RFIDサプライチェーンにおいて Format Transforming Encryptionを用いた 安全なEPC配送

豊田 健太郎[1]　大槻 知明[2]

概要：RFID サプライチェーンにおいて安全に EPC (Electronic Product Code) を配送することが求められている．近年，FTE (Format Transforming Encryption) を用いて EPC を暗号化することで，RFID タグのメモリの制約問題を解決できることがわかっている．しかしながら，その安全性に関する議論は不十分であった．そこで本研究では，FTE を用いて暗号化された EPC の安全性について分析を行う．その結果，FTE のフォーマットの指定を工夫することで攻撃者が正しい EPC を特定することを困難にでき，また攻撃者が EPC に関してどの程度の知識を持っているかによってフォーマットの選択方法が変わることを示す．

キーワード：RFID サプライチェーン, EPC 配送問題, FTE (Format Transforming Encryption), セキュリティ分析

Secure EPC Distribution with Format Transforming Encryption in RFID-enabled Supply Chain

Kentaroh Toyoda[1]　Tomoaki Ohtsuki[2]

Abstract: Secure EPC (Electronic Product Code) distribution is a crucial task on RFID-enabled supply chain since anyone can interrogate RFID tags in public transportation. Recently, FTE (Format Transforming Encryption) is found to be suitable to encrypt EPC since it clears the memory format constraint problem on an RFID tag. In this paper, we analyse the security perspective of the secure EPC distribution with FTE (Format Transforming Encryption). Our analysis shows that by carefully specifying these formats, an attacker is infeasible to identify genuine EPCs and the best strategy for choosing the formats varies according to attacker's knowledge.

Keywords: RFID-enabled supply chain, secure EPC distribution, FTE (Format Transforming Encryption), security analysis

1. はじめに

産業界における重大な問題のひとつに，偽物による被害がある [1]．OECD (Organisation for Economic Co-operation and Development) によると世界における偽物の市場は 2,500 億米ドルに及ぶとされている [2]．したがって，偽物の流通を抑制する仕組みが求められており，RFID (Radio Frequency IDentification) がサプライチェーンにおいて商品の “真贋保証書” としての役割を果たす技術として注目されている [3]．RFID の付加された商品は各サプライチェーン・パーティの保持する RFID リーダにより読み取られ，商品毎に入出荷イベントが記録・管理される．これにより商品が製造者から配送業者を経て小売

[1] 慶應義塾大学大学院
Graduate School of Keio University, Japan
[2] 慶應義塾大学 理工学部 情報工学科
Department of Information and Computer Science, Keio University, Japan

店まで届けられる際に，偽物が混入していないことを確認できる．この時，RFID に記述される商品識別コードを EPC (Electronic Product Code) と呼び，商品の種類毎に様々なフォーマットが EPCglobal GS1 (Global Standard One) によって定義されている [4]．例えば，SGTIN (Serialized Global Trade Item Number) および SSCC (Serial Shipping Container Code) はそれぞれ製品の識別，コンテナの識別に使用される．

しかしながら，攻撃者が正しい EPC を読み取り，それを偽物の商品に付加した場合，商品の真贋判定を行うことはできなくなることが指摘されている [5]．パーティ間の輸送中に攻撃者が商品に付加された RFID にアクセスできる可能性があることから，安全に EPC を配送する必要がある．その手法の一つとして，EPC を共通鍵暗号方式で暗号化した上で，共通鍵を閾値秘密分散を用いて複数のシェアに分割し，そのシェアを RFID タグの空きスペース (User Memory 領域) に書き込む方式が提案されている [6]．以降，この研究を基にした改良方式が種々検討されている ([7–15] 等)．

しかしながら，実用を想定した場合に検討すべき課題がいくつか存在する．そのうちのひとつとして，どの暗号化方式を用いればよいかという問題がある．暗号化方式の検討が必要な理由のひとつとして，RFID タグのメモリ領域が限られていることが挙げられる．RFID タグの EPC 格納領域は通常 64 ビットから 256 ビット程度であり，ブロック長が 256 ビットである AES (Advanced Encryption Standard) は不適である．さらに，メモリ領域が限られているため，32 および 64 ビットといった比較的短い共通鍵を使用することが考えられる [6]．このとき，攻撃者が全数探索により正しい共通鍵を特定する可能性がある．攻撃者は正しい共通鍵を使用した際のみ復号文が EPC のフォーマットに従うことを判断基準に候補となる鍵をひとつひとつ試すため，いずれ正しい共通鍵を特定できる．したがって，EPC の暗号化にどの暗号化方式を用いるべきかを検討することは実用を想定した場合に重要な問題である．

