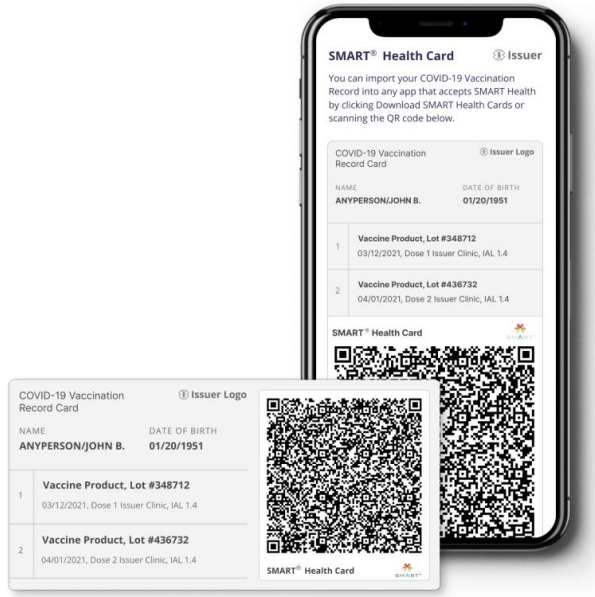
# Privacy-Friendly QR codes for Identity

Christian Paquin

@chpaquin

Microsoft Research




ICMC22

# About me

- Cryptography engineer in *MSR Security & Crypto* team
  - Privacy-preserving identity (anonymous credentials, zero-knowledge proofs)
  - Post-Quantum Cryptography
  - Searchable encryption
- Helped with the development of the SMART Health Card Framework (discussed today)

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

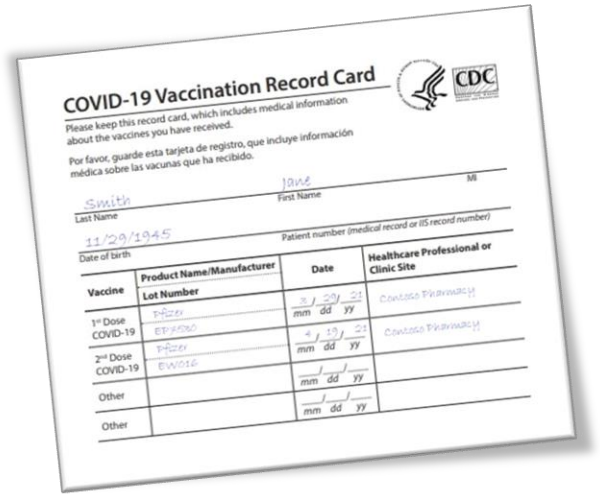
# What we will discuss today

1. SMART Health Cards (covid-19 immunization certificates)
2. Generalized privacy-preserving QR codes
3. Privacy trends in identity

# SMART Health Cards (SHC)

SMART = Substitutable Medical Applications, Reusable Technologies

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


COVID-19 Vaccination Record Card

CDC

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

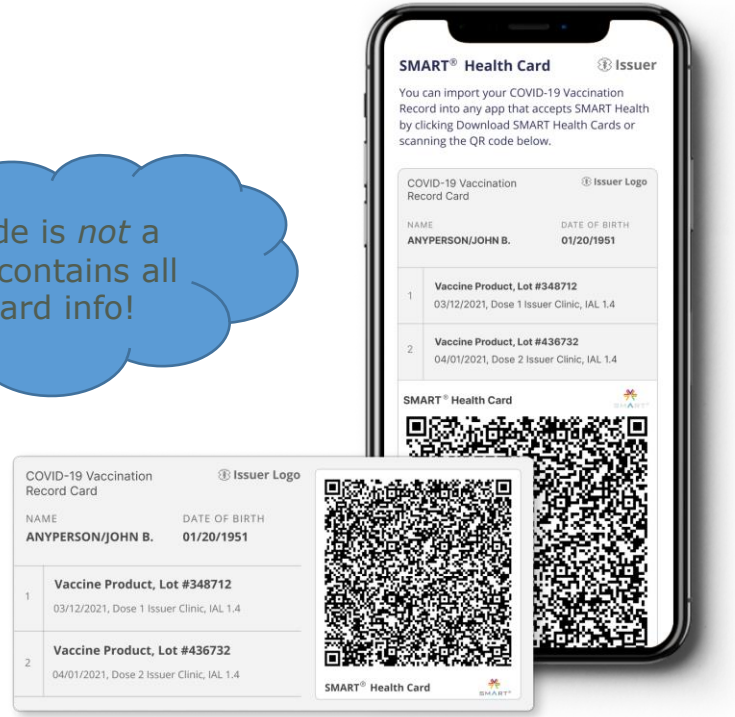
Por favor, guarde esta tarjeta de registro, que incluye información médica sobre las vacunas que ha recibido.

