Blockcerts Open Badges 証明書に関する仕組みと考察

福田 岐弦[1,a)]　水野 重弦[1,b)]　渡邉 健[1,c)]　佐古 和恵[1,d)]　寺田 雅之[2]　吉濱 佐知子

概要：情報処理学会は従来紙で発行していた資格証明書の一部を Open Badges 規格を用いて電子的に発行することを発表した．本稿では情報処理学会の発行する電子資格証明書のサービスの妥当性を検証することを目的として, それを構成する Open Badges 規格や, オープンソースソフトウェアである Blockcerts の公開情報から調査した技術的概要を解説する．また, 電子資格証明書の発行をアウトソーシングする事による問題についても考察する．

キーワード：Open Badges, ブロックチェーン, DX, アウトソーシング

Blockcerts Open Badges and some considerations on its application

Kigen Fukuda[1,a)]　Shigeo Mizuno[1,b)]　Ken Watanabe[1,c)]　Kazue Sako[1,d)]　Masayuki Terada[2]
Sachiko Yoshihama

1. はじめに

情報処理学会では, 学会運営のデジタル化を推進する一環として, 従来紙で発行していた資格証明書の一部を Open Badges 規格 [6] を用いて電子的に発行すると発表した [13]. ある個人に特定のスキルが備わっていることを保証することは, 今後の日本の流動的な労働市場において, その人の価値を高めるものである．そしてそれを公平的な観点から保証することは, 専門家によって構成されている学会でこそ可能であり, その資格証明書が信頼たりうるものになる．これからも, 学会が社会に価値を提供する存在意義の一つになりうる．

一方で, 物理的な紙で資格証明書を印刷して発行, 郵送するのは事務手続きとしてコストがかかるものであり, コロナ禍でその負担が顕著になった．一方, 資格証明書の単なる電子化であると, コピーや編集が容易になるため, 偽造されたものが出回る懸念がある．そこで海外でも実績のある Open Badges 規格にのっとり, 検証サイトを用いて検証可能な電子資格証明書発行を開始した．また, 電子資格証明書の発行にあたり, ブロックチェーン技術を用いた Blockcerts の仕組みの詳細についてと呼ばれるソフトウェアを用いて実装している．

本稿では, Open Badges の規格を拡張した Blockcerts (Blockcerts Open Badges) の技術的概要を公開情報 [8] [3] から調査した結果を解説するとともに, それをアウトソーシングすることのリスクについて議論する．

2. 情報処理学会の発行する電子資格証明書

情報処理学会は，同学会が認定する資格を取得した人の一部に資格証明書を電子的に発行している．この電子資格証明書はアウトソーシング先の会社のサービスを活用している．このサービスは Open Badges v2.0 の規格に基づいた電子資格証明書をアウトソーシング先の会社が代理発行するものである．また，オープンソースの Blockcerts v2.0 を用いて，発行した電子資格証明書の情報をブロックチェーンに登録することにより，証明書の改竄を困難にしている．

発行された電子資格証明書は SNS などを通じて公開・共

[1] 早稲田大学
Waseda University
[2] (株)NTT ドコモ
NTT DOCOMO, inc.
[a)] kigenfukuda@toki.waseda.jp
[b)] shigeowaseda1024@akane.waseda.jp
[c)] kenwaz113@ruri.waseda.jp
[d)] kazuesako@aoni.waseda.jp

有することができる．Facebook で公開されていた CITP のバッジでは，資格取得者の名前や資格の受賞年度および発行日，資格発行者の情報が確認できた．また，この資格証明書は 2 つの種類の JSON ファイルとしてダウンロードすることができた．

図 1 に示すようにこれらの 2 種類の JSON ファイルを，それぞれ検証サイトで検証することによって，下記に例示する検証サイトを用い, 共有された情報を入力して検証することができる．例えば，IMS Global が提供している Open Badges 2.0 Validator (`https://openbadgesvalidator.imsglobal.org/`)，badgr (`https://badgecheck.io/`) などがある．ブロックチェーンを用いた検証をする検証サイトの例としては，ネットラーニング社のオープンバッジ検証サイト (`https://www.netlearning.co.jp/openbadge/verify.html`)，Blockcerts Universal Verifier (`https://www.blockcerts.org/`) などがある．

