr Structure.

上記①，②を盛り込んだアドレス乱数化プロトコルの乱数化開始シーケンスを図 4 に示す．LoRaWAN の activation と同様の手順によって Dev と GW の接続関係が確立され，Dev の固有アドレス DevAddr および Dev と GW の共有暗号鍵が両者へ設定されているものとする．Dev から GW への最初のデータフレーム(a)送信時のシーケンスを図 4 に示す．本データフレームのアドレスは乱数化せず DevAddr を使用する．GW は本データフレーム中のフレームカウンタ FCnt が初期値であることから最初のデータと認識し，送信元 Dev に対するアドレス乱数化を開始する．すなわち，GW は Dev へのアドレス乱数同期コマンドを Ack に相乗りさせて送信し((b))，Dev がその応答((c))を GW へ送信することで，GW と Dev 間でアドレス乱数列の生成を同期させる．いったんアドレス乱数列が同期すると，以後のデータフレーム送信アドレス乱数列が同期すると，以後のデータフレーム送信((d))とこれに対する Ack ((e))毎に，Dev と GW は新たな乱数を生成して乱数アドレスを更新する．

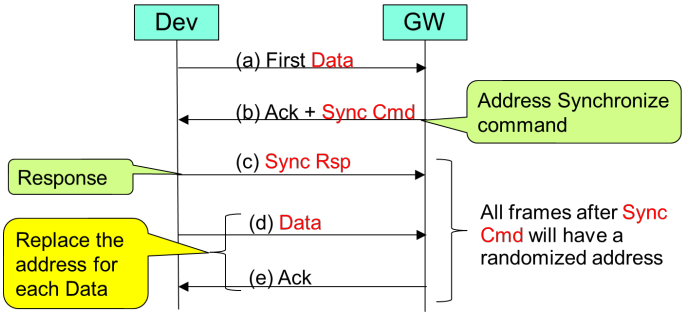


図 4　アドレス乱数化の初期シーケンス
Figure 4　Initial Address Randomization Sequence.

図 4 において赤字で表されるデータ(Data)，同期コマンド(Sync Cmd)，同期応答(Sync Rsp)に，LoRaWAN のフレームペイロード暗号化を適用し内容を秘匿する．さらに全フレームに LoRaWAN の MIC 認証を適用し，改ざんを防止する．

上記に加え，アドレス乱数化プロトコルには下記の状況への対処も盛り込んで頑健性を高めている．

④ 他デバイスとのアドレス衝突の回避

GW は Dev 毎に異なる暗号鍵と初期ベクトル(IV)からのアドレス乱数列を生成するが，複数の Dev を収容する GW において異なる Dev のアドレス乱数列が同時に同じアドレスを生成する確率は 0 ではない．そこで GW においてある Dev からのデータ受信を受け新たな乱数アドレスを算出するとき，他の Dev のアドレスとの衝突の有無を調べ，衝突があればアドレス乱数列のさらに先のアドレス乱数を使用することにした．このとき乱数アドレスの同期を維持するため，GW から Dev への Ack においてアドレス乱数列のスキップを通知する．

一方，実運用される LoRa ネットワークの数が増えると他の LoRa ネットワークと運用エリアが重なり，周波数などの無線パラメータが一致するとそのフレームを受信する可能性が生じるが[15]，この場合は次のようになる．

提案するアドレス乱数化は③で述べたように非 LoRa Alliance メンバ用 Prefix を含む乱数アドレスを割当てるので，他の LoRa ネットワークが LoRa Alliance メンバであればアドレス衝突は起きない．そこで，他の LoRa ネットワークが LoRa Alliance メンバでなくアドレス衝突が起きる場合を考える．

(1) 他の LoRa ネットワークからのデータフレームを GW が受信し，そのアドレスが GW の保持している Dev の乱数アドレスのいずれかと一致する場合：提案するアドレス乱数化においては全ての LoRa フレームを MIC 認証の対象としており，MIC は暗号鍵と IV とデータ内容から計算されるため，他の LoRa ネットワークからのデータフレームは MIC 不一致として破棄される．例え非常に低い確率で MIC 認証をパスしたとしても，フレームカウンタ等の値が整合していなかったら，やはり破棄される．このように他の LoRa ネットワークからのデータフレームは GW が受け入れず，GW と Dev 間の本来のデータフレーム送受信へ影響する確率は非常に低い．

