Age verification and privacy. An impossible equation?

Olivier Blazy

Selected Areas in Cryptography 2024

Age Verification

More and more pressing issue

- "Internet should not be a place outside the law"
  - How to have the same limits/access online as in the physical world
- Several issues
  - Who is behind the screen?
    - 1 user = 1 login = 1 person?
  - What kind of guarantees do we want?
    - Block all underage? Let all legit users access? Both?
  - Do we take the physical world as a guideline? Should we do better?
    - Should we leak someone name when checking their age? To whom?
    - Should age verifiers be able to see who is consulting what?

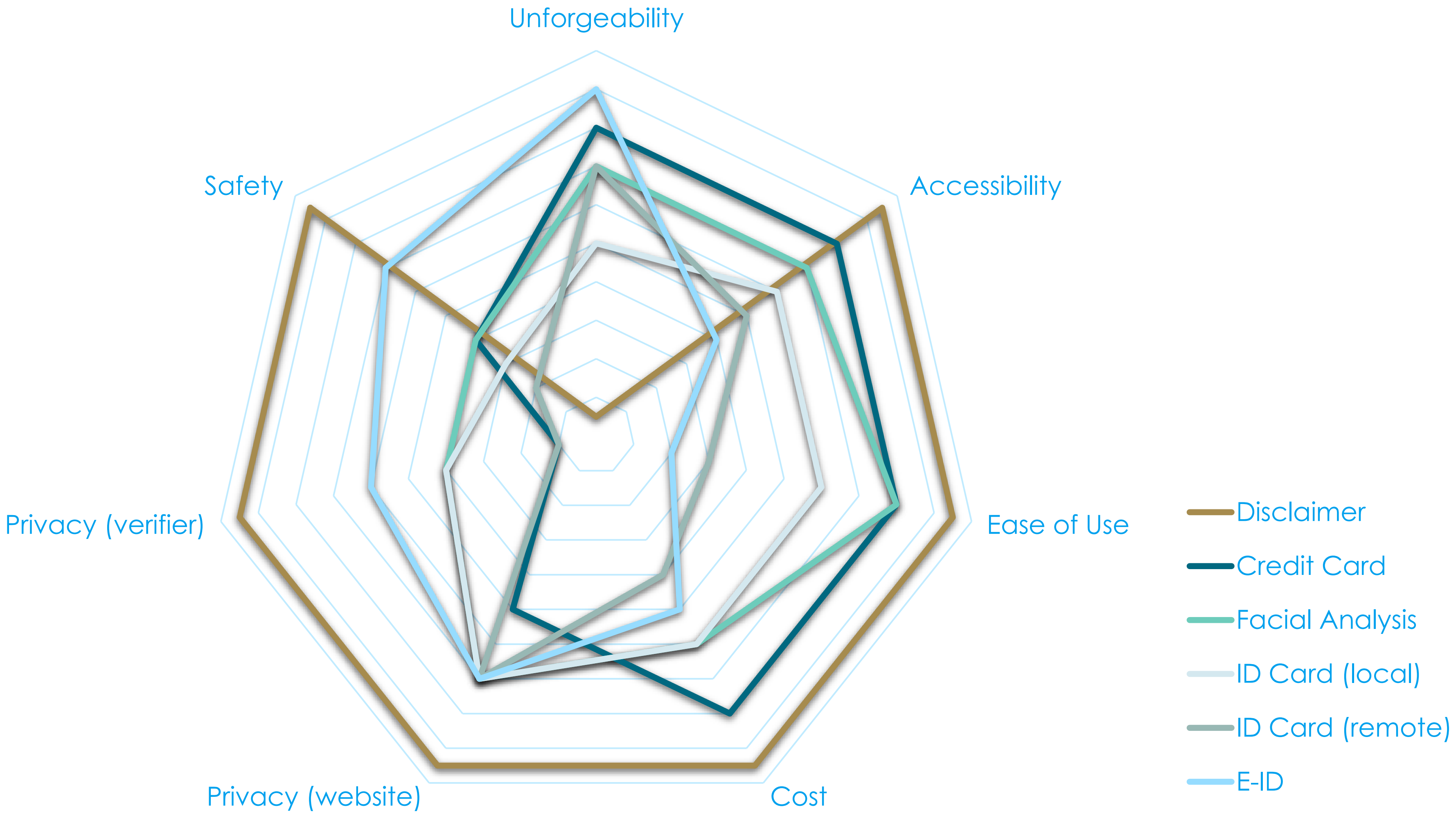
The problem starts to be legislated in many countries... (CA: Bill S-210)

The cryptographer’s answer

Use anonymous credentials!