有効期限切れや発行時のミスが発覚するなどの要因により，バッジが証明書として失効することがある．10 月 28 日現在，情報処理学会が発行した電子資格証明証で失効したものは 511 件あり，そのうち，メールアドレスの変更が 240 件で最多，期限切れが 145 件，登録ミスが 83 件，資格更新が 33 件である[*1]．他にも，不正が発覚したときなどにバッジを失効させるケースが将来的にはあると考えられる．

3. Hosted Badge と Blockcerts の組み合わせ

Open Badges v2.0 ではバッジデータへのデジタル署名の付与は必須ではなく, HostedBadge 型という, 発行者または発行代理者が保存・管理する電子資格証明書の情報 (以下, 原本データ) を原本保持サーバに保存し, 検証者が原本データの内容を検証サイトを使用し目視で確認する検証方法が定義されている [8].

従って, 情報処理学会の発行するバッジデータ自体にはデジタル署名は付与されていない．しかし, 情報処理学会が発行するバッジにおいては Blockcerts を利用し, 改竄耐性を向上させている．詳しくは 5 章で説明するが, 代理発行社が複数のバッジデータをまとめて, ブロックチェーン上にそのハッシュ値を署名付きで書き込んでいる．あくまで, Open Badges v2.0 が定めているのは Hosted Badge という仕様であり, Hosted Badges には電子資格証明書の検証にあたって原本データを全面的に信頼しなくてはならないという問題がある．その問題は Blockcerts というソフトウェアを利用することで回避できる．

そのため, 図 1 に示すように, 同じバッジデータを示す 2 種類の JSON ファイルがある．原本データの URI とバッジデータを内包する Hosted Badge としての JSON ファイルと, その JSON ファイルの内容にブロックチェーン上の参照情報を追記して作成する Blockcerts としての JSON ファイルである．

検証者はそれぞれの検証サイトにおいて, 対応する JSON ファイルを入力として検証を行うことができる．

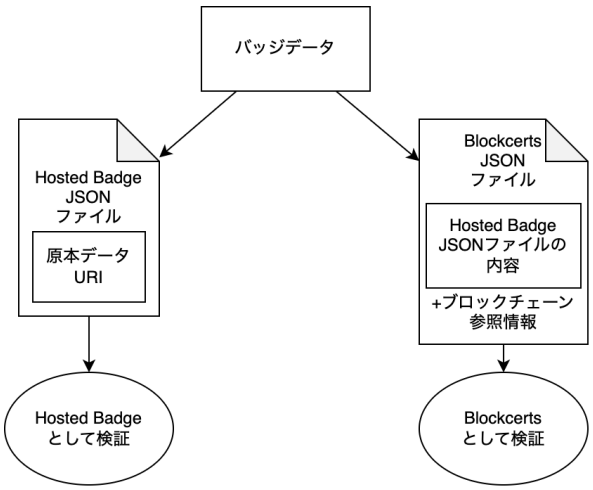

図 1 HostedBadge と Blockcerts のバッジデータと検証方法

また, 2 章で述べた失効についても, 同じバッジを示す 2 種類のバッジデータがあるため, Hosted Badge としての失効方法と Blockcerts としての失効方法がある．詳細は 4.2 節と 5.2.3 節で述べる．

4. Hosted Badge としての検証の詳細

1EdTech Consortium Inc. [5] は, 教育者, 管理者, 学習者の日々の活動をより簡単することを目的に, 試験, シラバス管理など教育情報システムの 25 以上の技術標準を策定している国際標準化団体である．Open Badges とは MIT の Digital Credentials Consortium と 1EdTech が共同で定める技術標準規格に沿って発行される, スキルや経験を表現するデータフォーマットである．授与されたバッジの内容は検証可能であり, バッジはオープンバッジウォレットで被資格取得者によって一元管理され, ソーシャルメディア等で簡単にバッジを提示することができる．2022 年の Badge Count レポート [4] によると発行されたオープンバッジデータの数は 7 千万を超えている．

1EdTech の提供するソフトウェアスイート [2] を用いた運用の自己テストを行い, 承認を受けたサービスは 2023 年 10 月時点で 27 件あり, そのうち 26 件が 2018 年にリリースされた Open Badges v2.0 に準拠したものである [6]．本章では, Open Badges v2.0 における Hosted Badge の検証方法について詳しく説明する．

4.1 検証方法

Hosted Badge としての検証に必要なのは, Hosted Badge

[*1] この情報は Facebook で公開されていた CITP のバッジの JSON ファイルに記載されている失効リストから確認した．

JSON ファイルである．Hosted Badge JSON ファイルには原本データの URI が記述されている [8].

