


SMART Health Cards & Privacy-Friendly QR codes for Identity

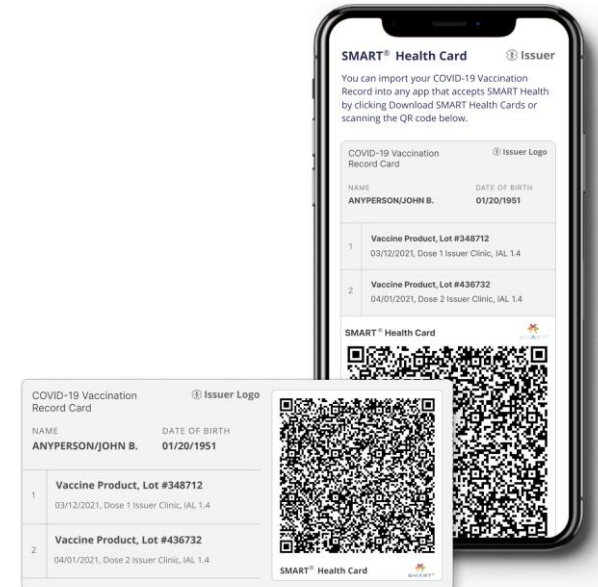
Christian Paquin

 @chpaquin



Microsoft Research

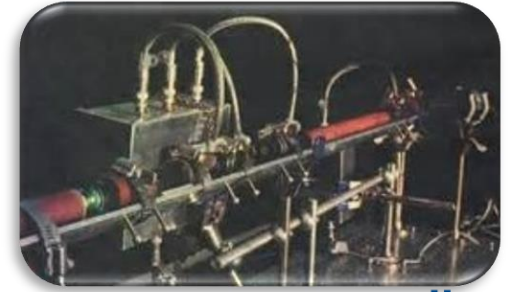
nsec 2022



About me



- Studied quantum cryptography 25+ years ago at University of Montreal
- Worked in the industry as a cryptographic engineer
- Now with the *MSR Security & Crypto* team, working on
 - Privacy-preserving identity (anonymous credentials)
 - Searchable encryption
 - Post-Quantum Cryptography
- Helped with the development of the SMART Health Card Framework



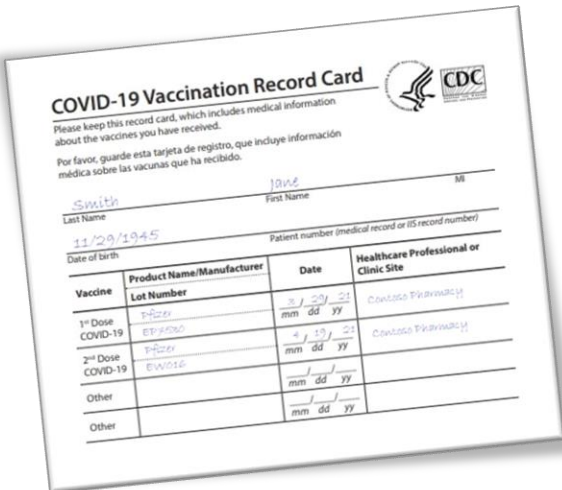
Université 
de Montréal

<https://www.microsoft.com/en-us/research/group/security-and-cryptography/>

SMART Health Cards (SHC)

- SMART Health Cards are the tamper-resistant, electronic equivalent of CDC Covid-19 cards
 - Available as a file or a QR code

SMART = Substitutable
Medical Applications,
Reusable Technologies



COVID-19 Vaccination Record Card

Please keep this record card, which includes medical information about the vaccines you have received.

Por favor, guarde esta tarjeta de registro, que incluye información médica sobre las vacunas que ha recibido.


Last Name: Smith First Name: Jane SS: 88
Date of birth: 11/09/1945 Patient number (medical record or IIS record number):

Vaccine	Product Name/Manufacturer	Lot Number	Date	Healthcare Professional or Clinic Site
1 st Dose COVID-19	Pfizer	EP-0000	3 / 20 / 21	COVID-19 Vaccination
2 nd Dose COVID-19	Pfizer	EP-0000	4 / 19 / 21	COVID-19 Vaccination
Other	EWING		mm dd yy	
Other			mm dd yy	



QR code is *not* a URL, it contains all the card info!



