Computer Security Symposium 2017
23 - 25 October 2017

# スマートシティ IoT における情報流通制御方式の提案

佐々木 貴之[†1] 濱本 亮[†1] 森田 佑亮[†1] 三好 一徳[†1] 小林 俊輝[†1]

**概要**： IoT のコンセプトをスマートシティに適用し，省エネやより良いサービスを提供する試みが始まっている．IoT 化された都市基盤は便利である反面，既存の IT システムと同様に，サイバー攻撃にさらされるリスクがある．特に，電気ガス水道のようなライフラインや交通のような都市基盤が攻撃されると，多くの市民の生活が影響を受けるため，これを確実に防ぐ必要がある．また，監視カメラによる治安の向上や，ヘルスケアサービスの場合には，カメラ画像や医療データのようなプライバシ情報を確実に守る必要がある．これらの課題を解決するために，本論文では情報の生成から消滅までを一貫して管理する情報流通制御方式を提案する．本方式の特徴は，IoT プラットフォームのコンポーネント間で情報が送受信される際に，PEP がその送受信を監視し，情報の流通履歴を基に PDP が情報の送受信の可否を判断する点にある．加えて，各コンポーネントが情報に署名を付与することによって，履歴情報の改ざんを防止している点にある．さらに，本論文では，初期プロトタイプの実装を示し，実用的であることを示す．

**キーワード**：スマートシティ，IoT

# Information Flow Control for Smart City IoT

Takayuki Sasaki[†1] Ryo Hamamoto[†1] Yusuke Morita[†1]
Kazunori Miyoshi[†1] Toshiki Kobayashi[†1]

***Abstract***: Smart city is a concept to provide better services using IoT techniques. A smart city platform with IoT is useful for citizens, but we can identify risks of cyber attacks as well as existing IT systems. Especially, if critical life lines such as electricity and water services are attacked, the damage affects daily life of citizens. Therefore we have to prevent such attacks. Moreover, smart city IoT would handle privacy data such as pictures of surveillance cameras for public safety and medical data for healthcare services, and such privacy-sensitive data must be protected from the attacks. To mitigate the attacks, we propose an information flow control method for smart city IoT. Our method controls information flows by a PEP that monitors the flows among components of the smart city IoT platform and a PDP that makes decision on the basis of information flow history. We further show an initial prototype and its feasibility.

***Keywords***: Security, Smart city, IoT

## 1. 背景

近年，デバイスをネットワークに接続し，機器同士や機器とクラウドが通信する Internet of Things(IoT)技術を用いて，環境のセンシングや機器の細かい制御を行うことで，質の高いサービスの提供が試みられている．同様の取り組みが，都市の基盤に対しても行われており，スマートシティと呼ばれている．適用先の分野は，行政サービス，交通システム，スマートヘルス，スマートエネルギー等，多岐に渡る[1]．このような IoT 化されたシステムは利便性が高い反面，サイバー攻撃にさらされるリスクがある．例えば，行政サービスや交通システムの停止，スマートヘルスシステムからの個人情報の流出が懸念される．

一般的に，安全なシステムを構築するためには，そのプラットフォームが安全なことと，そのプラットフォーム上で情報が適切に管理されていることの 2 点が求められ，これら要件はスマートシティ IoT でも同じである．これら要件を満たすために，筆者らは，トラステッドコンピューティングを利用したプラットフォームの真正性検証による不正機器・不正ソフトウェアコンポーネントの排除と，そのプラットフォーム上で情報の生成から消滅までを追跡し，一貫した流通制御を行う流通制御技術を開発しており，本論文では，後者の情報流通制御について詳細に述べる．

スマートシティ IoT と，通常の IoT との違いは，複数のサービスが 1 つの IoT 基盤上で動作することである．1 つのサービスごとに基盤を作ることも可能であるが，複数のサービスを 1 つの基盤上で動作させ，複数の情報を基にサービスを構築・提供した方が，価値の高いサービスを提供できる．例えば，交通システムとイベント情報のシステムを組み合わせることにより，より精度の高い交通情報の予測が可能になる[2]．また，ヘルスケアサービスの医療情報と，料理情報提供サービスの連携による，ヘルシーな料理のレコメンドも考えられる．このように，複数のサービスの連携は有用であるが，その反面，情報の管理が難しくなるという課題がある．例えば，プライバシ情報である医療情報は，確実に保護される必要がある．これ以外にも，例