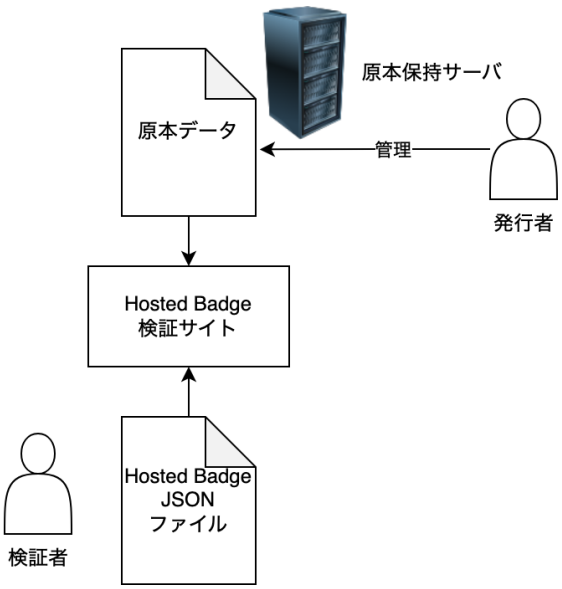

図 2 HostedBadge の検証方法

原本保持サーバ内の原本データには以下の情報が, テキスト形式またはそれを記載したドキュメントへのリンクとして記載される.

- バッジ情報: バッジ ID, 発行日時, バッジの名称, 発行基準など
- 発行者情報: 組織名称, 組織説明, メールアドレスなど
- 資格取得者情報: ハッシュ化されたメールアドレス[*2] など

原本データの記述例を図 3 に示す. これは Facebook で公開されていた CITP のバッジの JSON ファイルを元に作成した仮想の例である.

検証者は検証サイトを使用して, 受け取った JSON ファイルから登録されている原本データの内容を閲覧し, 登録されているバッジの内容を確認する．Hosted Badge としての検証は原本データの内容が「Source of Truth」とみなすことであると、文献 [8] に記載されている.

4.2 失効の手法

発行者はバッジを失効する際には以下のいずれかの方法で失効を行う.

- 原本データの URI に対して HTTP ステータスコード 410 Gone を返すように原本保持サーバの設定を変更する.
- 原本データの内容を書き換え, 失効されたことを示す

[*2] 図 3 の 12 行目の記述がハッシュ化されたメールアドレスの例である.

```
1  {
2      "@context": "https://w3id.org/openbadges/v2",
3      "type": "Assertion",
4      "id": "https://代理発行社.jp/api/v1.0/openbadge/v2/As
           sertion/...",
5      "badge": "https://代理発行社.jp/api/v1.0/openbadge/
           v2/BadgeClass/...",
6      "image": "https://代理発行社.jp/api/v1.0/openbadge/
           v2/Assertion/.../image",
7      "verification": {
8          "type": "HostedBadge"
9      },
10     "issuedOn": "202y-mm-ddT00:00:00+09:00",
11     "recipient": {
12         "identity": "sha256$...",
13         "type": "email",
14         "hashed": true,
15         "salt": "..."
16     },
17     "evidence": [
18         {
19             "type": "Evidence",
20             "name": "認定番号",
21             "description": "..."
22         },
23         {
24             "type": "Evidence",
25             "name": "有効期限",
26             "description": "202y/mm/dd"
27         },
28         {
29             "type": "Evidence",
30             "name": "初回認定日",
31             "description": "201y/mm/dd"
32         }
33     ]
34 }
```

図 3 原本データの記述例

revoked プロパティを false から true に変更する.

検証者が原本データにアクセスした際に 410 Gone ステータスか, 失効状態を示す原本データを得た場合, バッジは失効されていると確認される.

4.3 メールアドレスの利用

Open Badges v2.0 では資格取得者の識別のための情報として主にメールアドレスが利用される [8]. 発行者は資格取得者のメールアドレスを受け取り, メールアドレスにソルトをつけて SHA-256 でハッシュ化し, そのハッシュ値とソルトをバッジデータ及び原本データに記入する. 授与されたバッジを保管するウォレットサービスが, インポートするバッジが同じメールアドレスの資格取得者かどうかの確認を行うほか, バッジを提示された検証者が, 氏名とメールアドレスで資格取得者を特定することができる.