本論文では，FTE (Format Transforming Encryption) を暗号化方式として用いた安全な EPC 配送を提案する．FTE は平文と暗号文ののメッセージフォーマットをそれぞれ指定できる共通鍵暗号方式である [16–18]．FTE を用いる際に，平文および暗号文のフォーマットとして EPC のメッセージフォーマットを指定することで，格納領域の問題を解決できるだけでなく，攻撃者はどの鍵候補を試したとしても復号文が EPC のフォーマットとなることから正しい共通鍵を特定するのを困難とする．指定するメッセージフォーマットと攻撃者の正しい共通鍵の特定できる確率について考察を行い，攻撃者の輸送される商品についてどの程度知識があるのかによって，指定すべきフォーマットが異なることを示す．

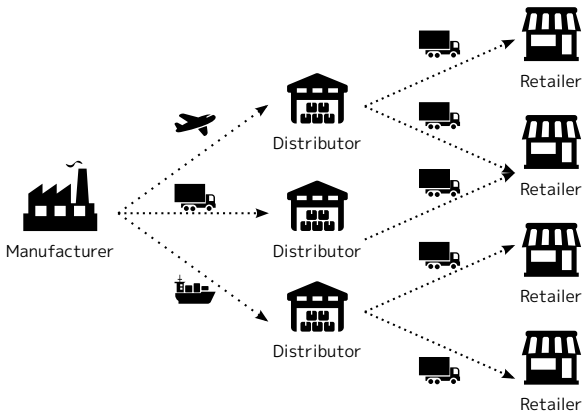

図 1 想定するサプライチェーン．
Fig. 1 Assumed supply chain structure.

論文の構成は以下の通りである．前提とするシステムモデルを 2 章で説明し，関連研究を 3 章で紹介する．提案方式を 4 章で述べ，セキュリティ分析を 5 章で行う．特性評価を 6 章にて示し，最後に 7 章で結論をまとめる．

2. システムモデル

2.1 サプライチェーン・モデル

本論文において想定するサプライチェーンは製造者 (Manufacturers)，配送業者 (Distributors)，小売店 (Retailers) の 3 つのパーティから構成される．**図 1** に想定するサプライチェーンを示す．製造者は商品を製造，梱包し，配送業者に配送する．配送業者は一旦商品を取り出し，それぞれの小売店に向けて再梱包し，配送する．小売店は商品の在庫を保管し，店頭に並べて販売する．さらに各パーティは EPCglobal C1G2 (Class 1 Generation 2) に準拠した UHF (Ultra High Frequency)RFID リーダを使用し，入出荷時に商品に付加されたタグの読取を行う．

製造者は商品の種類に合わせて，使用する EPC フォーマットを適切に選択する [4]．代表的なフォーマットには SGTIN，SSCC，CPI (Component Part Identifier)，GID (General Identifier) などがある．例えば，SGTIN には製造者コード，商品コード，シリアルナンバーなどが含まれており，製品に付加される．EPC のフォーマットは 8 ビットのヘッダが含まれ，これにより RFID リーダによって読み取られた EPC がどのフォーマットであるかを判別できる．例えば，96 ビットの SGTIN である SGTIN-96 は “11000000” が指定されている．

2.2 攻撃者モデル

攻撃者のゴールは製造者の製造する商品の正しい EPC を得ることである．そのために，攻撃者はサプライチェーンのパーティ間で商品が輸送される経路において商品に付加されたタグを読み取るモデルを想定する [6,7,19]．**図 2** および**図 3** に攻撃者の存在領域および位置の例を示す．本

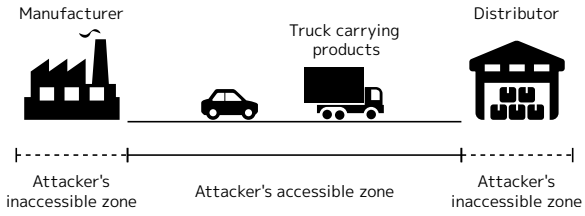

図 2　攻撃者の商品に近付ける領域.

Fig. 2　Attacker's accessible zone.

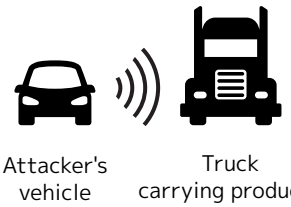

図 3　攻撃者の位置の例.

Fig. 3　An example of attacker's position in the accessible zone.