†1 NEC セキュリティ研究所

えば，カメラの監視による治安の向上の場合には，カメラの映像・画像の漏えいや目的外利用が原因でプライバシを侵害する可能性もあり，保護が必要である．

情報の保護だけではなく，スマートシティ IoT 基盤そのものを守るためにも，情報の管理が必要である．トルコのパイプライン爆破[3]では，監視のための Web カメラがハッキングされ，それを踏み台にして，制御システムに侵入したと言われている．この事件の本質的な課題は，Web カメラを用いた監視のシステムと，パイプラインの制御システム間の情報の流通の制御が，適切に実施されていなかった点にある．このように，情報保護とシステム保護の両方の観点から，情報の管理はスマートシティ IoT のセキュリティの要であり，多種多様な情報に応じて適切な管理ポリシを設定することで，これらの事態を回避可能と考えられる．特に，上記の問題は，情報の生成から消滅までの流通を一貫して制御することで解決できる．具体的には，流通を制御して機密性やプライバシ性の高い情報と低い情報の混在による漏洩を防止できる．また，カメラからのハッキングも，カメラで生成された情報が重要な制御システムに到達しないように制御することで防止可能である．このような制御を実現するために，筆者らは，情報の生成から消滅までを一貫して追跡し，情報の流通履歴に基づいてアクセス制御を行うことによって流通を制御する情報流通制御方式を提案する．さらに，本論文では，プロトタイプの実装による実現可能性の評価とパフォーマンスの評価を行う．具体的には，情報流通制御方式を導入した際のオーバヘッドである履歴の記録と流通可否判断について処理時間を測定した結果，1 コンポーネント当たり 32msec であり，実用的な処理時間であった．

## 2. スマートシティのセキュリティ要件

提案方式を説明する前に，スマートシティ IoT のセキュリティ要件を定義し，要件を満たすために防ぐ必要がある攻撃について述べる．加えて，既存のスマートシティ IoT プラットフォームが、要件を満たしているかを分析する．

### 2.1 セキュリティ要件

前章で述べたように，スマートシティ IoT では，情報の保護とシステムの保護のため，情報の生成から消滅までの一貫した管理が有用である．以下では，セキュリティ要件を定義するために，スマートシティ IoT の特徴を洗い出し，その特徴に基づいて，セキュリティ要件を定義する．

**要件１　機密性：**スマートシティ IoT の構成は，一般的な IoT と同じように，デバイス，IoT ゲートウェイ，クラウドと，それらの要素の間のネットワークから構成される．従って，通常の IT のセキュリティが対象としているクラウド部分の制御だけではなく，IoT ゲートウェイや，デバイスまで含めて情報の流通を制御する必要がある．具体的には，情報はセンサ等のデバイスで生成され，IoT ゲートウェイを経由して，クラウドに到達する．そして，クラウドによって処理が行われ，その処理に対応する機器の制御指示が，クラウドから IoT ゲートウェイを経由してデバイスに届く．この一連の作業の間に，情報の処理に関わるコンポーネント以外が，情報を取得できてはならない．ここで，コンポーネントとは，特定の処理を行うソフトウェアやハードウェアを意味する．

**要件２　完全性：**機密性と同様に，上記の一連の流れで，情報の処理に関わるコンポーネント以外が，情報を書き換えることができてはならない．

**要件３　セキュリティ処理の確実な実施：**スマートシティ IoT の特徴として，様々なサービスが 1 つのプラットフォームで動作する．サービスが扱う情報の種類によっては，匿名化や統計処理などの処理を行う必要がある．このような処理を，一律ではなく，必要に応じて，確実に強制する必要がある．また，セキュリティポリシで決められた情報に対する処理は，必ず実施されなければならない．

上述の 3 つ以外のセキュリティ要件として可用性が挙げられるが，可用性は本論文のスコープ外とする．その理由は，可用性を低下させる DoS 攻撃は，情報の流通制御では防ぐことができない場合が多いためであり，コンピュータリソースの使用量の制限など，他の対策を導入する必要がある．