4.4 HostedBadge の検証の限界

上記のように, Hosted Badges には電子資格証明書の検証にあたって原本データを「Source of Truth」として全面的に信頼しなくてはならないという問題がある．したがっ

て, 検証する際には, バッジの原本データが配置されている URL が信頼できるドメインにあることを検証者の目で確認することが望ましい. これを怠ると, 検証者は偽の URL を誤って受け入れるリスクがある. そのため, 情報処理学会のバッジを検証する場合には, 原本データの URL が代理発行社のドメインであることを別途確認する必要がある.

原本保持サーバがオフラインの際は検証者による原本データの閲覧は不可能になる. そのため発行者が原本保持サーバの運営を停止すると, 発行された全てのバッジが検証不可能になり電子資格証明書として無効となってしまう. また発行者を全面的に信頼できない場合, 資格保持者や検証者に通知されることなく原本データの書き換えや削除が行われる可能性がある.

5. Blockcerts としての検証の詳細

Blockcerts は MIT メディアラボと Learning Machine 社 (現 Hyland Credentials 社) が開発を開始した, ブロックチェーンを使用して証明書を作成, 発行, 表示, 検証するためのオープンソースソフトウェアである. 現在も Hyland Credentials 社が積極的に開発・提供に携わっている [9].

Blockcerts としてのバッジの記述例を図 4 に示す. これは Facebook で公開されていた CITP のバッジの JSON ファイルを元に作成した仮想の例である.

Blockcerts の仕組みでは発行者 (Issuer), 受取人 (Recipient), 検証者 (Verifier) の三者を想定する. 図 4 のように情報処理学会が発行したバッジデータには,「Issuer」の情報として情報処理学会を特定する情報が記載されている[*3]. しかし, 検証に使うデータは代理発行社のドメインに置かれている. 以下、本章において, Blockcerts の発行者とは代理発行社を指す.

5.1 Blockcerts としてのバッジデータ発行

発行者はバッジデータの発行時に, 一度に発行するバッジデータをまとめ, それらのバッジデータからマークルツリーを構築し, マークルルートを算出する. それぞれのバッジデータにはバッジデータ全体のハッシュ値, マークルルートの値, マークルルートへのパスがバッジデータに書き込まれる[*4]. Blockcerts では, Merkle Proof Signature Suite 2019 [12] に基づきマークルツリーを算出する. 図 5 にマークルツリーの構築の図を示す.

以上で算出したマークルルートをトランザクションに記述し, ビットコインやイーサリアムなどのブロックチェーンに登録する[*5]. トランザクションの作成時に, 発行者の秘密鍵を用いた署名が行われる. トランザクションを検証

```
 1 {
 2   "@context": [
 3     "https://w3id.org/openbadges/v2",
 4     "https://w3id.org/blockcerts/v2",
 5     {
 6       "displayHtml": ...
 7     }
 8   ],
 9   "id": "https://代理発行社.jp/.../openbadge/v2/...",
10   "type": "Assertion",
11   "recipient": {
12     "identity": "sha256$...",
13     "type": "email",
14     "hashed": true,
15     "salt": "..."
16   },
17   "displayHtml": "<section> ... </section>",
18   "badge": {
19     "@context": "https://w3id.org/openbadges/v2",
20     "id": "urn:uuid:...",
21     "type": "BadgeClass",
22     "name": "認定情報技術者(CITP) ",
23     "description": "...",
24     "image": "https://代理発行社.jp/.../openbadge/v2/Badg
           eClass/.../image",
25     ...
26     "issuer": {
27       "@context": "https://w3id.org/openbadges/v2",
28       "id": "https://代理発行社.jp/.../openbadge/v2/Block
           Cert/Issuer/...",
29       "type": "Issuer",
30       "name": "一般社団法人情報処理学会",
31       "description": "...",
32       "url": "https://www.ipsj.or.jp",
33       "email": "openbadge@ipsj.or.jp",
34       "image": "https://代理発行社.jp/.../openbadge/v2/Is
           suer/.../image",
35       "revocationList": "https://代理発行社.jp/.../openba
           dge/v2/Issuer/RevocationList/..."
36     }
37   },
38   "verification": {
39     "publicKey": "...",
40     "type": ["HostedBadge","MerkleProofVerificatio
         n2017","Extension"]
41   },
42   "issuedOn": "202y-mm-ddT00:00:00+09:00",
43   "image": "https://代理発行社.jp/.../image",
44   ...
45   "signature": {
46     "type": ["MerkleProof2017","Extension"],
47     "merkleRoot": "e96...",
48     "targetHash": "08f...",
49     "proof": [
50       {"left": "8ec..."},
51       ...
52       {"left": "0ff..."}
53     ],
54     "anchors": [
55       {
56         "sourceId": "0xf...",
57         "type": "ETHData",
58         "chain": "ethereumMainnet"
59       }
60     ]
61   }
62 }
```