論文では，攻撃者はトラック内の全商品のタグを読み取れると仮定する．これは一部の商品のタグのみ読み取れる従来の仮定 ([6,7,19]) よりも弱い．

3. 関連研究

本章では，上述の攻撃者に対してこれまで検討されてきた防御策を関連研究として紹介し，さらに未解決の課題を列挙する．

攻撃者のゴールは正しい EPC を得ることにあるため，暗号化により安全に EPC を配送する方式が検討されてきた ([7–15] 等)．これらのほとんどは，商品の EPC を共通鍵暗号方式で暗号化しておき，その共通鍵をどのように次のパーティに配送するかについて検討している．その中でも，共通鍵を閾値秘密分散法 [20] によって複数の部分鍵に分割し，その部分鍵を商品タグの空きスペース (User Memory 領域等) に書き込む方式が多く検討されている ([7–15] 等)．これらの方式の利点は，共通鍵配布のために別途サーバを立てる，もしくは安全な通信路を用意する必要がない点である．最初に Langheinrich と Marti によって Shamir の閾値秘密分散法を安全な EPC の配送に用いた手法が提案された [7]．Juels 等は Shamir の方式の場合，部分鍵が元の共通鍵と同じになることを問題に挙げ，代わりにリードソロモン符号化を用いた閾値秘密分散法を提案している [6]．さらに Lv 等は計算量削減のために XOR (exclusive OR) を用いた閾値秘密分散法を提案している [10]．

上述のように，安全な EPC の配送のために多くの手法が検討されている．しかしながら，実用を想定した場合に解決されていない課題が存在する．そのうちのひとつとして，RFID のメモリ領域が限られていることにより，暗号化された EPC が EPC 格納領域に収まらない可能性がある点がある．例えば AES のようにブロック長が 256 ビットの共通鍵暗号方式を用いた場合，暗号化された EPC は 256 ビットの倍数の長さとなる．しかしながら，一般的な EPC C1G2 に準拠した RFID タグの EPC 領域の 32 ビットから 256 ビット程度であることを考えると AES の利用は適さない．この問題を緩和するために Blowfish のようにブロック長が 64 ビット程度と短い方式によって改良が検討されているが，EPC 長はフォーマット毎に異なっているため完全に解決しているとは言えない．さらに 2 つ目の問題点として，同様にメモリ領域の大きさの制限により鍵長の短い共通鍵を使用する場合に，攻撃者が正しい共通鍵を特定できる可能性がある点が挙げられる．したがって，これらの課題を解決する手法が求められている．これまで安全な EPC 配送のためにどの暗号化方式を使用するべきかについての議論はされてこなかった．本研究では，適切な暗号化方式を選択することによってこれらの課題を解決できることを示す．

4. 提案方式

本論文では，共通鍵暗号方式に FTE を用いた安全な EPC 配送手法を提案する．FTE は平文と暗号文ののメッセージフォーマットをそれぞれ指定できる共通鍵暗号方式である [16–18]．FTE を用いる際に，平文および暗号文のフォーマットとして EPC のメッセージフォーマットを指定することで，格納領域の問題を解決できるだけでなく，攻撃者はどの鍵候補を試したとしても復号文が EPC のフォーマットとなることから正しい共通鍵を特定するのを困難とする．ここで $\mathcal{F}_{\text{plain}}$ および $\mathcal{F}_{\text{cipher}}$ がそれぞれ平文および暗号文のフォーマットを表すものとし，それぞれ PCRE (Perl Compatible Regular Expression) 形式の正規表現で表す．例として，SGTIN-96 のタグが配送される場合，$\mathcal{F}_{\text{plain}} = 00110000(0|1)\{88\}$ および $\mathcal{F}_{\text{cipher}} = (0|1)\{96\}$ とすることが考えられる．ただし，$(0|1)\{96\}$ は PCRE 表現において 96 ビットのバイナリ系列を表す．このように FTE に対して適切に $\mathcal{F}_{\text{plain}}$ および $\mathcal{F}_{\text{cipher}}$ を指定することで，上述の 1 つ目の課題である RFID の EPC 領域に収まるように暗号化を行うことを可能とする．さらに，攻撃者がいずれの鍵候補を試して得られた復号文は $\mathcal{F}_{\text{plain}}$ に従うため，鍵長が短かったとしてもいずれの鍵が正しい鍵であるかを特定することは困難である．

以下ではまず FTE の概要を紹介し，提案方式における各パーティの手続きを述べる．最後に指定すべきフォーマット $\mathcal{F}_{\text{plain}}$ および $\mathcal{F}_{\text{cipher}}$ についての考察および本方式の利点および欠点を述べる．

4.1 FTE の概要

FTE は元々，正規表現型 DPI (Deep Packet Inspection) で防御されたネットワークに対してブロックされるべきプロトコルのメッセージを通過させることを目的として開発された [16]．すなわち，本来ならば通過できないプロトコル