### 2.2 対象とする攻撃

ここでは，上述の 3 要件を満たすために防ぐべき攻撃を整理する．コンポーネント間の情報の受け渡し方法を考えると，コンポーネントが直接情報の送受信を行う場合と，データベース経由で情報の送受信を行う場合が考えられる．さらに，攻撃を 2 つのタイプに分類する．1 つは，情報を元々扱っていたコンポーネントがマリシャスであり，機密性や完全性を侵害する場合である．具体的には，マリシャスなコンポーネントが，本来情報を扱うべきではないコンポーネントに情報を送信することが考えられる．もう 1 つは，元々情報を扱っていなかったコンポーネントが，通信に割り込み，機密性と完全性を侵害する場合である（図 1）．

データベース経由で情報を送受信する場合でも同様に，元々情報を扱うコンポーネントがマリシャスである場合と，情報を扱っていなかったコンポーネントがマリシャスである場合に，情報の完全性と機密性が侵害される場合を考える．

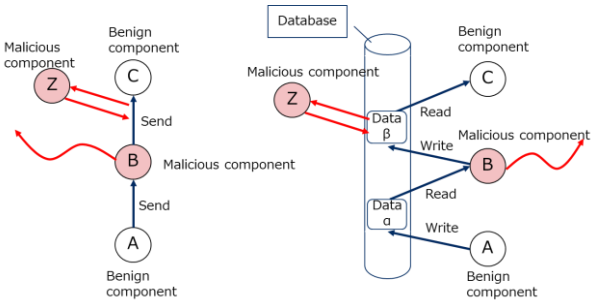


図 1　想定する攻撃

### 2.3　既存のスマートシティ IoT プラットフォームのセキュリティ機能の課題

スマートシティ IoT プラットフォームの 1 つに，欧州で開発されている FIWARE[4]があり，セキュリティのコンポーネントとして，認証・認可のモジュールが用意されている．具体的には，標準のコンポーネントとして，認証を行う KeyRock，Policy Decision Point(PDP)である AuthZForce，Policy Enforcement Point(PEP)である Wilma が用意されており，サブジェクト，オブジェクト，アクションに基づく単純なモデルで流通制御を実施することができる．

このような流通制御方式でも，情報の流通を制御することはできる．ただし，情報の種類（一般情報や，個人情報など）が混じらないように，コンポーネントや記録場所を分離しておく必要があり，多様な情報を扱う IoT では不向きである．さらに，ポリシを細かく指定する必要があるため，デバイスの数や情報の種類が多い IoT の世界では，ポリシの管理コストが増加するという課題がある．具体的には，情報の保存場所を指定する必要があるため，各コンポーネントが情報をどこに格納するのかを理解したうえで，流通制御ポリシを書く必要がある．例えば，図 1 の例のように，コンポーネント A→コンポーネント B→コンポーネント C と情報が流れる場合には，以下のポリシを記述する必要がある．

- コンポーネント A が場所 $\alpha$ に情報を書いてよい
- コンポーネント B が場所 $\alpha$ から情報を読んでよい
- コンポーネント B が場所 $\beta$ に情報を書いてよい
- コンポーネント C が場所 $\beta$ から情報を読んでよい

このポリシの記述にミスがあると，機密性や完全性が侵害される恐れがある．

上記のポリシの記述の問題に加えて，上記のような単純な流通制御ポリシでは，「匿名化されていれば外部に送信可能」というポリシを書くことはできない．なぜなら，過去に情報がどう扱われたかをポリシに記述することができないためである．すなわち，2.1 で述べた要件 3 を満たすことはできない．

## 3.　情報流通制御方式とアーキテクチャ

本章では，前章で述べたセキュリティ要件を満たすために，流通履歴に基づいた情報流通制御方式を提案する．本方式を実現するために，スマートシティ IoT プラットフォームは，情報の流通履歴の収集と流通経路の制御を強制する情報流通制御モジュールと，情報の流通履歴に基づいて流通の可否を判定する流通可否判定モジュールから構成される(図 2)．また，コンポーネント間の情報流通の制御は，図 3 に示すように，情報流通制御モジュールがコンポーネントの情報流通をフックして，流通可否判定モジュールに問い合わせを行う．そして，流通可否判定モジュールはその情報の流通履歴とポリシに基づいて流通の可否を判定し，情報流通制御モジュールは判定が可である場合にのみ，情報送信先のコンポーネントに情報を送信する．詳しくは 3.2 で述べる．