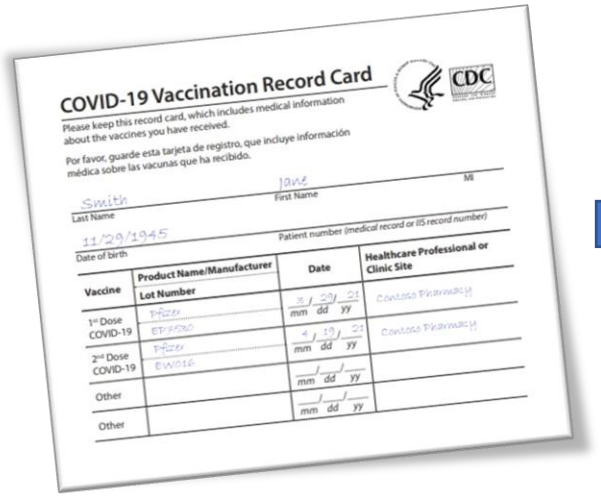
Smith (Last Name) Jane (First Name) MI

11/29/1945 (Date of birth) Patient number (medical record or IIS record number)

| Vaccine | Product Name/Manufacturer / Lot Number | Date | Healthcare Professional or Clinic Site |
|---|---|---|---|
| 1st Dose COVID-19 | Pfizer EP5360 | 3/20/21 | Contoso Pharmacy |
| 2nd Dose COVID-19 | Pfizer EW0182 | 4/10/21 | Contoso Pharmacy |
| Other | | | |
| Other | | | |

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





COVID-19 Vaccination Record Card

CDC

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.

Smith (Last Name) Jane (First Name) MI

11/29/1945 (Date of birth) Patient number (medical record or IIS record number)

| Vaccine | Product Name/Manufacturer / Lot Number | Date | Healthcare Professional or Clinic Site |
|---|---|---|---|
| 1st Dose COVID-19 | Pfizer / EP7030 | 3/29/21 | Contoso Pharmacy |
| 2nd Dose COVID-19 | Pfizer / EW016 | 4/19/21 | Contoso Pharmacy |
| Other | | | |
| Other | | | |

1. Encode the immunization info as a FHIR bundle

FHIR = Fast Health Interoperability Resources

## 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>
        }
    }
}
```


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":"d630duSMWmVfmOtrMKZX6izJfcampjK1h0D4jrXxJwU"
```

## 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>
        }
    }
}
```

```
eyJ6aXAiOiJERUYiLCJhbGciOiJFUzI1NiIsImtpZCI6ImQ2MzBkdVNNV21WZm1PdHJNS1pYNml6SmZjYW1waksxaDBENGpyWHhKd1UifQ.
3VJNj9MwEP0raLgmTZx2-5ETonCAA6q0CxxQD44zbYz8EdlO2rLKf2ectguI3R44Ivky9pvn997MI0jvoYQmhNaXWeY1d6FBrkIzEdzVPsMj161CnxGwQwcJmMpByeZslc-
XxXyRL_IEegHlI7wR1gQ8Bii_PTEeDofJYTqxbp8VOVtmwmGNJkiufNYz2CYQTi3Gji_o5E7ySuH6CUPfvSjt9blIY3ETJ7XujPzBg7TmJlDYXtZsFUX9knnfVd9RhOhv10hHKn3kKWE2ySeM-
OLt287UCiPGobedE_gwuoLLw9UlCKsUsZ2V0AfuRNaJuVPqs1MEuPaXOQGuxTPEG7JD_XEeXOOZhGupiA_utQwNvexljyZG-
5EbhO1AvipJnt_xECnYanaXMpYWKxiG5FkR7LaID38m6wMPnR9dxp0JGOfScyGkwbWtRwZha2n2o15_8gH1ZftoII1ajHsSA828rDPRH4lAjJ1Q5EsYtkMC7cX5KGeHDk3U9ntwBLJCdG58imYfpD5
TFCzNp9FwAsqGT52uaKVLeL9Z3E1j4C26nXU6XpJALoJ18Z9a-lbxGO2aNtx6-2rTcKe5OFFy25fCK_7D8GYp-yu8rzmb_3t2dIbhJw
.1pd_USQlfJwOhwZnsg0vL_c3zu8ScIl2YLxZk-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