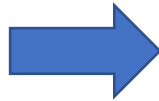
COVID-19 Vaccination Record Card 

Please keep this record card, which includes medical information about the vaccines you have received.
Por favor, guarde esta tarjeta de registro, que incluye información médica sobre las vacunas que ha recibido.

Last Name: Smith First Name: Jane Sex: M

Date of birth: 11/29/1945 Patient number (medical record or ID record number): _____

Vaccine	Product Name/Manufacturer	Date	Healthcare Professional or Clinic Site
1 st Dose COVID-19	<u>Pfizer</u>	<u>3 / 29 / 21</u> mm dd yy	<u>Contoso Pharmacy</u>
2 nd Dose COVID-19	<u>Pfizer</u>	<u>4 / 19 / 21</u> mm dd yy	<u>Contoso Pharmacy</u>
Other		<u>mm dd yy</u>	
Other		<u>mm dd yy</u>	



FHIR encoding:

```
{
  "resourceType": "Bundle",
  "type": "collection",
  "entry": [
    { "fullUrl": "resource:0", "resource": {
      "resourceType": "Patient",
      "name": [{ "family": "Smith", "given": ["Jane"] }],
      "birthDate": "1945-11-29"
    }
  },
  {
    "fullUrl": "resource:1", "resource": {
      "resourceType": "Immunization",
      "status": "completed",
      "vaccineCode": { "coding": [{ "system": "http://hl7.org/fhir/sid/cvx", "code": "208" } ] },
      "patient": { "reference": "resource:0" },
      "occurrenceDateTime": "2021-03-29",
      "lotNumber": "EP7530",
      "performer": [{ "actor": { "display": "Contoso Pharmacy" } } ]
    }
  },
  {
    "fullUrl": "resource:2", "resource": {
      "resourceType": "Immunization",
      "status": "completed",
      "vaccineCode": { "coding": [{ "system": "http://hl7.org/fhir/sid/cvx", "code": "208" } ] },
      "patient": { "reference": "resource:0" },
      "occurrenceDateTime": "2021-04-19",
      "lotNumber": "EW016",
      "performer": [{ "actor": { "display": "Contoso Pharmacy" } } ]
    }
  }
]
}
```

```
{
  "iss": "https://smarthealth.cards/examples/issuer",
  "nbr": 1619068267070,
  "vc": {
    "type": [
      "https://smarthealth.cards#health-card",
      "https://smarthealth.cards#immunization",
      "https://smarthealth.cards#covid19"
    ],
    "credentialSubject": {
      "fhirVersion": "4.0.1",
      "fhirBundle": <FHIR BUNDLE>
    }
  }
}
```



1. Encode the immunization info as a FHIR bundle

FHIR = Fast Health
Interoperability
Resources

2. Encode the FHIR bundle into a JSON Web Signature (JWS) payload

Payload follows a
Verifiable Credential
format

4. Encode the JWS header with compression alg ID (DEF), signature alg ID (ES256), and issuer key ID

```
"zip": "DEF",  
"alg": "ES256",  
"kid": "d630duSMwVfmOtrMKZX6izJfcampjK1h0D4jrXxJwU"
```