のメッセージを通過できるプロトコルのフォーマットの形式となるように暗号化を行う．したがって，FTE では平文と暗号文のメッセージフォーマットを正規表現で指定するのが一般的である ([16–18] 等)．FTE の基本的なアイディアは，“rank-encipher-unrank” と呼ばれ，フォーマット $\mathcal{F}_{\text{plain}}$ に従う平文 m は $i \in \mathbb{Z}_{|L(\mathcal{F}_{\text{plain}})|} = \{0, 1, \ldots, |L(\mathcal{F}_{\text{plain}})|-1\}$ として $\mathcal{F}_{\text{plain}}$ が構成する言語空間 $L(\mathcal{F}_{\text{plain}})$ において順序付けされ，i は $j \in \mathbb{Z}_{|L(\mathcal{F}_{\text{cipher}})|}$ に暗号化される．最後に $L(\mathcal{F}_{\text{cipher}})$ の j 番目の言葉が暗号文として出力される．復号はこれと逆のプロセスにより行われる．

文献 [18] の表記を参考にし，本論文において FTE は暗号化・復号アルゴリズムのペア $(\mathsf{Enc}, \mathsf{Dec})$ として表されるものとし，Enc および Dec は以下のように定義される．

- Enc は共通鍵 $\kappa \in K$，平文フォーマット $\mathcal{F}_{\text{plain}}$，暗号文フォーマット $\mathcal{F}_{\text{cipher}}$，平文 $M \in L(\mathcal{F}_{\text{plain}})$ を入力とし，暗号文 $C \in L(\mathcal{F}_{\text{cipher}})$ もしくは “failure” シンボル $\perp$ を出力する確率的もしくは決定的アルゴリズムである．ここで $\mathsf{Enc}_\kappa^{\mathcal{F}_{\text{plain}} \to \mathcal{F}_{\text{cipher}}}(M)$ を共通鍵 κ，平文フォーマット $\mathcal{F}_{\text{plain}}$，暗号文フォーマット $\mathcal{F}_{\text{cipher}}$ を用いて平文 M を FTE 暗号化する関数と定義する．
- Dec は共通鍵 $\kappa \in K$，平文フォーマット $\mathcal{F}_{\text{plain}}$，暗号文フォーマット $\mathcal{F}_{\text{cipher}}$，暗号文 $C \in L(\mathcal{F}_{\text{cipher}})$ を入力とし，C が $\mathsf{Enc}_\kappa^{\mathcal{F}_{\text{plain}} \to \mathcal{F}_{\text{cipher}}}(M)$ を満たすような平文 M を出力する決定的アルゴリズムである．Enc と同様に，$\mathsf{Dec}_\kappa^{\mathcal{F}_{\text{cipher}} \to \mathcal{F}_{\text{plain}}}(C)$ を共通鍵 κ，平文フォーマット $\mathcal{F}_{\text{plain}}$，暗号文フォーマット $\mathcal{F}_{\text{cipher}}$ を用いて暗号文 C を FTE 復号する関数と定義する．

4.2 各パーティの手続き

上記の FTE の定義に基づき，商品が製造者によって製造されてから小売店まで運搬される手続きを示す．

まず製造者は n_P 個の製品を製造し，各々に対して EPC を割り当てる．さらに，共通鍵 κ を生成し，各 EPC を $\mathsf{Enc}_\kappa^{\mathcal{F}_{\text{plain}} \to \mathcal{F}_{\text{cipher}}}(\text{EPC})$ によって暗号化する[*1]．共通鍵の配送はこれまでに提案されている任意の方式 ([6, 10, 14] 等) を使用できるが，ここでは，文献 [14] の方式を使用した場合について説明する．より具体的には，κ を Shamir の (n_P, τ)-閾値秘密分散法で部分鍵 $\{S_1, S_2, \ldots, S_{n_P}\}$ に分割した上で，暗号化された EPC と共に商品に付加された RFID タグに書き込む．さらに攻撃者が全ての部分鍵を読み取ったとしても正しい共通鍵を特定できないように，n_D 個の偽のタグも同梱する．このとき偽のタグは商品の受領者 (ここでは配送業者もしくは小売店) が認識できるように，いずれの商品にも付加せずそのまま同梱する．

受領者は商品を受領するとまず n_D 個の偽のタグを除き，n_P 個の商品のみの暗号化された EPC と部分鍵を RFID リーダで読み取る．κ を Shamir の (n_P, τ)-閾値秘密分散法で復元し，$\mathsf{Dec}_\kappa^{\mathcal{F}_{\text{cipher}} \to \mathcal{F}_{\text{plain}}}(\cdot)$ を用いて復号を行い，各商品の正しい EPC を得る．さらに商品を別のパーティに配送する際には，新しく鍵 κ を生成した上で，上記の製造者と全く同じ手順を踏むことで EPC および共通鍵を配送する．