eyJ6aXAiOiJERUYiLCJhbGciOiJFUzI1NiIsImtpZCI6ImQ2MzBkdVNNV21WZm1PdHJNS1pYNml6SmZjYW1waksxaDBENGpyWHhKd1UifQ.
3VJNj9MwEP0raLgmTZx2-5ETonCAA6q0CxxQD44zbYz8EdlO2rLKf2ectguI3R44Ivky9pvn997MI0jvoYQmhNaXWeY1d6FBrkIzEdzVPsMj161CnxGwQwcJmMpByeZslc-
XxXyRL_IEegHlI7wR1gQ8Bii_PTEeDofJYTqxbp8VOVtmwmGNJkiufNYz2CYQTi3Gji_o5E7ySuH6CUPfvSjt9blIY3ETJ7XujPzBg7TmJlDYXtZsFUX9knnfVd9RhOhv10hHKn3kKWE2ySeM-
OLt287UCiPGobedE_gwuoLLw9UlCKsUsZ2V0AfuRNaJuVPqs1MEuPaXOQGuxTPEG7JD_XEeXOOZhGupiA_utQwNvexljyZG-
5EbhO1AvipJnt_xECnYanaXMpYWKxiG5FkR7LaID38m6wMPnR9dxp0JGOfScyGkwbWtRwZha2n2o15_8gH1ZftoII1ajHsSA828rDPRH4lAjJ1Q5EsYtkMC7cX5KGeHDk3U9ntwBLJCdG58imYfpD5
TFCzNp9FwAsqGT52uaKVLeL9Z3E1j4C26nXU6XpJALoJ18Z9a-lbxGO2aNtx6-2rTcKe5OFFy25fCK_7D8GYp-yu8rzmb_3t2dIbhJw
.1pd_USQlfJwOhwZnsg0vL_c3zu8ScIl2YLxZk-Su7MRK1EIeREOkov831duM1FkmORMiucbMozw0IVZK0OmR1A

6. Encode the JWS as a numeric QR

shc:/567629095243206034602924374044603122295953265460346029254077280433602870286471674522280928643605327721625541333341050442456404355527293338046744336463093864456144420474526270755223212433266776422759305504406057360106412933611232742435036952315864394575050008243966652220200968032275753623070777534477112455633405693130570556547158722806370707287362761267736512121032280361736644366459335243425644045509252169622877245577413570326104090422657526743674542964326721765645706354004375437637315028245658276328107437045836112160605035392456236657294439687553671141344171647464263329626072573344770522443639600626616050660824107638722709224035577338617112536328440624392910437261357721581039642963234443714570254043126265655741551237593459730403592730650662304224057638501320034317105111040226035266653565524505874726631317412406322307040704505410320577237335229724135687004322472355243343626727539352426102923504324564334344559267267602050727136743373567563617645260008245359340420736067296571507522265445265524332674442307560260825623710315228230611640974323565371255756703292634573854762662745342713774455952056505660408501158270445577166282804526127703820110511692335372707632061290436082470447162322105443083026562723620640126571742131292255260811606444576723083925227733671225742070682639080572523041315631124506240461072205096543400943672920316629041145125200635375263405523371750900056939543056083425257605085722305010231126446700767211697764535006710555285359297401046755504038366357297434597445657058037331505406777211385428630544317545620038721032373004242856372434626673110604557232042562643437326072545332667774032841453003346437042