図 4 Blockcerts としてのバッジの JSON 記述例

[*3] 図 4 の 26 行目から始まる部分がその例である.
[*4] 図 4 の 45 行目から始まる部分がその例である.
[*5] 図 4 の 58 行目はイーサリアムに登録される例である.

する公開鍵はバッジデータの verification [*6]という項目に，トランザクションの識別子は signature の anchor [*7]という項目に記述される．なお，ここで signature とは，デジタル署名ではなく，バッジデータのハッシュ値や，マークルツリーに関連するハッシュ値のことを指している．

実際に Ethereum 上に書き込まれたトランザクションを Ethplorer [1] から参照し，その内容を図 6 に示す．図に示された通り，代理発行団体が管理するアドレス (=公開鍵) (0x8349...) から，誰にも使われないバーンアドレスと呼ばれるアドレス (本例では 0xdeaDDeAD...) に対して 0 ETH が送金され，その input data にマークルルートの値 (e96b...) が書かれている．

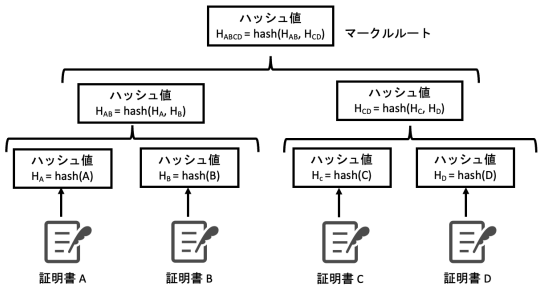

図 5 複数のバッジデータからマークルルートの算出

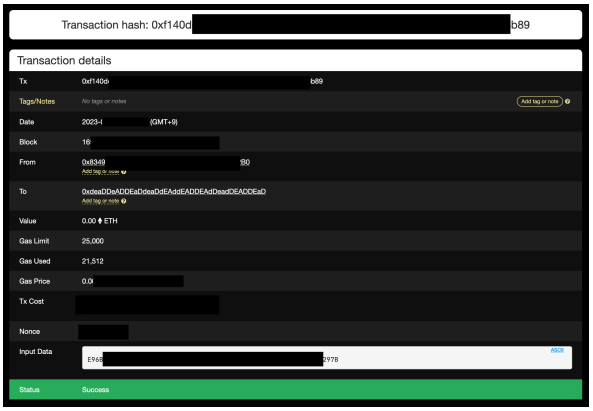

図 6 Ethereum 上のトランザクションの例

5.2 Blockcerts としての検証

Blockcerts としてのバッジデータは図 7 のように検証が行われる．

Blockcerts のバッジデータには上記で述べたように，主に以下の情報が記述される．

- マークルルートを登録したブロックチェーンのトランザクション ID

*6 図 4 の 38 行目から始まる部分がその例である．
*7 図 4 の 54 行目から始まる部分がその例である．

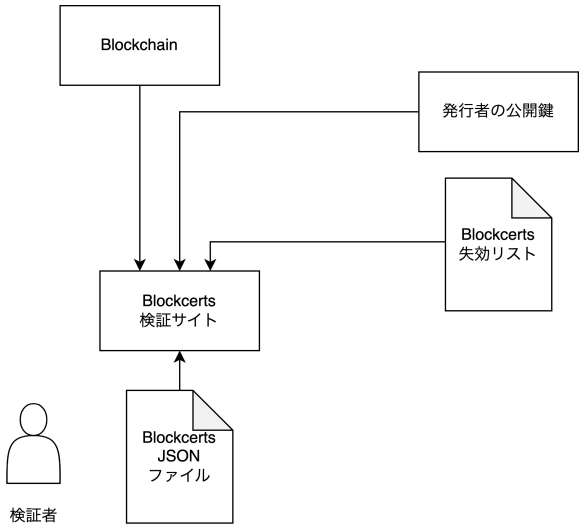

図 7 Blockcerts の検証方法

- 発行者の識別情報 (以下は一部)
  - Issuer の名前
  - ブロックチェーンのトランザクションを検証する公開鍵
- 証明する資格の内容
- 失効リストの URL