[*1] ただし $\mathcal{F}_{\text{plain}}$ および $\mathcal{F}_{\text{cipher}}$ の選択法については次節で説明する．

4.3 $\mathcal{F}_{\text{plain}}$ および $\mathcal{F}_{\text{cipher}}$ の選択法

FTE を EPC の暗号化に用いる目的のひとつとして，暗号化された EPC が EPC 格納領域に収まるようにすることが挙げられる．そのために，$\mathcal{F}_{\text{plain}}$ を元の EPC の空間を完全に覆うように選択し，$\mathcal{F}_{\text{cipher}}$ を用いる RFID タグの EPC 格納領域に収まるように指定すればよい．しかしながら，$\mathcal{F}_{\text{plain}}$ および $\mathcal{F}_{\text{cipher}}$ をより詳細に選択することで，攻撃者が輸送中の RFID タグを読み取ったとしても元の正しい EPC を復元することを困難にすることが可能である．FTE の暗号化・復号アルゴリズム $(\mathsf{Enc}, \mathsf{Dec})$ は，共通鍵 κ だけでなく，$(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ も必要であることから，攻撃者がこの $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ のペアを知らない場合，正しい EPC を特定することはより困難となる．このことから，攻撃者が $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ を知り得るかどうかによって，どのように $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ を選択するかが変わる．以下では，攻撃者の $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ に対する知識の有無に応じてそれぞれ議論を行う．

4.3.1 攻撃者が $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ を知らない場合

この場合，攻撃者は共通鍵 κ だけでなく，$(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ も推測する必要がある．この推測を困難にするためには，元の網羅したい EPC 空間に加え，いくつかの使われないフォーマットを $\mathcal{F}_{\text{plain}}$ に含ませることが考えられる．例えば，$\mathcal{F}_{\text{plain}}$ が 64 ビットと 96 ビットのバイナリ系列を網羅している場合に，1，10，1{72} といった使用されないフォーマットを付加し，$\mathcal{F}_{\text{plain}} = (0|1)\{64\}|(0|1)\{96\}|1|10|1\{72\}$ と指定することが考えられる．攻撃者は例え余計なフォーマットが $\mathcal{F}_{\text{plain}}$ に含まれているとわかったとしても，どのフォーマットが含まれているかを推測することは困難である．ここで $\mathcal{F}_{\text{plain}}$ を選択する上で 2 つの点を考慮に入れる必要がある．1 つ目は，$\mathcal{F}_{\text{plain}}$ に指定する正規表現の空間が大きくなる程，暗号化に必要なメモリ量および計算時間が増加する点である．したがって，余計なフォーマットの構成する言語空間が大きくなりすぎないようにする必要がある．例えば $(0|1)\{64, 96\}$ のように任意の 64 ビットから 96 ビットまでのバイナリ系列を表現する空間の指定は避けるべきである．2 つ目として，$\mathcal{F}_{\text{plain}}$ の言語空間の大きさは，$\mathcal{F}_{\text{cipher}}$ のそれよりも小さい必要がある．これは平文が $\mathcal{F}_{\text{cipher}}$ の暗号文に対して 1 対 1 対応するように写像されるためである．

一方で，$\mathcal{F}_{\text{cipher}}$ は $\mathcal{F}_{\text{plain}}$ よりも選択の幅が少ない．これは，攻撃者は暗号化された $\mathcal{F}_{\text{cipher}}$ に従う EPC を見るこ

とができるためである．攻撃者は正しい $\mathcal{F}_{\text{cipher}}$ に従う全ての暗号化された EPC を見ることは現実的には不可能であるが，本論文では $\mathcal{F}_{\text{cipher}}$ は攻撃者にとって既知として扱う．

4.3.2 攻撃者が $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ を知っている場合

現実的には，攻撃者が $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ を知っている場合が考えられる．例えば，サプライチェーン全体で使用する $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ が固定の場合，$(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ はパブリックなものとして扱う必要がある．このとき，攻撃者が $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ を知らない場合と比較して，よりタイトに $\mathcal{F}_{\text{plain}}$ を指定すべきである．より具体的には，$\mathcal{F}_{\text{plain}} = 00110000(0|1)\{88\}$ とすることにより，任意の鍵で復号された EPC が SGTIN-96 の形式となる．これにより，攻撃者はどの鍵候補が正しい鍵であるかを特定することが困難となる．これは HE (Honey Encryption) と似た考えである．HE は任意のパスワードによって復号された平文がいずれも正しいように見える，パスワードを用いた暗号化方式のひとつである [21]．しかしながら，攻撃者が正しい EPC を特定することが困難な $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ のペアについては不明確であり，今後の研究課題である．