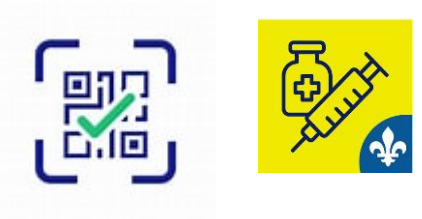
7. Create the QR code image

# SMART Health Cards Framework

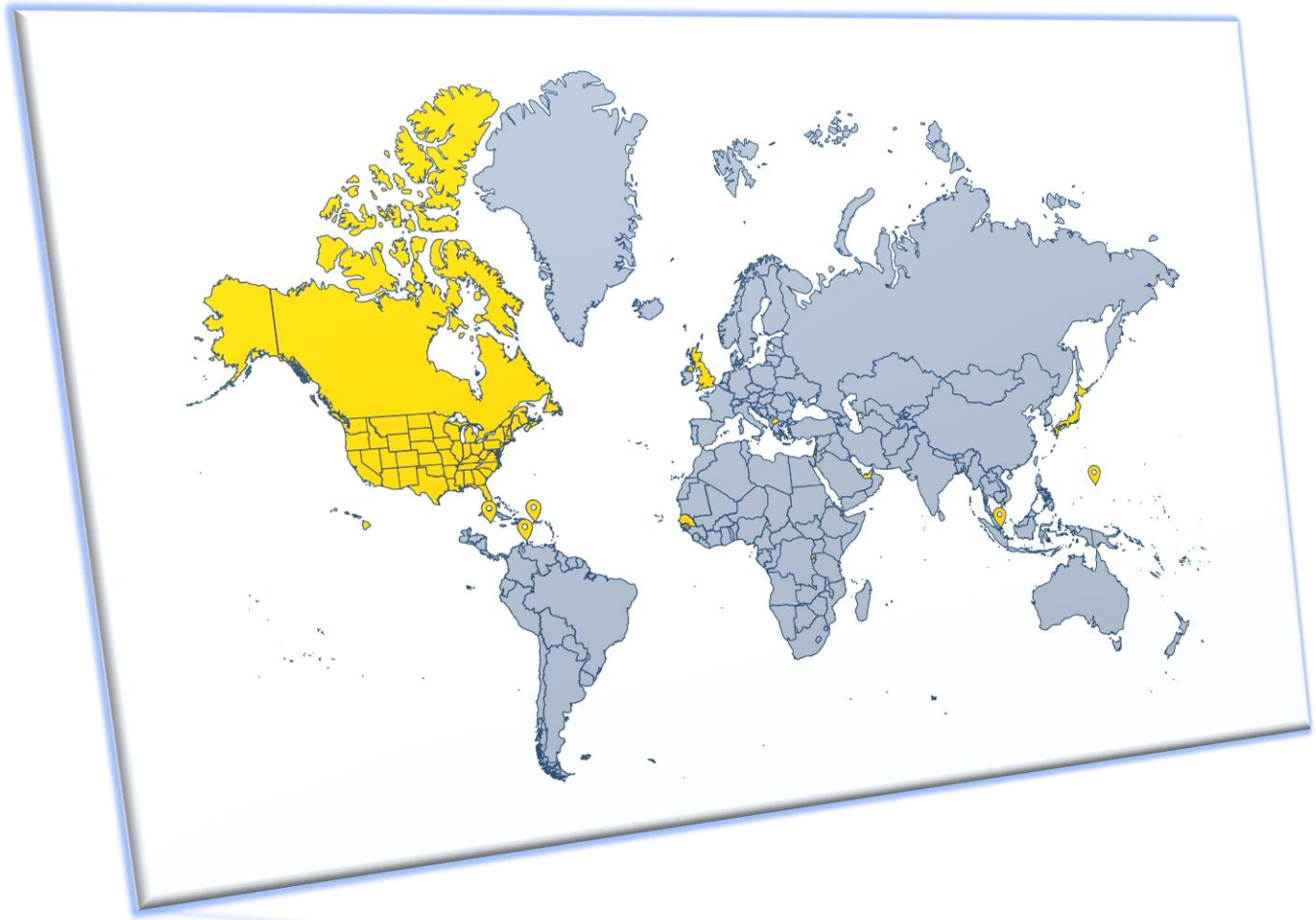
- Defines how *authenticated and immutable clinical facts* are encoded in a QR Code
  - Payload data includes immunization, lab results, etc.
  - Not an ID document, must be presented *with* one
  - Not a “green checkmark” credential
- Worked started before the pandemic, but accelerated and focused on 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/