検証の際に以下の 4 点を確認する．

- Issuer の公開鍵によるトランザクションの検証
- ハッシュ値を用いてバッジデータが改竄されていないかの検証
- 発行者によって失効されていないかの確認
- 有効期限の確認

5.2.1 Issuer の公開鍵によるトランザクションの検証

Blockcerts としてのバッジデータが確かに Issuer により発行されているものであることを確認する．

まず，バッジデータには発行者の公開鍵もしくはそれを特定する URL が記述されており，その URL の先にある公開鍵とバッジデータの記述された公開鍵が一致しているかを確認する．その公開鍵を用いてトランザクションが発行者によって作成されたものであることをトランザクションの署名から検証する．

5.2.2 ハッシュ値を用いてバッジデータが改竄されていないかの検証

バッジデータに記述された内容がブロックチェーンに登録された時点から改竄されていないかを確認する．Blockcerts では Merkle Proof Signature Suite 2019 [12] に基づきバッジデータが改竄されていないか検証を行う．

バッジデータには以下の 4 つがバッジデータに記述されている．

- Merkle Root が記載されているブロックチェーンのト

ランザクションの ID
- Merkle Root の値
- バッジデータのハッシュ値
- マークルルートからバッジデータまでの Merkle Tree のパス

以上を用いて，検証者は以下を確認する．

- Merkle Root が確かにブロックチェーンのトランザクションに記載されていること
- 受け取ったバッジデータをハッシュ化し，バッジデータに記述されているハッシュ値と一致すること
- 受け取ったバッジデータのハッシュ値が確かにその Merkle Tree の中に含まれていること

5.2.3 発行者によって失効されていないかの確認

バッジデータの revocationList の属性に Issuer が失効させたバッジデータの ID のリスト (以下，失効リスト) が公開されているリンクが記述される．検証者は失効リストを取得し，受け取ったバッジデータの ID が含まれるかを確認し，含まれた場合は失効されていると判断する．図 4 の 35 行目に失効リストの URL が記載されているが，その先には図 8 のようなデータが記述されている．

```
 1 {
 2   "@context":"https://w3id.org/openbadges/v2",
 3   "type":"RevocationList",
 4   "id":"https://代理発行社.jp/...",
 5   "revokedAssertions":[
 6     {
 7       "id":"https://代理発行社.jp/.../openbadge/v2/...",
 8       "revocationReason":"テストのため"
 9     }
10   ]
11 }
```

図 8 失効リストの例

5.2.4 有効期限の確認

バッジデータに有効期限を示す expires の属性が記述されていた場合，現在時刻と記述された時刻を比較し，有効かどうかを判定する．

検証において，誰もが独立に自分たちで検証できるためにオープンソースのライブラリ [10] [11] の OSS が提供されている．

6. アウトソーシングのリスク

以上, 詳細に見てきたとおり, 電子化をすることにより, ユーザは確かに情報処理学会が発行した資格を持っていることを電子的に証明することができる．また, 紙の資格証明書発行と比較して, 紙の印刷, 郵送という学会事務を軽減することができる．現在, 情報処理学会はこの処理をアウトソーシングし, 代理発行を依頼している．

web サーバの運営や会計処理など, 事務手続きをアウトソーシングすることは現在でも多く行われている．しかし, 資格証明書は紙で発行することで永きにわたり授与者にその価値を保証するものである．それを紙でなく電子で提供するサービスをアウトソーシングすることは, 継続性がもとめられない通常の手続きのアウトソーシングとは異なるリスクが発生する．本章では特に, 継続性の観点からアウトソーシングのリスクを検討する．