(2) 他の LoRa ネットワークからの Ack を Dev が受信しそのアドレスが乱数アドレスと一致する場合：同様に MIC 不一致として破棄される．

(3) データフレーム，Ack 以外のフレーム種別の場合：(1)，(2)と同様に MIC 認証により破棄される．

上記のように本プロトコルはアドレス衝突を回避するように配慮しており，他の LoRa ネットワークからのフレームによってアドレス衝突が万一生じたとしても，MIC 認証を適用していることから GW と Dev の間の乱数アドレス更新手順が乱れる可能性は非常に小さい．またデータフレームとその Ack の送受信毎にアドレスを更新するため，アドレス衝突が継続することはない．

⑤ アドレス乱数化開始の誤認防止

GW は Dev からのデータフレーム中の 16 ビットのフレームカウンタ FCnt が初期値であることをアドレス乱数化の開始契機とするため， FCnt 値が更新されて初期値と同じ値になったときに GW がアドレス乱数化の開始タイミングと誤認しないことが必要である．このため，一度アドレス乱数化を開始した Dev について，GW が再度の同期を行わないようにする．なお FCnt を更新するとき単純に加算すると， フレーム傍受者が FCnt 値から複数のフレームを関連付ける可能性があるので，アドレスに加えて FCnt 値も乱数化している．

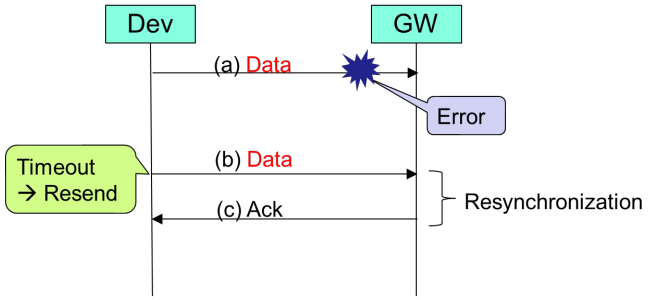
アドレス乱数化と暗号化・MIC 認証を併用することで，データの秘匿や改ざん防止のみならずトラヒック解析等への防御が可能になると期待できる．

3.3 アドレス同期の維持

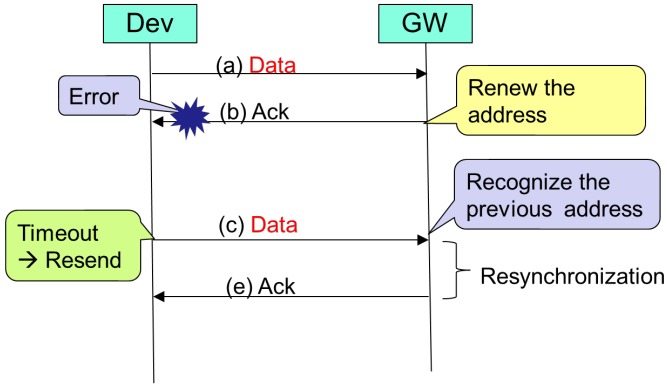
Dev からのデータフレーム送信と GW からの Ack 返送に同期した乱数アドレス更新において，これらのフレームが物理レイヤの雑音などに起因して不達となったときは Dev からのデータフレーム再送によって同期を維持する．フレーム不達時の再送による再同期の例を図 5 に示す．

図 5 (a)のデータフレームがエラーにより不達となる場合は，再送データフレームのアドレスおよび FCnt 値は元のデータフレームと同じであるため，GW はこのフレームを受信して Ack を返送し乱数アドレスを更新する．

図 5 (b)の Ack がエラーにより不達となる場合は，GW は Ack 返送後に乱数アドレスを更新するがそれまでのアドレスと FCnt 値も記憶しておき，再送データフレームを受信するとこれを識別して Ack を返送することで，乱数アドレスを再同期する．



(a) Data Frame Error



(b) Ack Frame Error

図 5　アドレス乱数化の再同期

Figure 5　Address Randomization Resynchronization.

上記の動作に対応した Dev の状態遷移の例を図 6 に示す．Dev は立ち上げ後に Running 状態へ遷移し，データ生成のタイミングに応じてデータフレーム送信とタイマ起動を行い，Ack_Wait 状態へ遷移する．Ack_Wait 状態において Ack を受信すると Running 状態へ戻って次のデータ生成を待つが，Ack_Wait 状態にてタイマが満了するとデータフレームを再送する．またデータフレームを一定回数再送しても Ack を受信できないときは， 次のデータ生成を待つため Running 状態へ遷移する．

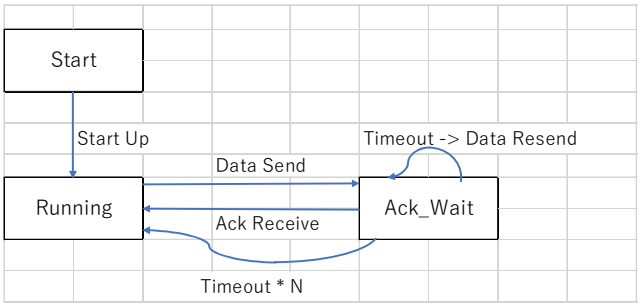


図 6　Dev の状態遷移例

Figure 6　End-device State Transition Example.