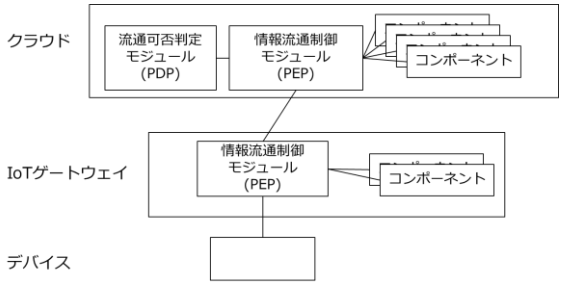


図 2　情報流通制御アーキテクチャ

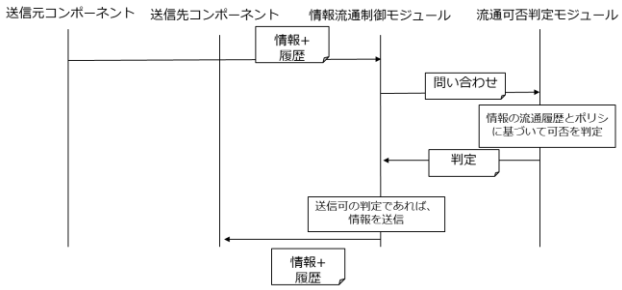


図 3 情報流通制御のシーケンス

### 3.1　情報流通制御方式

情報の生成から消滅まで一貫した流通制御とは，デバイスで生成された情報が，あらかじめ決められた経路を通り，消滅することである．ここで，経路とは，どのデバイスで生成されたか，どのコンポーネントを経由したか，最終的にどのデバイス・サーバで消滅したかを示す情報の履歴である．情報の消滅時に，この情報流通経路である履歴が，予め決められた経路と一致していれば，その情報は，目的外使用されることなく，正しく使用されたと言える．提案方式は，情報の流通履歴に基づいて，あるコンポーネントが次のコンポーネントに情報を渡して良いか否かを判断するため，上述した経路に従って情報が流通することを保証できる．従って，提案方式は，以下のようにセキュリティ要件を満たす．

**要件 1 (機密性):** 情報の流通の経路制御によって，処理に関係のないコンポーネントは，その情報にアクセスするこ

とはできない．従って，機密性の要件を満たすことができる．

**要件２(完全性)：**機密性と同様に，情報の処理に関係のないコンポーネントは，その情報にアクセスできないため，完全性は保証される．ただし，機密性と同様に情報の処理に関連するコンポーネントは改ざんされていないことが保証されている必要があり，remote attestation によるコンポーネントの真正性検証が必要である．なぜなら，内容の書き換え(完全性)に関しては，正当な情報の処理であるのか，不正な処理であるかを判別できないためである．これらのコンポーネントの真正性は，remote attestation で担保することができる．remote attestation については，本論文の対象外であり，例えば Intel SGX を用いた方式[6]が利用可能である．

**要件３(セキュリティ処理の確実な実施)：**確実なセキュリティ処理は，あらかじめ指定されたコンポーネントを情報が確実に通過することで実現できる．例えば，プライバシ保護コンポーネントを情報が確実に通るように制御することによって，その情報に対して確実にプライバシ保護処理が施されることを保証することができる．情報処理の場合も，機密性・完全性と同様に，コンポーネントが改ざんされていないことを，remote attestation によって保障する必要がある．なぜなら，情報流通の制御だけでは，処理が正しく行われているかを判別できないためである．

### 3.2 情報流通制御を実施するアーキテクチャ

上記の情報流通制御を実現するために，Policy Enforcement Point(PEP)である情報流通制御モジュール，Policy Decision Point(PDP)である流通可否判定モジュールから構成される情報流通制御アーキテクチャを提案する(図 2). 以下では，それぞれのモジュールについて説明する．

#### 3.2.1 情報流通制御モジュール

情報流通制御モジュールは，クラウドや各 IoT ゲートウェイに設置される．そして，情報流通制御モジュールは，コンポーネント間の情報の送受信をフックし，監視と制御を行う．具体的には，情報流通制御モジュールは，主に 2 つタスクを行う．