5. セキュリティ分析

FTE を EPC の暗号化に用いる提案方式に対してセキュリティ分析を行う．攻撃者の目的は，暗号化されている EPC を復号し，正しい EPC を特定することである．攻撃者は n_P 個の暗号化された EPC および $\mathcal{F}_{\text{cipher}}$ を与えられるが，正しい $\mathcal{F}_{\text{plain}}$ および κ を知らないと仮定する．したがって複数の $\mathcal{F}_{\text{plain}}$ および κ を試す．さらに，攻撃者は n_E 個の余計な要素が含まれた $\mathcal{F}_{\text{plain}}$ の候補集合 $\{\mathcal{F}_{\text{plain}}\}$ を与えられ，正しい $\mathcal{F}_{\text{plain}}$ ならびに共通鍵 κ を特定する．この条件の下では，攻撃者は以下のオラクルにアクセスできる．

- $\mathcal{O}_{\mathsf{KeyGen}}(K)$: 鍵候補 $\hat{\kappa} \in K$ を返す．
- $\mathcal{O}_{\mathsf{Enc}}(M, \hat{\kappa}, \hat{\mathcal{F}_{\text{plain}}}, \mathcal{F}_{\text{cipher}})$: パラメータ $(\hat{\kappa}, \hat{\mathcal{F}_{\text{plain}}}, \hat{\mathcal{F}_{\text{cipher}}})$ により平文 M を FTE 暗号化し，暗号文 C を返すオラクル．ただし $M \notin \hat{\mathcal{F}_{\text{plain}}}$ の場合, failure シンボル $\perp$ を返す.
- $\mathcal{O}_{\mathsf{Dec}}(C, \hat{\kappa}, \hat{\mathcal{F}_{\text{plain}}}, \mathcal{F}_{\text{cipher}})$: パラメータ $(\hat{\kappa}, \hat{\mathcal{F}_{\text{plain}}}, \mathcal{F}_{\text{cipher}})$ により暗号文 C を FTE 復号し，平文 M を返すオラクル. ただし $C \notin \mathcal{F}_{\text{cipher}}$ の場合, failure シンボル $\perp$ を返す.
- $\mathcal{O}_{\mathsf{Choose}}(\{\mathcal{F}_{\text{plain}}\})$: 集合 $\{\mathcal{F}_{\text{plain}}\}$ からランダムに $\hat{\mathcal{F}_{\text{plain}}}$ を返す.

上記のオラクルを使用し，攻撃者は以下のチャレンジを行う．

チャレンジ：**Cha**
入力：n_P 個の正しい EPC$\{M_i\}$，n_P 個の暗号化された EPC$\{C_i\}$，鍵空間 K，平文フォーマットの候補集合 $\{\mathcal{F}_{\text{plain}}\}$，暗号文フォーマット $\mathcal{F}_{\text{cipher}}$
手続：
$\hat{\kappa} \leftarrow \mathcal{O}_{\mathsf{KeyGen}}(K)$
$\hat{\mathcal{F}_{\text{plain}}} \leftarrow \mathcal{O}_{\mathsf{Choose}}(\{\mathcal{F}_{\text{plain}}\})$
$\hat{M}_i \leftarrow \mathcal{O}_{\mathsf{Dec}}(C_i, \hat{\kappa}, \hat{\mathcal{F}_{\text{plain}}}, \mathcal{F}_{\text{cipher}})$
出力：全ての $i \in [1, n_P]$ に対し，$\hat{M}_i = M_i$ ならば 1 を返す．そうでなければ 0 を返す．

Claim 1. 上記のチャレンジに対し, $\mathcal{F}_{\text{plain}}$ を知らない攻撃者の成功確率は以下のようにバウンドされる．

$$\Pr[\mathbf{Cha} \Rightarrow 1] < 2^{-(n_E + |\kappa|)}.$$

Proof. 全ての正しい EPC を得るためには，正しい κ および $\mathcal{F}_{\text{plain}}$ が必要となる．このとき攻撃者は κ および $\mathcal{F}_{\text{plain}}$ に対して 1 つずつ試す必要がある. したがって正しい κ および $\mathcal{F}_{\text{plain}}$ を見つける確率は $2^{-(n_E+|\kappa|)}$ となる． □

6. 特性評価

提案方式の暗号化および復号に掛かる計算時間およびメモリ量を評価する．FTE の計算には libFTE[*2]を用いる．Ubuntu 12.04 の走るシングルコア CPU および 1GB の RAM を持つ仮想環境において提案方式を実装し，各 1,000 回の計算を行うベンチマークスクリプトによって得られた平均値を評価する．各試行において，元の EPC は $\mathcal{F}_{\text{plain}}$ に従うようにランダムに生成する．