3. Minimize, compress, and base64url the JWS payload

```
{  
  "iss": "https://smarthealth.cards/examples/issuer",  
  "nbf": 1619068267070,  
  "vc": {  
    "type": [  
      "https://smarthealth.cards#health-card",  
      "https://smarthealth.cards#immunization",  
      "https://smarthealth.cards#covid19"  
    ],  
    "credentialSubject": {  
      "fhirVersion": "4.0.1",  
      "fhirBundle": <FHIR BUNDLE>  
    }  
  }  
}
```

```
eyJ6aXAiOiJERUYiLCJhbGciOiJFUzI1NiIsImtpZCI6ImQ2MzBkdVNNV21WZm1PdHJNSlYmNm16SmZjYWlwaksxadbENGpyWHhKd1UifQ.  
3VJNj9MwEP0raLgmTZx2-5ETonCAA6q0CxxQD44zbYz8EdlO2rLKf2ectguI3R44Ivky9pvn997MI0jvoYQmhNaXWeY1d6FBrkIzEdzVPsMj161CnxGwQwcJmMpByeZslc-  
XxXyRL_IIEgHlI7wRlgQ8Bii_PTEeDofJYTqxbp8VOVtmwmGNJkiufNYz2CYQTi3Gji_o5E7ySuH6CUPfvSjt9b1IY3ETJ7XuJpZBg7TmJlDYXtZsFUX9knnfVd9RhOhv10hHKn3kKWE2ySeM-  
OLt287UCiPGobedE_gwuolLw9U1CKsUsZ2V0AfuRNAJuVPqs1MEuPaXOQGuxTPEG7JD_XEeXOOZhGupiA_utQwNvexljyZG-  
5Ebh01AvipJnt_xECnYanaXMPYWKxiG5Fkr7LaID38m6MPnR9dpx0JGOofScyGkbWtRwZha2n2o15_8gH1ZftoII1ajHsSA828rDPRH41AjJlQ5EsYtkMC7cX5KGeHdk3U9ntwBLJCdG58imYfpD5  
TFCzNp9FwAsqGT52uaKVLel9Z3E1j4C26nXU6XpJALoJl8Z9a-lbxGO2aNtx6-2rTcKe5OFFy25fCK_7D8GYp-yu8rzmb_3t2dIbhJw  
.1pd_USQ1fJwOhwZnsg0vL_c3zu8ScI12YLxZk-Su7MRK1EIEReOkov831duM1FkmORMiucbMozw0IVZK0OmR1A
```

Compact JWS = base64url of
<header>.<payload>.<sig>

5. Sign the JWS header and payload with the issuer private key, encode the signature in the JWS

The issuer is identified by the
"iss" URL in the payload, from
where the validation public key
can be retrieved

eyJ6aXAiOiJERUYiLCJhbGciOiJFUzI1NiIsImtpZCI6ImQ2MzBkdVNNV2lWZm1PdHJNSlpyNm16SmZjYWlwaksxadbENGpyWHhKd1UifQ.
3VJNj9MwEP0raLgmTZx2-5ETonCAA6q0CxxQD44zbYz8EdlO2rLkf2ectguI3R44Ivky9pvn997MI0jvoYQmNaXWeY1d6FBrkIzEdzVPsMj161CnxGwQwcJmMpByeZslc-
XxXyRL_IEegHlI7wRlgQ8Bii_PTEeDofJYTqx8p8VOVtmwmGNJkiufNYz2CYQTi3Gji_o5E7ySuH6CUPfvSjt9b1IY3ETJ7XujPzBg7TmJlDYXtZsFUX9knnfVd9RhOhv10hHKn3kKWE2ySeM-
OLt287UCiPGobedE_gwu0LLw9U1CKsUsZ2V0AfuRNAJuVPqslMEuPaXOQGuxTPEG7JD_XEeXOOZhGupiA_utQwNvexljyZG-
5Ebh01AvipJnt_xECnYanaXMPYWKxiG5FkR7LaID38m6wMPnR9dpx0JGOfScyGkwbWtRwZha2n2o15_8gH1ZftoII1ajHsSA828rDPRH4lAjJlQ5EsYtkMC7cX5KGeHDK3U9ntwBLJCdG58imYfpD5
TFCzNp9FwAsqGT52uaKVLLeL9Z3Elj4C26nXU6XpJALoJl8Z9a-lbxGO2aNtx6-2rTcKe5OFFy25fCK_7D8GYp-yu8rzmb_3t2dIbhJw
.lpd_USQlfJwOhwZnsg0vL_c3zu8ScI12YLxZk-Su7MRK1EIEREOkov831duM1FkmORMiucbMozw0IVZK0OmR1A