6.1 Hosted Badge としての検証の継続性

初めに, Hosted Badge としての検証可能性の継続について議論する．4 章で述べた Hosted Badge としての性格上, 検証の際に原本データ保持サーバへのアクセスが発生する．なんらかの理由で原本データ保持サーバにアクセスできなくなった場合, 検証者は Hosted Badge としての検証がいかなる検証サイトでも不可能になる．また, 電子資格証明書の失効時に適切に原本データの書き換えが行われない可能性や, 過失により原本データを消失してしまう可能性がある．

これを回避するためには, 情報処理学会による原本データ保持サーバの自主運営が有効である．しかし, 情報処理学会には検証プログラムの管理コストや, 多数の原本データのドキュメントを管理するコストが発生する．発行するバッジの数が増えると, 原本データの管理コストも増大し, 特にバッジを失効する際は各原本データの内容を逐次書き換える必要がある．

なお, Hosted Badge としての検証が不可能になった際には, Blockcerts としての JSON ファイルの検証により電子資格証明書の検証が継続できる．次に, Blockcerts としての検証の継続性について議論する．

6.2 Blockcerts としての検証の継続性

次に, Blockcerts としての検証可能性の継続について議論する．情報処理学会の発行するバッジは Blockcerts としても検証可能であり, 5 章で説明したように, 検証には発行者の公開鍵と電子資格証明書の失効情報が必要である．しかし, Blockcerts v2.0 の仕様上, 発行者の公開鍵と失効リストは URL で指定されており, 現在はアウトソーシング先のドメインである．なんらかの理由で公開鍵の URL が参照できなくなった場合, Blockcerts の検証において公開鍵を取得できず, 電子資格証明書を検証することができない．また, 失効リストの URL が参照できなくなった場合も同様である．

これを回避するためには, 情報処理学会で公開鍵の URL と失効リストの URL を準備し, 公開鍵と失効情報の提供をすることが有効である．これらを提供するのは通常の web サイトを準備する手間と大きくは変わらない．

Blockcerts はオープンソースソフトウェアであるため,

全ての Blockcerts 検証サイトが停止しても, 個人が検証プログラムを入手して検証を継続できる. 情報処理学会が独自にデプロイし検証サイトを運営することで検証の継続性を高めることができる.

6.3 検証結果の正しさへの依存

4 章で述べた Hosted Badge の性格上, Hosted Badge として検証する際には, 参照した原本データを全面的に信頼して, 検証結果を信じるしかない. そのため, 原本保持サーバの内容が不正に改竄され, 偽の検証結果が提示されても, それに気が付くことは難しい[*8].

Blockcerts としての検証においても, 失効リストが不正に作成されることで, 例えば本来有効である電子資格証明書が失効されたと検証されてしまう. また, 公開鍵 URL で指定された公開鍵情報が不正に書き換えられると, 発行済みの電子資格証明書の Blockcerts としての検証ができなくなってしまう. この依存は通常のアウトソーシングでも同様であるが, 継続性が求められるためアウトソーシング先は永きにわたりサーバとドメインを厳重に管理する必要がある. たとえば, アウトソーシング先のドメイン名がなんらかの理由で転売されることがあれば, 乗っ取られた場合とおなじリスクが発生する.

なお, 原本データや公開鍵提供ページ, 失効リストにはデジタル署名が付与されていない[*9]. 今後, 発行者が検証に必要なデータにデジタル署名を付与することで, 電子資格証明書の検証結果の信憑性を高める仕様となることが望ましい.

6.4 情報処理学会でないドメイン名によるなりすましリスク

バッジの発行者の情報としては情報処理学会の詳細が記述されるものの，原本保持サーバや発行者の公開鍵にはアウトソーシング先のドメイン名が使われる. しかし, アウトソーシング先のドメイン名は必ずしも会員や検証者に周知されていないため, 不正なドメイン名にすり替えられてもそれに気付くのは難しい. したがって，ここに不正な書き換えをする者が所有する不正ドメインの情報を記述し，ブロックチェーンに登録すれば，Hosted Badge としての検証も，Blockcerts としての検証も通ってしまう情報処理学会を装ったバッジが作成できてしまう.

これを防ぐには, 情報処理学会自身のドメインにおいて, 検証に必要なデータを提供することが考えられる.