表 1 に 1EPC を暗号化/復号するのに必要な計算時間およびメモリ量を示す．この表において，“Case #” は試行した $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ のインデックスを表し，encryption, decryption, required memory はそれぞれ暗号化に要した時間，復号に要した時間，必要なメモリ量を示す．

まず，暗号文の長さによる計算時間とメモリ量を比較する．1 から 4 までの $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ に対し，暗号化および復号に必要な計算時間およびメモリ量は線形に増加していることがわかる．さらに暗号化および復号のいずれも 1 ms オーダである．このことから，商品数が 1,000 個程度が想定されるサプライチェーンにおいて，暗号化および復号に 1 秒程度あれば十分であり，実用上の問題はないと言える．さらに必要なメモリ量も $\mathcal{F}_{\text{plain}}$ および $\mathcal{F}_{\text{cipher}}$ の大きさに応じて増加していることがわかる．また，メモリ量もキロバイトオーダであり，実用的であることがわかる．

次に，攻撃者が $\mathcal{F}_{\text{plain}}$ を知らない場合を考える．すなわち，(i) 余計な正規表現を加えない場合と (ii) 加えた場合で比較を行う． Case 5 および 6 に着目すると，余計な正規表現を加えた場合，わずかに暗号化および復号に掛かる時間が増加することがわかる．一方，Case 7 の結果からわかる通り，不必要に余計な正規表現を加えた場合，$\mathcal{F}_{\text{plain}}$ の言語空間が大きくなり，暗号化および復号に時間を要す

[*2] https://libfte.org

表 1　1EPC を暗号化/復号するのに必要な計算時間およびメモリ量.
Table 1　Computation time and required memory to encrypt/decrypt an EPC.

Case #	$\mathcal{F}_{\text{plain}}$	$\mathcal{F}_{\text{cipher}}$	Encryption [ms]	Decryption [ms]	Required Memory [KB]
1	(0|1){64}	(0|1){64}	0.55	0.65	265
2	(0|1){64}	(0|1){96}	0.73	0.74	343
3	(0|1){64}	(0|1){128}	0.85	0.87	440
4	(0|1){64}	(0|1){192}	1.11	1.05	663
5	(0|1){64}|(0|1){96}	(0|1){128}	1.04	1.09	558
6	(0|1){64}|(0|1){96}|1|10|1{72}	(0|1){128}	1.09	1.11	642
7	(0|1){64, 96}	(0|1){128}	3.92	3.94	2,078
8	00110000(0|1){88}	(0|1){96}	0.71	0.75	412
9	(00101100|00101101)(0|1){88}	(0|1){96}	0.73	0.79	408
10	(00101100|00101101|$\cdots$|00111100)(0|1){88}	(0|1){96}	0.73	0.78	444

るだけでなく，メモリ量も 3 倍以上必要となることがわかる．このことから，不必要に余計な正規表現を $\mathcal{F}_{\text{plain}}$ に加えることは避けるべきであると言える．

最後に，攻撃者が $\mathcal{F}_{\text{plain}}$ を知っている場合を考える．Case 8 においては SGTIN-96 のフォーマットのみ，Case 9 では GDTI-96 および PGSRN-96 のみ，Case 10 では全ての 96 ビット用の EPC フォーマットを使用する場合を表している．これらの結果から，いずれの場合も計算時間およびメモリ量に大きな差がないことがわかる．これは Case 8 から 10 に関して，$\mathcal{F}_{\text{plain}}$ における言語空間の大きさにほとんど違いがないためだと考えられる．いずれの結果においても，ミリ秒オーダで計算が完了し，必要なメモリ量もキロバイトオーダであるため，実用可能であると言える．

7. 結論

本論文では，RFID サプライチェーンにおいて EPC を安全に配送するために，FTE を暗号化方式に用いる手法を提案した．FTE を用いることにりょい，暗号化された EPC が使用する RFID タグの EPC 格納領域に収まることを保証するだけでなく，$(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ を適切に選択することで攻撃者が正しい EPC を特定することを困難にする．FTE の暗号化・復号アルゴリズム $(\mathsf{Enc}, \mathsf{Dec})$ は，共通鍵 κ だけでなく，$(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ も必要であることから，攻撃者が $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ を知り得るかどうかによって，どのように $(\mathcal{F}_{\text{plain}}, \mathcal{F}_{\text{cipher}})$ を選択するかが変わることを示した．攻撃者が $\mathcal{F}_{\text{plain}}$ を知らない場合，$\mathcal{F}_{\text{plain}}$ に含むべきフォーマットに加えて余計なフォーマットを取り入れるべきである．一方，知っている場合は $\mathcal{F}_{\text{plain}}$ に余計なフォーマットを入れず，含むべき EPC のフォーマットのみを指定することで攻撃者が正しい鍵を特定する確率を下げることを提案した．計算機シミュレーションにより様々な $\mathcal{F}_{\text{plain}}$ を想定した場合における計算時間および必要なメモリ量を評価し，いずれもミリ秒オーダで暗号化/復号が完了し，必要なメモリ量もキロバイトオーダであることを示した．