6. Encode the JWS as a
numeric QR



shc:/5676290952432060346029243740446031222959532654603460292540772804336028702864716745222809286
436053277216255413333410504424564043555272933380467443364630938644561444204745262707552232124332
667764227593055044060573601064129336112327424350369523158643945750500082439666522202009680322757
536230707775344771124556334056931305705565471587228063707072873627612677365121210322803617366443
664593352434256440455092521696228772455774135703261040904226575267436745429643267217656457063540
043754376373150282456582763281074370458361121606050353924562366572944396875536711413441716474642
633296260725733447705224436396006266160506608241076387227092240355773386171125363284406243929104
372613577215810396429632344437145702540431262656557415512375934597304035927306506623042240576385
632003431710511104022603526665356552450587472663131741240632230704070450541032057723733522972413
568700432247235524334362672753935242610292350432456433434455926726760205072713674337356756361764
526000824535934042073606729657150752422654452655243326744423075602608256237103152282306116409743
235653712557567032926345738547626627453427137744559520565056604085011582704455771662828045261277
038201105116923353727076320612904360824704471623222105443083026562723620640126571742131292255260
811606444576723083925227733671225742070682639080572523041315631124506240461072205096543400943672
920316629041145125200635375263405523371750900056939543056083425257605085722305010231126446700767
211697764535006710555285359297401046755504038366357297434597445657058037331505406777211385428630
544317545620038721032373004242856372434626673110604557232042562643437326072545332667774032841453
0033464370420



7. Create the QR
code image



SMART Health Cards Framework

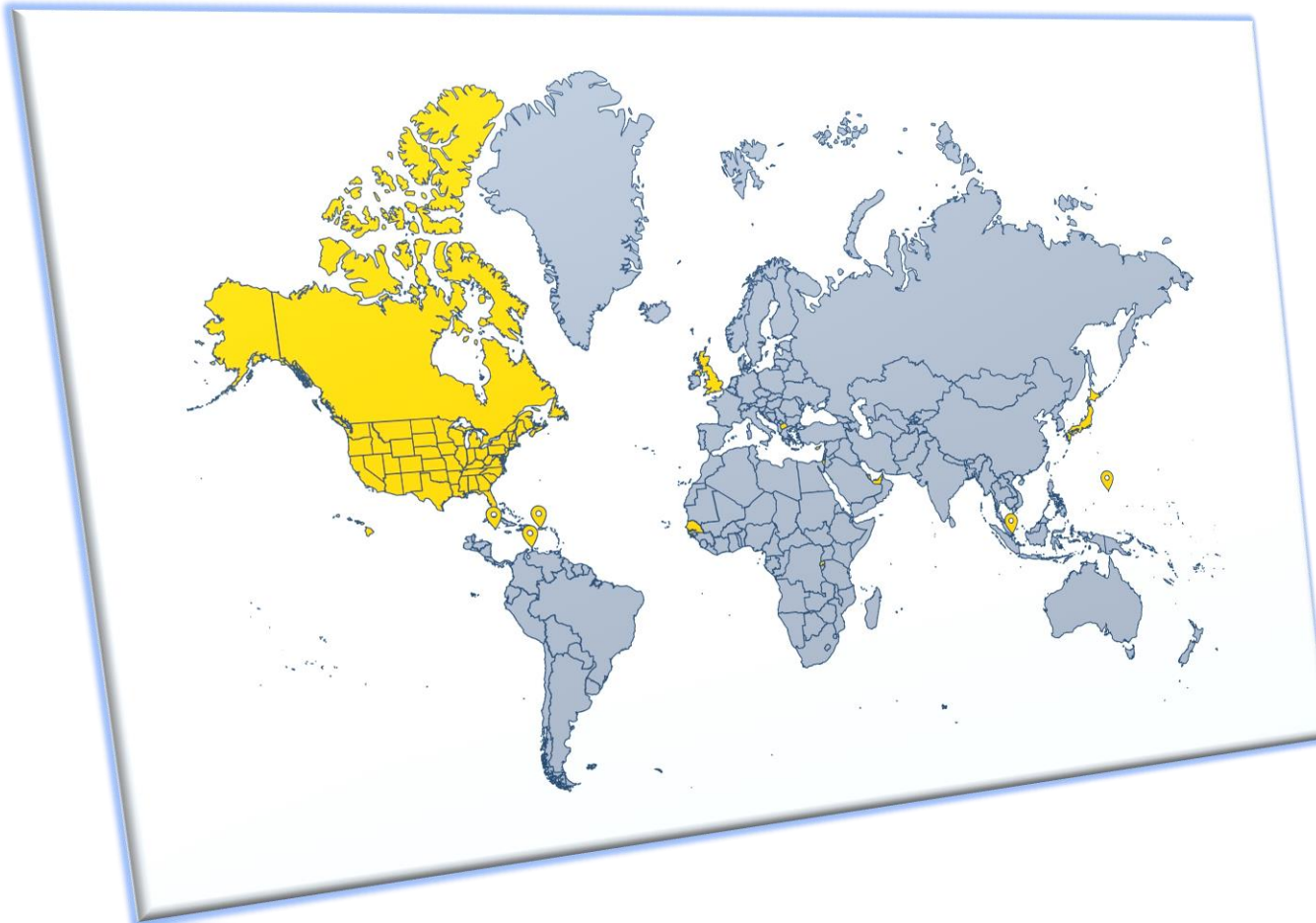
- Defines how *authenticated and immutable clinical facts* are encoded in a QR Code
 - Payload data includes immunization, lab results, etc.
 - Not an ID document, must be presented *with* one!
 - Not a “green checkmark” credential!
- Worked started before the pandemic, but accelerated and focused on Covid-19
- Open specifications
- Large ecosystem support
 - Major Electronic Health Records vendors
 - Native support in iOS and Android
 - Many jurisdiction apps (e.g., VaxiCode)