- 情報の流通履歴の収集：情報流通制御モジュールは，情報の送信元コンポーネントと送信先コンポーネントを判定する．ここで，情報とは，デバイスから送信される１つのメッセージを想定している．
- 情報流通の制御：情報流通制御モジュールは，あるモジュールが他のモジュールに発行する情報の要求メッセージと，あるモジュールが他のモジュールに情報を送信するメッセージをフックする．例えば，HTTP の実装の場合，前者が GET メソッド，後者が POST メソッドに相当する．そして，送信元のコンポーネント ID，送信先のコンポーネント ID を基に，流通可否判定モジュールに，許可して良いか否かを問い合わせる．最後に，問い合わせの結果に基づいて，情報の流通の実施，もしくは，流通のブロックを行う．

#### 3.2.2 流通可否判定モジュール

流通可否判定モジュールは，内部に保存されている流通制御ポリシと，情報流通制御モジュールからの問い合わせに含まれている流通履歴情報を基に，問い合わせに対して可否を判断する．情報流通ポリシは，図 4 のように，情報流通経路を表したグラフで記述される．

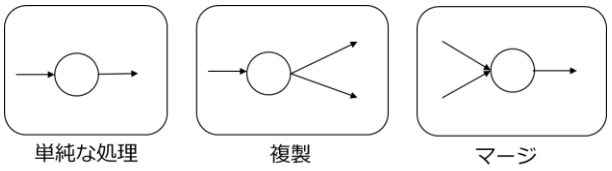


図 4　情報流通のパターン

図 4 左のパターンは，単純な処理であり，１つのコンポーネントが情報を受け取り，処理を行い，次のコンポーネントに転送する場合である．図 4 中央のパターンは，複製であり，１つのコンポーネントが何らかの処理をした後に，2 つのコンポーネントに情報を転送する場合である．図 4 右のパターンは，1 つのコンポーネントが２つのコンポーネントから情報を受け取り，情報をマージする処理を行った後に，次のコンポーネントに情報を転送する処理である．情報流通制御ポリシは，この３パターンを用いて記述される．例えば，各部屋に複数の温度センサがあり，部屋ごとの平均を計算するモジュールと一時間ごとの平均を計算するモジュールを経由し，ヘルスケアサービスと室温管理サービスに提供する場合のポリシは，図 5 のようになる．

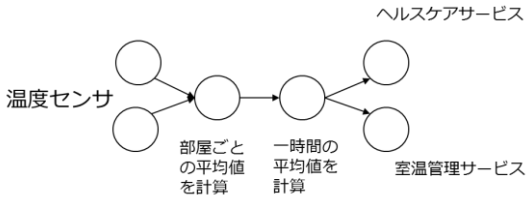


図 5 情報流通ポリシの例

流通可否判定モジュールは，コンポーネント間の情報流通が，このポリシに規定されているかを判断し，規定されている場合のみ，情報の受け渡しを許可する．このとき，情報を渡すコンポーネントだけではなく，情報の過去の履歴も確認を行う．これは，情報の流通経路を一貫して制御するためである．例えば，コンポーネント X からコンポーネント Y への情報の送信であっても，その情報がどこから来たのかを知らなければ判断できない．例えば，カメラの画像を考えると，ビルの屋上に設置されていて風景を写しているお天気カメラと，街頭に設置されていて人々の顔を映しているカメラからの画像は，扱いを変える必要がある．従って，流通可否判定モジュールは，情報の送信元と送信

先だけ確認するのではなく，情報の生成から現在までの履歴と，情報流通制御ポリシを照らし合わせて，流通の可否を判断する．

### 3.3 履歴情報の生成

履歴情報の生成は，情報を処理したコンポーネントが，情報にメタデータとして，タグ(署名)を付与することによって行う．例えば，図 6 の例では，コンポーネント A で情報が生成され，コンポーネント B と C を経由するにしたがって，署名が付与されていく様子を示しており，コンポーネントそれぞれが独自の署名鍵を持ち，情報を次のコンポーネントに送る際に，署名を付けて転送する．情報流通制御モジュールが情報をフックした際，情報は経由したコンポーネントの数だけ署名が付与されている．従って，情報流通制御モジュールは，署名を検証することによって，情報の流通履歴を知ることが出来る．