4. 実装

Arduino マイコン[16]を搭載した LoRa デバイス LoRa mini-JP と LoRa ゲートウェイ[17]において，LoRaWAN 実装のオープンソース LMIC [18]の AES (Advanced Encryption Standard)関数を利用し，暗号化・MIC 認証・アドレス乱数化を実装した．

アドレス乱数列の計算は，AES 暗号化関数 $aes128()$ を用い下の(1)式により実行するようにした．ここで Key は Dev と GW が共有する暗号鍵，IV は Dev の固有アドレス $DevAddr$，Dev からの最初のデータを GW が受信した時刻 $Setuptime$，カウンタ $Counter$および固定値から成る初期ベクトルである．図 4 に示したアドレス乱数化の開始以降，Dev と GW はデータフレーム送信後および Ack 送信後にカウンタを 1 づつ進めて新たな乱数アドレスを計算し，次のデータフレーム送受信のアドレスとしてアドレス更新を行う．ただし，データフレーム再送時には以前送信したデータフレームをそのまま送信するためアドレスは更新されない．再送時を除くと，データフレームの送信とその Ack というフレーム伝送の最小単位毎にアドレスが更新されるため，フレームを傍受しても複数のデータフレームにまたがって特定の端末を追跡することは困難である．

$$RandomizedAddress = aes128(Key, IV) \text{ -------- (1)}$$
$$IV = (DevAddr, Setuptime, Counter)$$

Dev と GW の 1 対 1 の通信において Dev の初期シーケンスをモニタした例を，図 7 に示す．図 4 に示したフレーム(a)～(e)の送受信が 1 行目～5 行目に表示されており，1 行目～3 行目のアドレス乱数化の同期の後，データフレーム毎にアドレスと FCnt 値がランダムに更新され，データフレームと同一のアドレスと FCnt 値を持つ Ack の受信によってデータフレームの送達が確認されていることが読み取れる．

Dev と GW のバイナリコードサイズ（単位: Byte）を表 1 に示す．乱数化・暗号化・MIC 認証全てを含むコードサイズ(1)は暗号化・MIC 認証のみ含むコードサイズ(2)に対し

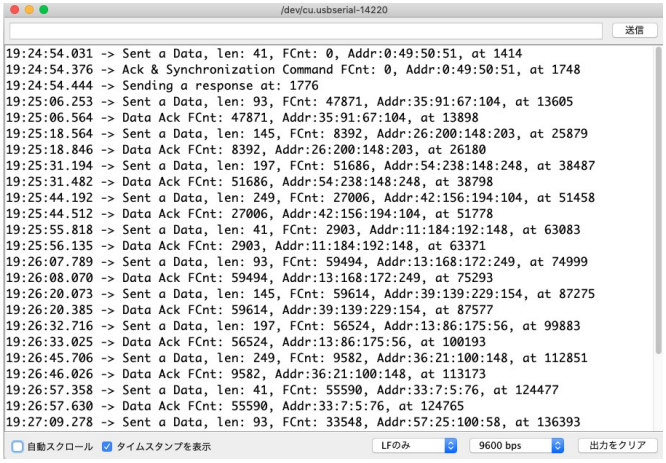


図 7　Dev における初期シーケンスのモニタ

Figure 7　Initial Sequence Monitored at the End-device.

Dev で 5.2%増，GW で 6.6%増であり，AES 暗号化関数を利用して実装したことからアドレス乱数化の追加によるオーバヘッドを小さくできた．参考までに乱数化・暗号化・MIC 認証を全て含まないコードサイズ(3)との比較では，Dev で 29.4%増，GW で 22.6%増であった．

また乱数アドレス算出の実行時間を Dev において Arduino の millis()関数を用いて測定すると，1 ms 程度であった．よって，アドレス乱数化を追加することが Dev や GW の負荷に及ぼす影響は軽微と言える．

表 1　アドレス乱数化のコードサイズ

Table 1　Address Randomization Code Size.