<https://smarthealth.cards/>

<https://demo-portals.smarthealth.cards/>



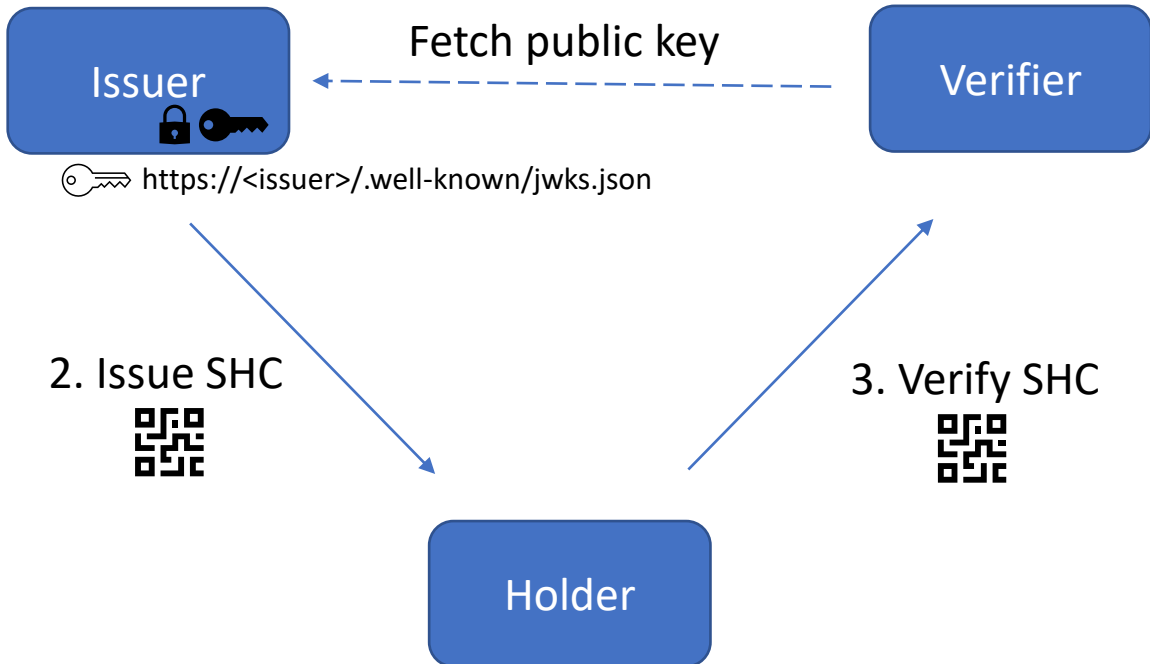
SHC Worldwide Adoption



- Aruba
- Canada
- Cayman Islands
- Cyprus
- Hong Kong
- Israel
- Japan
- North Macedonia
- Qatar
- Rwanda
- Senegal
- Singapore
- United Arab Emirates
- United Kingdom
- United States

SHC workflow

1. Generate keys



1. Issuer generates key pair and publishes public JWK set
2. Issuer creates a SHC for the user (paper or electronic QR code)
3. User presents SHC to verifier, which then:
 - Scans the QR code
 - Retrieves the issuer's public key (lookup VCI trusted issuer directory)
 - Validates the SHC sig
 - Decodes the payload