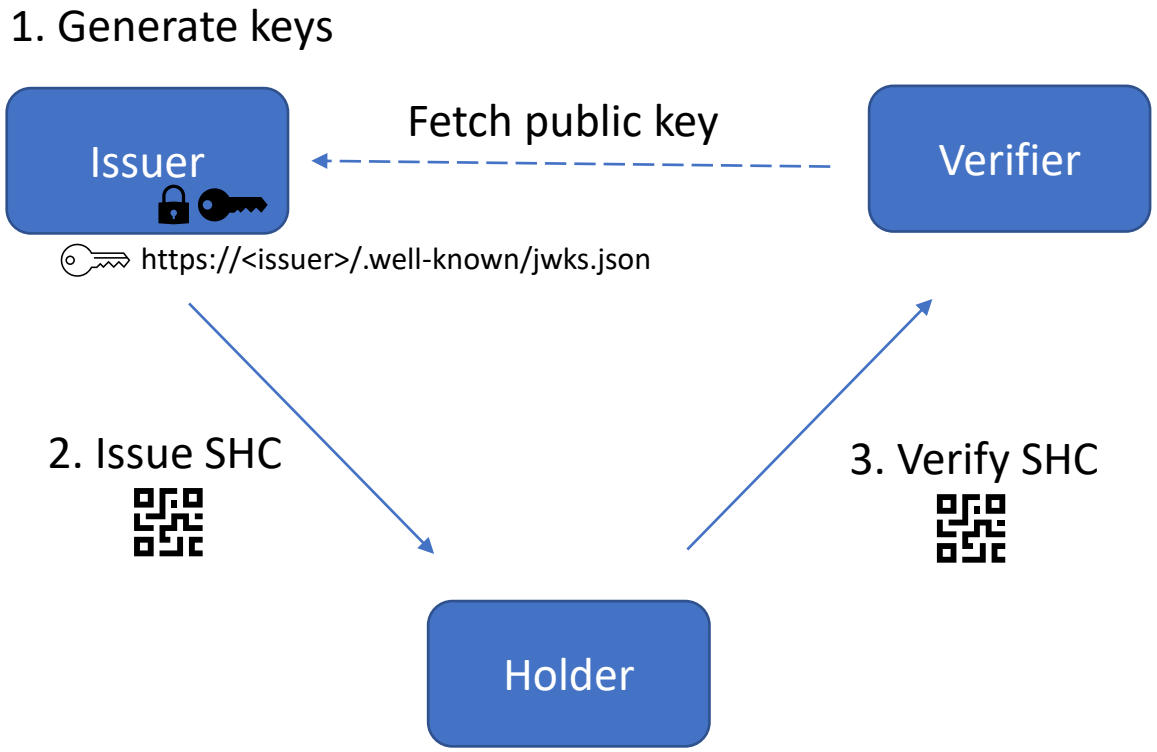
# SHC Worldwide Adoption



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

Source: https://vci.org/

# SHC workflow




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

# VCI

- “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)
  - 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