参考文献

[1] FDA: Combating Counterfeit Drugs, A Report of the Food and Drug Administration (2004).
[2] Avery, P. et al.: *The economic impact of counterfeiting and piracy*, OECD Publishing (2008).
[3] Zanetti, D., Capkun, S. and Juels, A.: Tailing RFID tags for clone detection, *Proc. of Network and Distributed System Security Symposium (NDSS)* (2013).
[4] EPCglobal: EPC Tag Data Standard (TDS) (2014).
[5] Staake, T., Thiesse, F. and Fleisch, E.: Extending the EPC network: the potential of RFID in anti-counterfeiting, *Proc. of ACM Symposium on Applied Computing*, pp. 1607–1612 (2005).
[6] Juels, A., Pappu, R. and Parno, B.: Unidirectional Key Distribution Across Time and Space with Applications to RFID Security, *Proc. of USENIX Security Symposium*, pp. 75–90 (2008).
[7] Langheinrich, M. and Marti, R.: Practical minimalist cryptography for RFID privacy, *Systems Journal, IEEE*, Vol. 1, No. 2, pp. 115–128 (2007).
[8] Langheinrich, M. and Marti, R.: RFID privacy using spatially distributed shared secrets, *Ubiquitous Computing Systems*, Springer, pp. 1–16 (2007).
[9] Cai, S., Li, T., Ma, C., Li, Y. and Deng, R. H.: Enabling secure secret updating for unidirectional key distribution in RFID-enabled supply chains, *Information and Communications Security*, Springer, pp. 150–164 (2009).
[10] Lv, C., Jia, X., Lin, J., Jing, J. and Tian, L.: An efficient group-based secret sharing scheme, *Information Security Practice and Experience*, Springer, pp. 288–301 (2011).
[11] Alfaro, J. G., Barbeau, M. and Kranakis, E.: Proactive threshold cryptosystem for EPC tags, *Ad hoc & sensor wireless networks*, Vol. 12, No. 3-4, pp. 187–208 (2011).
[12] Li, T., Li, Y. and Wang, G.: Secure and practical key distribution for RFID-enabled supply chains, *Security and Privacy in Communication Networks*, Springer, pp. 356–372 (2012).
[13] Abughazalah, S., Markantonakis, K. and Mayes, K.: Enhancing the Key Distribution Model in the RFID-Enabled Supply Chains, *Proc. of International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 871–878 (2014).
[14] Toyoda, K. and Sasase, I.: Secret Sharing Based Unidirectional Key Distribution with Dummy Tags in Gen2v2 RFID-enabled Supply Chains, *Proc. of IEEE Interna-*

tional Conference on RFID (2015).

[15] Toyoda, K. and Sasase, I.: Illegal Interrogation Detectable Products Distribution Scheme in RFID-Enabled Supply Chains, *IEICE Transactions on Communications*, Vol. E99-B, No. 4, pp. 820–829 (2016).

[16] Dyer, K. P., Coull, S. E., Ristenpart, T. and Shrimpton, T.: Protocol Misidentification Made Easy with Format-transforming Encryption, *Proc. of ACM Conference on Computer and Communications Security (CCS)*, pp. 61–72 (2013).

[17] Luchaup, D., Shrimpton, T., Ristenpart, T. and Jha, S.: Formatted Encryption Beyond Regular Languages, *Proc. of ACM Conference on Computer and Communications Security (CCS)*, pp. 1292–1303 (2014).

[18] Luchaup, D., Dyer, K. P., Jha, S., Ristenpart, T. and Shrimpton, T.: LibFTE: a toolkit for constructing practical, format-abiding encryption schemes, *Proc. of USENIX Security Symposium*, pp. 1–15 (2014).

[19] Cai, S., Li, Y. and Zhao, Y.: Distributed path authentication for dynamic RFID-enabled supply chains, *Information Security and Privacy Research*, Springer, pp. 501–512 (2012).

[20] Shamir, A.: How to share a secret, *Communications of the ACM*, Vol. 22, No. 11, pp. 612–613 (1979).

[21] Juels, A. and Ristenpart, T.: Honey Encryption: Security Beyond the Brute-Force Bound, *Advances in Cryptology–EUROCRYPT*, Springer, pp. 293–310 (2014).