- Vaccination Credential Initiative → Verifiable Clinical Information
- Oversees the development of SMART Health Cards
- 900+ members
- Created a trust framework for SHC issuers
 - 500+ trusted issuers (public health, pharmacies, health orgs)
 - Includes the Canadian provinces and territories
 - Each issuer is vetted, tested, and audited

<https://vci.org/>



SHC Revocation

- Issuer revocation is built-in: simply remove compromised keys from the public key set, invalidating all issued SHCs
- Reported fraud led to the need of a *per-SHC* revocation feature
- Issuers can publish a card revocation list (CRL) containing the identifiers of revoked cards. Verifiers check the presence of a CRL and if supported, make sure the SHC's revocation identifier (RID) is not listed
- One method for legacy (pre-v1.2.0), one for new SHCs
 - *Legacy*: RID is derived from the card's content (hash of FHIR bundle)
 - *New*: RID is explicitly added to the card
- RID can be timestamped in the CRL to allow re-use of identifiers

VCI directory audit and snapshot

- Daily scripts for the VCI directory
 - **Audit script** reports TLS config issues, key/name duplications, additions/deletions, errors
 - https://github.com/the-commons-project/vci-directory/blob/main/logs/daily_audit.json
 - **Snapshot script** creates a signed snapshot of keys and CRLs of trusted issuers, allowing offline validation
 - https://github.com/the-commons-project/vci-directory/blob/main/logs/vci_snapshot.json
- Integrated in VCI's daily github action runs
- Detected many issues and increases trust in the directory

General Claim QR

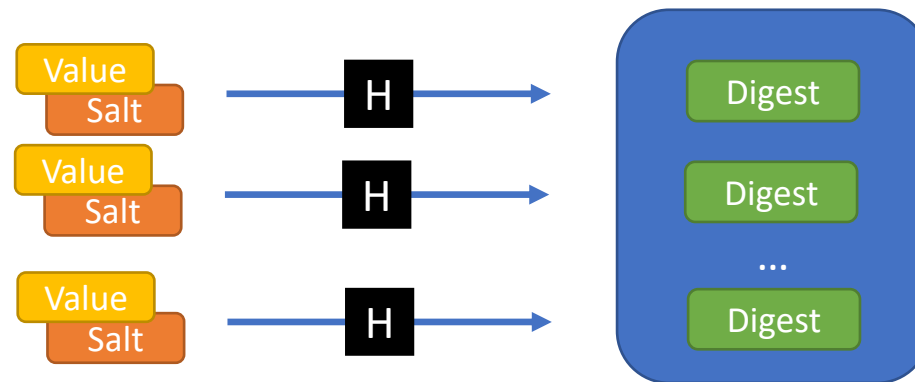
Generalized case: Claim QR

- Presenting claims as a QR code is an interesting paradigm
 - Low friction, easy to deploy
 - Inclusive tech spectrum: paper to smart wallets
 - Fills a gap between paper and online-only credentials
- Claim QR: explores encoding general claims as QR code
 - Reuses many components from the SHC framework
 - Generic JSON Web Token (JWT) payload
 - Removes some underused SHC designs (QR chunking, no VC)

<https://github.com/microsoft/claimqr>

Adding selective disclosure

- One drawback of SHC (and any conventional credential) is that they only support *all-or-nothing* disclosure
 - For Covid-19 SHC, you always have to disclosure full name and DoB, along with immunization history
- We can add *subset claim disclosure* by encoding salted hash digests into the credentials, and disclosing a subset of the salts & claim values



Selective Disclosure for Claim QR

JWT to encode:

```
{
  "iss": "https://example.org/cqr",
  "nbf": 1648226603,
  "cqv": "0.1"
}
```



```
{
  "iss": "https://example.org/cqr",
  "nbf": 1648226603,
  "cqv": "0.1",
  "claimDigests": {
    "given_name": "HhpbopKFyKFoi8clJmp9HQ",
    "middle_name": "WvVAGKAcBB3uyzQGAW-5hQ",
    "family_name": "FGPg5BX2Hx4-qS_KJ_yuw",
    "birthdate": "rXTbQvY0bOWThHx4jIfLuA"
  },
}
```

Hash(s,v)