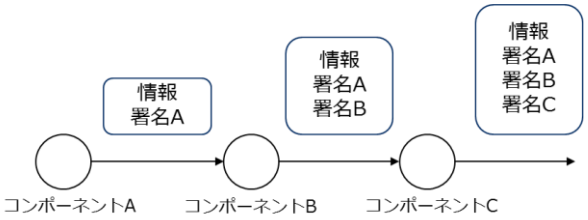


図 6 署名の付与による履歴の記録

### 3.4 TEE を用いた方式の強化

コンポーネントがマリシャスである場合に，履歴情報を偽装できるというセキュリティ課題がある．本方式の場合，１つのコンポーネントがマリシャスであったとしても，そのコンポーネントは他のコンポーネントの署名鍵を保有していないので，情報の流通履歴を詐称することは出来ない．しかし，リスクの１つとして，コンポーネントから鍵が盗まれ，マリシャスなコンポーネントが他のコンポーネントに成りすますことが考えられる．このリスクは，Intel SGX や ARM TrustZone などの Trusted Execution Environment (TEE)を利用して，軽減することができる．TEE は，隔離された環境を構築する技術であり，隔離環境はその外側とメモリ空間が隔離されており，その隔離は CPU が担保している．従って，コンポーネントと署名鍵を TEE の内部に配置し，TEE 内部で情報に対する処理と署名の付与を行うようにすれば，確実に情報はそのコンポーネントで処理され，かつ，署名はそのコンポーネントが付与したことを保証できる．具体的には，図 7 に示すように，TEE で保護された領域に，情報を処理する機能に加えて，署名鍵と署名機能を配置する．そして，メッセージごとに処理を行った後，メッセージに対して，内部に保存された署名鍵を用いて署名を行う．情報流通制御モジュールは，メッセージをコンポーネントから受け取ると，署名の検証を，検証鍵を用いて行う．ここでは，コンポーネントの署名鍵と，それに対応する情報通信制御モジュールが保有する鍵は，事前に管理者が配付済みであるとする．情報流通制御モジュールは，すべての署名が正しいことを確認できた場合に，履歴としてそれらを署名したコンポーネントの ID を履歴として，流通可否を流通可否判定モジュールに問い合わせる．

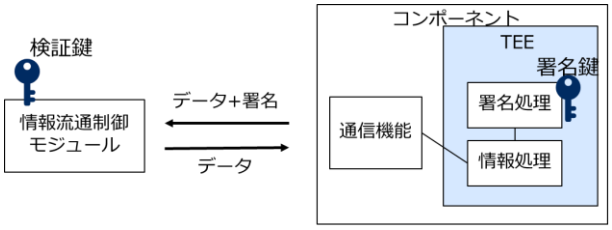


図 7 Intel SGX を用いた流通履歴の記録

## 4. 実装

提案方式のオーバヘッドを評価するために，履歴の記録のためのコンポーネントの署名処理と，流通可否判定モジュールの判断ロジックを実装した．情報流通制御モジュールなど，システム全体の実装は完了しておらず，今後の課題である．

### 4.1 情報流通履歴の記録

情報の流通履歴の記録の方法として，前章で述べた TEE 内で署名を行う方式を実装した．具体的には，TEE として，Intel SGX[6]を用い，SGX によって保護された領域である Enclave 内に情報に署名を付与する機能を実装した．署名は Intel SGX の SDK でサポートされている 3072bit の RSA 暗号署名鍵を用いて行われる．加えて，Enclave 内部と外部で情報をやり取りするために，情報入出力 API を実装した．

### 4.2 流通制御の強制

流通可否判定モジュールを，Neo4J を用いて実装した．Neo4J は，グラフデータベースであり，ノードと，ノード間のリレーションを直接データベースに記録可能である．この特性を用いて，情報流通ポリシを情報流通経路のグラフとして，データベースに記録した．流通の可否の判定は，ポリシのグラフを検索し，部分グラフが見つかれば流通許可，見つからなければ流通拒否とした．例えば，コンポーネント a で生成された情報をコンポーネント b と c 経由でコンポーネント d へと流通させる場合と，コンポーネント x で生成された情報をコンポーネント y 経由でコンポーネント z に流通させる場合の流通ポリシは，図 8 のようになる．

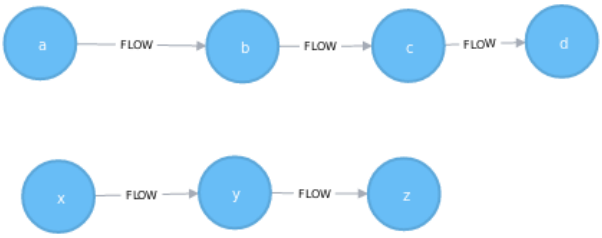


図 8　流通ポリシの例

この時，コンポーネント a で生成され，コンポーネント b を流通した履歴がある情報を，コンポーネント c から d に流通させる際のクエリは，以下のようになる．具体的には，1 行目で 4 個のコンポーネントからなる経路を指定し，2 行目と 3 行目で，コンポーネントの名前を条件として指定している．