6.5 バージョンアップの困難さ

2023 年 10 月時点で最新の Open Badges の規格である Open Badges v3.0 [7] は, W3C の定める標準規格 Verifiable Credentilals に準拠し, バッジデータに発行者のデジタル署名が付与された電子資格証明書のデータフォーマットである[*10]. 電子資格証明書の検証をバッジデータに付与されたデジタル署名で行うことができるため, 原本保持サーバが不要になる. そのため, Open Badges v3.0 へのアップデートによって上述したリスクを回避したり, 管理コストを下げることができる可能性がある. しかし, 情報処理学会がそのようなバージョンアップをしたいと思っても, アウトソーシング先との調整が発生する. また, 情報処理学会が新規発行から新機能で運営する別のアウトソーシング先と提携しても, 過去発行分に関しての公開鍵 URL, 原本保持, 失効情報の管理などは, 引き続き現在のアウトソーシング先に依頼する必要がある.

6.6 契約によるリスク回避

上記リスクを契約で回避する方法も考えられる. Hosted Badge としての検証の継続性を高めるためには, 原本データの正しい管理と原本保持サーバの継続的な運用の義務化が有効である. 同様に, Blockcerts としての検証の継続性を高めるためには, 公開鍵と失効リストの継続的な管理・公開の義務化が有効である.

しかし, アウトソーシング先がこれらの契約事項に合意し, 情報処理学会と長期間有効な契約を締結する保証はない. 現実的には, 原本データや公開鍵などの検証に必要な情報は情報処理学会のドメイン下で管理し, そのドメインの運用を行うサーバの管理をアウトソーシング先に委託する短期間の契約が考えられる. このような契約であれば, 契約更新によりアウトソーシング先が変更となっても電子資格証明書のサービスを継続しやすくなると思われる.

7. おわりに

本稿では情報処理学会が発行する電子資格証明書について公開情報から調査した結果を紹介した. 2 章で検証，失効の概要を説明し，3 章から 5 章にかけて技術的な詳細を述べた. また，6 章でアウトソーシングすることにより資格情報の検証の継続性が失われるリスクや，悪意により電子資格証明書の検証結果が捏造や改竄されるリスクがあることを述べた. また，情報処理学会による電子資格証明書発行サービスの自主運営の他、さまざまなリスクの回避方法について述べた.

専門家により構成される情報処理学会だからこそ，電子資格証明書の発行を担い社会に価値提供することことがで

[*8] なお, 本システムは Blockcerts を活用しているため, ブロックチェーンの情報を参照すれば, 原本データの捏造や偽造は検出することができる.
[*9] これは本サービスの利用する Open Badgesv2.0 や Blockcerts の仕様による.
[*10] なお, Open Badges v3.0 のステータスは 2023 年 10 月時点で ”Candidate Final” であり, 最終版として規格が確定したものではない.

きる. 今後, 学会が信頼たりうる資格証明書を発行することにこの議論が寄与することを期待したい.

参考文献

[1] Ethplorer. https://ethplorer.io/tx/0xf140dca75cddeffb22faa7d141c9dcbf685ff97fababdfa43614fc2a14cb8b89.
[2] 1EdTech. 1EDTech Open Badges 2.0 Certification Suite. https://openbadgesvalidator.imsglobal.org/openbadges20/index.html.
[3] 1EdTech. 1edtech/cert-schema. https://github.com/1EdTech/cert-schema/blob/master/docs/open_badge_v2_extensions.md.
[4] 1EdTech. Badge Count 2022. https://content.1edtech.org/badge-count-2022/.
[5] 1EdTech. Home. https://www.1edtech.org/.
[6] 1EdTech. Home — IMS Open Badges. https://openbadges.org/.
[7] 1EdTech. Open Badges Specification Candidate Final Public Spec Version 3.0. https://www.imsglobal.org/spec/ob/v3p0.
[8] 1EdTech. Open Badges v2.0 IMS Final Release. https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/index.html.
[9] Blockcerts. About blockcerts. https://www.blockcerts.org/about.html.
[10] blockchain certificates. cert-verifier. https://github.com/blockchain-certificates/cert-verifier.
[11] blockchain certificates. cert-verifier-js. https://github.com/blockchain-certificates/cert-verifier-js.
[12] Kim Hamilton Duffy (Learning Machine) and Dmitry Semenovskiy (Vaultie Inc.). Merkle proof signature suite 2019. https://w3c-ccg.github.io/lds-merkle-proof-2019/, 2021.
[13] 一般社団法人情報処理学会. オープンバッジ発行について – CITP 認定情報技術者. https://www.ipsj.or.jp/CITP/openbadge.html.