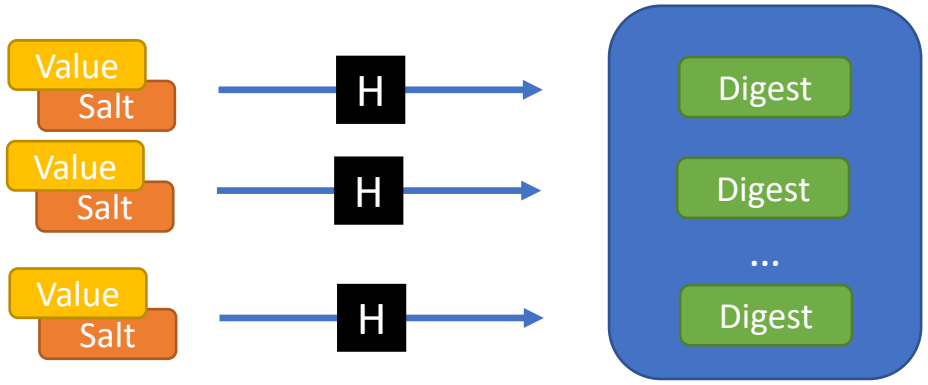
# Generalized case: Claim QR

- 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"
}
```

Random salt

Claim data

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

# Selective Disclosure for Claim QR

eyJhbGciOiJFUzI1NiIsInppcCI6IkRFRiIsImtpZCI6InRybXlyWHBxWEtCWk5kMTF1T2M1LThWMW0za0otSlRwTXhsd19ac3pCWVUifQ
.bc69DoIwFIbhezmzAiKiskEMoJgo0fgzEagFalooFBA03rt1cHM-z_vlvIAIARbkTcOFpaq4jxmnWCnrTEVVDSMokhSsiWksdN00takyN0
eAqk42mjKRd0RjwlYkw6KRQy_ISIeLqIgZlsTPeVLywB0CtyQLRDeML_1QVozcbhT_2Lk72V5gI8eZtsMz9OzHeJZ_WRozQocfc719Nn
Muuu_1xrg6RMEmGtqHZH-_b0VTMlnVl2MSdlct2Z2Pud8b93W6bW14vz8
.en_ISi57_dfkpaWwGSF2HALGOPS90JEq1P4LY14-aQC38CCQY9JKBnQPO5NNmK4HSjWB3JoW7RLgwKvFrjiNVQ
.Vc2xDoIwGATgd_lnYtVB0z4AgsGBGKNMpqGVVltaaCEQwruLigPr5bu7AQrZ8vJeUs2BDOCAQH6ruIiqa4wzCKCdklDWzsMYgJaMKb
7QO52-xDrEFrtZn77owx9US9Uv-IYd7P7cRUljZp7Q37bw3jqCEO-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": "rXTbQvY0bOWThHx4jIfLuA"
   },
}
```


Claim data


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

# Selective Disclosure for Claim QR

eyJhbGciOiJFUzI1NiIsInppcCI6IkRFRiIsImtpZCI6InRybXlyWHBxWEtCWk5kMTF1T2M1LThWMW0za0otSlRwTXhsd19ac3pCWVUifQ
.bc69DoIwFIbhezmzAiKiskEMoJgo0fgzEagFalooFBA03rt1cHM-z_vlvIAIARbkTcOFpaq4jxmnWCnrTEVVDSMokhSsiWksdN00takyN0
eAqk42mjKRd0RjwlYkw6KRQy_ISIeLqIgZlsTPeVLywB0CtyQLRDeML_1QVozcbhT_2Lk72V5gI8eZtsMz9OzHeJZ_WRozQocfc719Nn
Muuu_1xrg6RMEmGtqHZH-_b0VTMlnVl2MSdlct2Z2Pud8b93W6bW14vz8
.en_ISi57_dfkpaWwGSF2HALGOPS90JEq1P4LY14-aQC38CCQY9JKBnQPO5NNmK4HSjWB3JoW7RLgwKvFrjiNVQ
.[illegible]
[illegible]qoqvTF2gvHHe6H93S5997Fh2vKRzt6WqmZ7GNw

```
{
  "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"
  },
}
```

Claim data

```
{
  "given_name": { "s": "cXqehHqWI9Y", "v": "Christian" },
  [redacted]
  [redacted]
  " birthdate": { "s": "2ajyIsdYJUQ", "v": " 2020-11-01" }
}
```

# Selective Disclosure for Claim QR

eyJhbGciOiJFUzI1NiIsInppcCI6IkRFRiIsImtpZCI6InRybXlyWHBxWEtCWk5kMTF1T2M1LThWMW0za0otSlRwTXhsd19ac3pCWVUifQ
.bc69DoIwFIbhezmzAiKiskEMoJgo0fgzEagFalooFBA03rt1cHM-z_vlvIAIARbkTcOFpaq4jxmnWCnrTEVVDSMokhSsiWksdN00takyN0
eAqk42mjKRd0RjwlYkw6KRQy_ISIeLqIgZlsTPeVLywB0CtyQLRDeML_1QVozcbhT_2Lk72V5gI8eZtsMz9OzHeJZ_WRozQocfc719Nn
Muuu_1xrg6RMEmGtqHZH-_b0VTMlnVl2MSdlct2Z2Pud8b93W6bW14vz8
.en_ISi57_dfkpaWwGSF2HALGOPS90JEq1P4LY14-aQC38CCQY9JKBnQPO5NNmK4HSjWB3JoW7RLgwKvFrjiNVQ
. 9Ew-k-tIavhsYvrVgUrT8XzC5wtVNtdDQV9X3ypeYUP-7bhw3fQcU0I8G_ntX863ZEtrnd2OQvwQBoR9MveMmw2PqgzqZCrqlyKZWj
eSjLxjv0REKXXziLEi-7wPuetsu4cGRD--dr27r_n29cp5IOsWFFeH7Aw


```
{
  "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"
  },
}
```


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

Claim data


```
{
  "given_name": { "s": "cXqehHqWI9Y", "v": "Christian" },
  [redacted]
  [redacted]
  " birthdate": { "s": "2ajyIsdYJUQ", "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)
  - Selective-disclosure JWT https://github.com/oauthstuff/draft-selective-disclosure-jwt/
- 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