**送信可否判断のクエリの例：**

```
(origin)-[]->(hist1)-[]->(current)-[]->(next)
where origin.name="a" and hist1.name="b" and
current. name="c" and next.name="d"
return (origin),(hist1),(current),(next)
```

データベース内にこの部分グラフが見つかれば，情報の c から d への送信を許可し，見つからなければ禁止する．図 8 のポリシがデータベースに格納されている場合は，このクエリは部分グラフにヒットするため，情報の送信は許可される．一方，コンポーネント c からコンポーネント x に情報が渡される場合や，情報の流通履歴が異なる場合は，部分グラフを見つけることができないため，禁止される．

## 5.　評価

前章で述べたプロトタイプの性能を評価した．評価は，Intel Core i3-6100 CPU(3.70GHz)・メモリ 4G Byte の Windows 10 マシン上で、仮想 CPU1 個とメモリ 1G Byte を割り当て，Xubuntu 16.04 をインストールした VirtualBox の VM 上で行った．また，neo4j のバージョンは 3.2.2 を用いた．Intel SGX の SDK のバージョンは 1.9 で、シミュレーションモードを用いた．

まず，コンポーネントに関して，署名にかかる時間を測定した． 1000 バイトのデータに対する署名時間の 10 回の平均値は，15msec であった．今回は，Intel SGX を用いたが，IoT デバイスは ARM 等のより非力な CPU を搭載している場合が多いため，今後はそのようなデバイスでもプロトタイプを作成し，性能を評価する予定である．

次に，流通可否の判定時間を 100 回測し，平均時間を求めた．測定は，4 コンポーネントの長さの流通経路を 1000 パターン生成し，neo4j ストアした状態で行った．その結果，1 クエリあたりの処理時間は平均 17msec であり，その標準偏差は，6msec であった．大量のクエリが集中すると，PDP の neo4j がパフォーマンスのボトルネックになる可能性があるため，デバイスの数や情報の送受信の頻度が多い大規模なシステムでは，情報流通制御モジュール側に，結果をキャッシュする機能が必要であると考えられる．

まだ未実装である情報流通制御モジュールのメッセージのフックと，流通可否判定モジュールへの問い合わせの送受信のオーバヘッドを除くと，1 コンポーネントあたりの提案方式のオーバヘッドは，署名の 15msec+流通可否判定の 17msec である．全体のオーバヘッドは，32msec×経由するコンポーネント数となる．

## 6.　考察

### 6.1　他の流通制御方式との違い

情報の開示範囲を制御する方式として，Multi Level Security(MLS)がある．MLS では，情報やコンピュータに機密レベルが割り当てられており，情報は機密レベルの低いコンピュータから高いコンピュータへのみ流れるようになっている．これによって，機密レベルの高い情報が，機密レベルの低いコンピュータへ流出することを防止している．

Type Enforcement(TE)は，あらかじめプロセスがアクセス可能なリソースを定義しておく方式であり，SELinux などに採用されている．

これらのような制御モデルは，スマートシティ IoT には適さないと考えられる．なぜなら，最初に述べたようスマートシティ IoT の特徴は，様々な情報の統合によるより良いサービスの提供にあり，情報が一方通行な MLS やあるプロセスの制御のみを扱う Type Enforcement は不十分なためである．

### 6.2　Remote attestation との関係

Remote attestation と提案した履歴に基づく情報流通制御は補完関係にある．Remote attestation が完璧なら，システムが正しく動作することが保証されるので，情報流通制御は不要である．しかし，IoT デバイスには，HW の制約から，remote attestation を導入できない場合がある．また，ソフトウェアのバグや設定のミスまでは防ぐことができない．このような場合には，情報流通制御が必要であると考えられる．

### 6.3　トレースバック

本論文では，履歴に基づいた情報流通制御方式について述べたが，提案方式は情報のトレースバックにも有用である．一般的に，セキュリティインシデントが発生した際には，その発生原因に加えて，どのような被害があるかを明らかにする必要がある．提案方式では，情報に流通履歴が記録されているため，その流通履歴をログに記録しておくことで，インシデント発生時に漏洩した情報の出所や漏洩範囲の特定が容易になると考えられる．