Selectively-disclosable claims:

```
{
  "given_name": "Christian",
  "middle_name": "",
  "family_name": "Paquin",
  "birthdate": "2020-11-01"
}
```



Claim data

```
{
  "given_name": { "s": "cXqehHqWI9Y", "v": "Christian" },
  "middle_name": { "s": "6mQkh0F9p9s", "v": "" },
  "family_name": { "s": "1dGp7SxHLuo", "v": "Paquin" },
  "birthdate": { "s": "2ajylsdYJUQ", "v": "2020-11-01" }
}
```

Random salt



Selective Disclosure for Claim QR

eyJhbGciOiJFUzI1NiIsInppcCI6IkdFRiIsImtpZCI6InRybXlyWHBxWEtCWk5kMTF1T2M1LTlWMW0za0otSlRwTXhsc19ac3pCWVUifQ
.bc69DolwFibhezmaAiKiskEMoJgo0fgzEagFalooFBA03rt1cHM-z_vlvIAIARbkTcOFpaq4jxmnWCnrTEVVDSMokhSsiWksdN00takyNO
eAqk42mjkRd0RjwIYkw6KRQy_ISleLqIgZIsTPeVLyWb0CtyQLRDeML_1QVozcbhT_2Lk72V5gl8eZtsMz9OzHeJZ_WRozQocfc719Nn
Muuu_1xrg6RMEmGtqHZH-_b0VTMlnVI2MSdlct2Z2Pud8b93W6bW14vz8
.en_ISi57_dfkpaWwGSF2HALGOPS90JEq1P4LY14-aQC38CCQY9JKBnQPO5NNmK4HSjWB3JoW7RLgwKvFrjiNVQ
.Vc2xDolwGATgd_InYtVB0z4AgsGBGKNMpqGVVItaaCEQwruLigPr5bu7AQrZ8vJeUs2BDOCAQH6ruliqa4wzCKCdklDWzsMYgJaMKb
7QO52-xDrEFrtZn77owx9US9Uv-IYd7P7cRUIjZp7Q37bw3jqCEO-otoqvTF2gvHHe6H93S5997Fh2vKRzt6WqmZ7GNw



```
{
  "iss": "https://example.org/cqr",
  "nbf": 1648226603,
  "cqv": "0.1",
  "claimDigests": {
    "given_name": "HhpbopKFyKFoi8clJmp9HQ",
    "middle_name": "WvVAGKAcBB3uyzQGAW-5hQ",
    "family_name": "FGPg5BX2Hx4-qS_KJ_yuw",
    "birthdate": "rXTbQvY0bOWThHx4jlfLuA"
  },
}
```

Claim data

```
{
  "given_name": { "s": "cXqehHqWI9Y", "v": "Christian" },
  "middle_name": { "s": "6mQkh0F9p9s", "v": "" },
  "family_name": { "s": "1dGp7SxHLuo", "v": "Paquin" },
  "birthdate": { "s": "2ajylsdYIUQ", "v": "2020-11-01" }
}
```

Selective Disclosure for Claim QR

eyJhbGciOiJIUzI1NiIsInpvcCI6IHR5cGU6ImNpdGUiLCJ1eW91IjpbWmV0a0otSlRwTXhsc19ac3pCWVUifQ.
.bc69DolwFibhezmaAiKiskEMoJgoOfgzEagFalooFBA03rt1cHM-z_vlvIAIARbkTcOFpaq4jxmnWCnrTEVVDSMokhSsiWksdN00takyNO
eAqk42mjKrdORjwYkw6KRQy_ISleLqIgZIsTPeVlywB0CtyQLRDeML_1QVozcbhT_2Lk72V5gl8eZtsMz9OzHeJZ_WRozQocfc719Nn
Muuu_1xrg6RMEmGtqHZH-_b0VTMlnVI2MSdlct2Z2Pud8b93W6bW14vz8
.en_ISi57_dfkpaWwGSF2HALGOPS90JEq1P4LY14-aQC38CCQY9JKBnQPO5NNmK4HSjWB3JoW7RLgwKvFrjiNVQ
.V2wDobnAGTgrVgVrV88zCAGdGB6dNQV9XGypHdPE7bhw3fgr50187AQtX86BZF2B0200AVwQBolRqMvzVK6dJ2DygzqZCgJlyKZUj
zGJ0520REKPxZrE77wPua1594bGRD7d72Rujz29Cq370sw3fCE0AwcoqvTF2gvHHe6H93S5997Fh2vKRzt6WqmZ7GNw



```
{
  "iss": "https://example.org/cqr",
  "nbf": 1648226603,
  "cqv": "0.1",
  "claimDigests": {
    "given_name": "HhpbopKFyKFoi8clJmp9HQ",
    "middle_name": "WvVAGKAcBB3uyzQGAW-5hQ",
    "family_name": "FGPg5BX2Hx4-qS_KJ_yuw",
    "birthdate": "rXTbQvY0bOWThHx4jlfLuA"
  },
}
```