	(1) All	(2) Enc/MIC	(3) None	(1)/(2)	(1)/(3)
Dev	12,168	11,568	9,402	5.2%増	29.4%増
GW	18,650	17,498	15,218	6.6%増	22.6%増

5. おわりに

LoRa ネットワークにおいてデータ送受信と同期した乱数アドレス更新を可能にするアドレス乱数化プロトコルを新たに提案し，LoRa デバイスおよびゲートウェイへ実装した．LoRaWAN の暗号化・MIC 認証機能を利用することで，アドレス乱数化が効率的に実装でき，実行時のオーバヘッドも小さかった．今後はデバイス数を増やし，アドレス乱数化の効果を検証する予定である．

謝辞　本研究は，開発環境やプログラムなどのオープンソース成果物の利用なくしては実施できなかった．これらを提供して下さった方々へ感謝する．

参考文献

[1] 鄭立，IoT ネットワーク LPWA の基礎，リックテレコム，2017.
[2] LoRaWAN™ 1.1 Specification, LoRa Alliance, 2017.
[3] “Fukuoka City LoRaWAN™を活用した児童見守りトライアルについて ～保護者と塾の連携による通塾の見守り～”．NTT 西日本他報道発表，2018-02-26，https://www.ntt-west.co.jp/news

cms/fukuoka/7215/20180226.pdf，(参照 2020-05-14).
[4] “LiveRidge が福岡市と認知症見守り IoT 事業協定を締結～Fukuoka City LoRaWAN™を活用した実証実験事業を受託～”. 株式会社 LiveRidge 報道発表, 2018-02-26, https://prtimes.jp/main/html/rd/p/000000003.000022569.html.pdf，(参照 2020-05-14).
[5] 平櫻瞭太郎, 小島夢人, 加藤美範, 小林鈴花, 沢田真実, 袖美樹子, 向井宏明, LPWA を用いたバスロケーションシステムの提案, 信学技報, 2018, CQ2018-70, p. 37-42.
[6] Gruteser, M. and Grunwald, D., Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis, Mobile Networks and Applications, 2005, vol. 10, p. 315–325.
[7] Martin, J., Mayberry, T., Donahue, C., Foppe, L., Brown, L., Riggings, C., Rye, E. C., and Brown, D., A Study of MAC Address Randomization in Mobile Devices and When it Fails, Proc. Privacy Enhancing Technologies, 2017, vol. 4, p. 268-286.
[8] Vanhoef, M., Matte, C., Cunche, M., Cardoso, L., and Piessens, F., Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms, ACM Asia CCS 2016, 2016, p.413-424.
[9] 古屋優希, 朝比奈啓, 豊田健太郎, 笹瀬巌, MAC アドレスがランダム化された Wi-Fi 無線端末の同定のための非匿名化手法の検討, 信学技報, 2019, CS2019-17, p. 25-29.
[10] Sifalakis, M., Schmid, S., Hutchison, D., Network Address Hopping: A Mechanism to Enhance Data Protection for Packet Communications, ICC 2005, 2005.
[11] Luo, Y-B., Wang, B-S., Wang, X-F., Zhang, B-F. and Hu, W., RPAH: A Moving Target Network Defense Mechanism Naturally Resists Reconnaissances and Attacks, IEICE Trans. Inf. & Syst., 2017, vol. E100-D, no. 3, p. 496-510.
[12] Judmayer, A., Ullrich, J., Merzdovnik, G., Voyiatzis, A., Weippl, E., Lightweight Address Hopping for Defending the IPv6 IoT, 12th International Conference on Availability, Reliability and Security (ARES 2017), 2017, no. 20.
[13] 妹尾尚一郎，高エラー下のレイヤ 2 パケット認証方法のエラー検出性能の評価，信学会総合大会, 2018, B-8-5.
[14] Seno, S., Layer 2 Packet Authentication for IoT Sensor Networks, 12th International Conference on Mobile Computing and Ubiquitous Network (ICMU 2019), 2019, S2003.
[15] 妹尾尚一郎, 自営 LoRa ネットワークのランダムなアドレス割当の検討, 電気関係学会四国支部連合大会, 2019, 12-26, p. 129.
[16] https://www.arduino.cc/, (参照 2020-05-14).
[17] http://www.openwave.co.jp/lorawan/, (参照 2020-05-14).
[18] “Arduino-LIMC (LoraMAC-in-C) library”. https://github.com/matthijskooijman/arduino-lmic, (参照 2020-05-14).