## 7. 関連研究

### 7.1 プロセスの実行履歴に基づいたアクセス制御

プロセスの動作をモニタリングすることにより，セキュリティポリシの作成を支援する仕組が提案されている[7]. 具体的には，ファイルのオープンや読み書きの履歴から，SELinux のアクセス制御ポリシを生成する．また，プロセスが正しく動作しているときのみ，リソースにアクセスできるように，プロセスの実行履歴に基づいたアクセス制御方式が提案されている[8]．しかし，これらの制御は，プログラムに対するアクセス制御であり，プログラムの実行履歴を利用している．一方，提案方式は，情報の流通履歴であり，アクセス制御に用いる情報が異なっている．

### 7.2 リソースの操作履歴に基づいたアクセス制御

XML ドキュメントの履歴に基づいたアクセス制御[9]が提案されている．具体的には，XML ドキュメント内に，create, transfer, delete, change などの履歴を記録する．加えて，データベースには，ドキュメント間のコピーの関係が記録される．そして，このドキュメントにアクセスする際は，ドキュメントに記録されている履歴の中に，ポリシで規定したパターンがあるかを検索し，アクセス可否の判断を行う．また，情報の出どころ(Provenance)に基づくアクセス制御方式[10]が提案されている．クラウドシステムを対象としおり，クラウドのファイルに書き込む際に，ファイルの親子関係を記録する．クラウドのファイルに対して操作(例えば「外国へ送信」のような操作)をする際に，その操作が許可されるかを，provenance によって判定する．

これらのアクセス制御方式は，本方式と同じように情報の履歴に基づいた制御を行っているが，本方式は情報の流通にフォーカスしている点が異なっている．具体的には，複数のコンポーネント間の流通履歴を記録する点が上述した既存方式との差異であり，グラフでポリシを記述することによって，より直感的に流通のポリシを記述できる点がコントリビューションである．

## 8. 結論

スマートシティ IoT を堅牢にするために，本論文では情報の生成から消滅までを一貫して管理する情報流通制御方式を提案した．本方式では，IoT プラットフォームのコンポーネント間で情報が送受信される際に，PEP がその送受信を監視し，情報の流通履歴を基に PDP が情報の送受信の可否を判断する．さらに，プロトタイプを実装し，実用的であることを示した．

## 参考文献


[1] 米国におけるスマートシティに関する取り組みの現状, https://www.ipa.go.jp/files/000048357.pdf

[2] IEC, Orchestrating infrastructure for sustainable Smart Cities, http://www.iec.ch/whitepaper/pdf/iecWP-smartcities-LR-en.pdf

[3] ICS CP/PE (Cyber-to-Physical or Process Effects) case study paper – Media report of the Baku-Tbilisi-Ceyhan (BTC) pipeline Cyber Attack, https://ics.sans.org/media/Media-report-of-the-BTC-pipeline-Cyber-Attack.pdf

[4] FIWARE, https://www.fiware.org/

[5] ICS CP/PE (Cyber-to-Physical or Process Effects) case study paper – Media report of the Baku-Tbilisi-Ceyhan (BTC) pipeline Cyber Attack, https://ics.sans.org/media/Media-report-of-the-BTC-pipeline-Cyber-Attack.pdf

[6] Victor Costan and Srinivas Devadas, Intel SGX Explained, Cryptology ePrint Archive, http://eprint.iacr.org/2016/086

[7] 原田 季栄 保理江 高志 田中 一男, プロセス実行履歴に基づくアクセスポリシー自動生成システム, Network Security Forum, 2003

[8] 高田 喜朗, 王 静, 関 浩之, 実行履歴に基づくアクセス制御の形式モデルと検証, 電子情報通信学会論文誌. D, 情報・システム, 2008

[9] Patrick Röder, Omid Tafreschi, and Claudia Eckert. 2007. History-based access control for XML documents. In Proceedings of the 2nd ACM symposium on Information, computer and communications security (ASIACCS '07)

[10] Adam Bates, Ben Mood, Masoud Valafar, and Kevin Butler. 2013. Towards secure provenance-based access control in cloud environments. In Proceedings of the third ACM conference on Data and application security and privacy (CODASPY '13)