Claim data

```
{
  "given_name": { "s": "cXqehHqWI9Y", "v": "Christian" },
  "middle_name": { "s": "GhQL959p9", "v": "" },
  "family_name": { "s": "1JG_76_HU", "v": "Dip" },
  "birthdate": { "s": "2ajylsdYIUQ", "v": "2020-11-01" }
}
```

Selective Disclosure for Claim QR

eyJhbGciOiJFUzI1NiIsInppcCI6IkdFRiIsImtpZCI6InRybXlyWHBxWEtCWk5kMTF1T2M1LTlWMW0za0otSIRwTXhsd19ac3pCWVUifQ
.bc69DolwFibhezmaAiKiskEMoJgo0fgzEagFalooFBA03rt1cHM-z_vlvIAIARbkTcOFpaq4jxmnWCnrTEVVDSMokhSsiWksdN00takyNO
eAqk42mjKRd0RjwIYkw6KRQy_ISleLqIglZIsTPeVlywB0CtyQLRDeML_1QVozcbhT_2Lk72V5gl8eZtsMz9OzHeJZ_WRozQocfc719Nn
Muuu_1xrg6RMEmGtqHZH-_b0VTMlnVI2MSdlct2Z2Pud8b93W6bW14vz8
.en_ISi57_dfkpaWwGSF2HALGOPS90JEq1P4LY14-aQC38CCQY9JKBnQPO5NNmK4HSjWB3JoW7RLgwKvFrjiNVQ
.9Ew-k-tlavhsYvrVgUrT8XzC5wtVNtdDQV9X3ypeYUP-7bhw3fQcU0I8G_ntX863ZEtrnd2OQvwQBoR9MveMmw2PqgzqZCrqlyKZWj
eSjLxjv0REKXXziLEi-7wPuetsu4cGRD--dr27r_n29cp5IOsWFFeH7Aw



```
{
  "iss": "https://example.org/cqr",
  "nb": 1648226603,
  "cqv": "0.1",
  "claimDigests": {
    "given_name": "HhpbopKFyKFoi8clJmp9HQ",
    "middle_name": "WvVAGKAcBB3uyzQGAW-5hQ",
    "family_name": "FGPg5BX2Hx4-qS_KJ_yuw",
    "birthdate": "rXTbQvY0bOWThHx4jlfLuA"
  },
}
```

"HhpbopKFyKFoi8clJmp9HQ" = Hash("cXqehHqWI9Y", "Christian") ✓

Claim data

```
{
  "given_name": { "s": "cXqehHqWI9Y", "v": "Christian" },
  "middle_name": { "s": "WvVAGKAcBB3uyzQGAW-5hQ", "v": "" },
  "family_name": { "s": "FGPg5BX2Hx4-qS_KJ_yuw", "v": "" },
  "birthdate": { "s": "2ajylsdYIUQ", "v": "2020-11-01" }
}
```

Demo

<https://github.com/microsoft/ClaimQR/tree/main/sample>



Road ahead for digital ID

- Long-lived credentials with selective disclosure are coming
 - Mobile Driver License (ISO/IEC 18013-5:2021)
- More privacy features are needed
 - *Unlinkability* between issuance and presentation
 - *Derived* claims (e.g., date of birth → over-21)
- Explored in advanced identity systems
 - U-Prove (<https://microsoft.com/uprove>)
- Making its way in upcoming identity standards
 - Verifiable Credentials
<https://www.w3.org/TR/vc-data-model>

cpaquin@microsoft.com

 @chpaquin