Well… that did not go so well…

A very complex balance



Unforgeability

It should be impossible to

- produce a valid token if under the threshold

  Use of signature mechanism / certificate

  A signature is not anonymous/pseudonymous normally

- replay a token

  Interaction are necessary

  There should be a nonce to avoid replay

  (This could be alleviated (partially) using secure environments)

Protection against web site

The web site should not learn

- Our identity

  in the physical world this is sadly not the case: Selective attributes

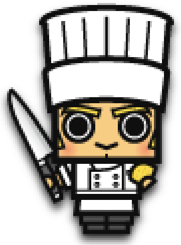
- Our age (Knowing you are above 18/21, does not mean knowing you are 38)

  The answer is just a bit and not the age. (GDPR: Minimisation)

- Who certified the age

  Signatures should be anonymous (group/ring signatures)

Protection against the verifier



The verifier should not learn:

- Which age is required (SN > 15 vs X-site > 18)

    There are 3 thresholds (in the GDPR), one could simply request the 3…

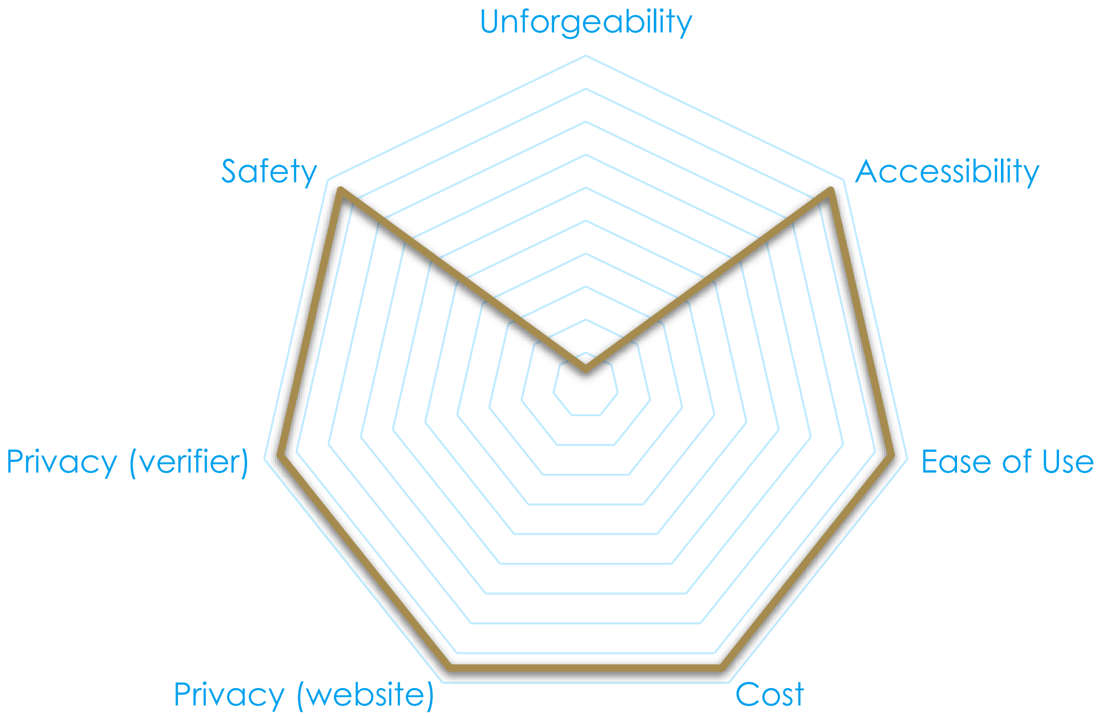
- Which website is making the request

    The challenge should not contain info about the website

    No direct communication between the website and the verifier (attention aux x-referral…)

Physical world: Nobody knows what you are going to do with your id card

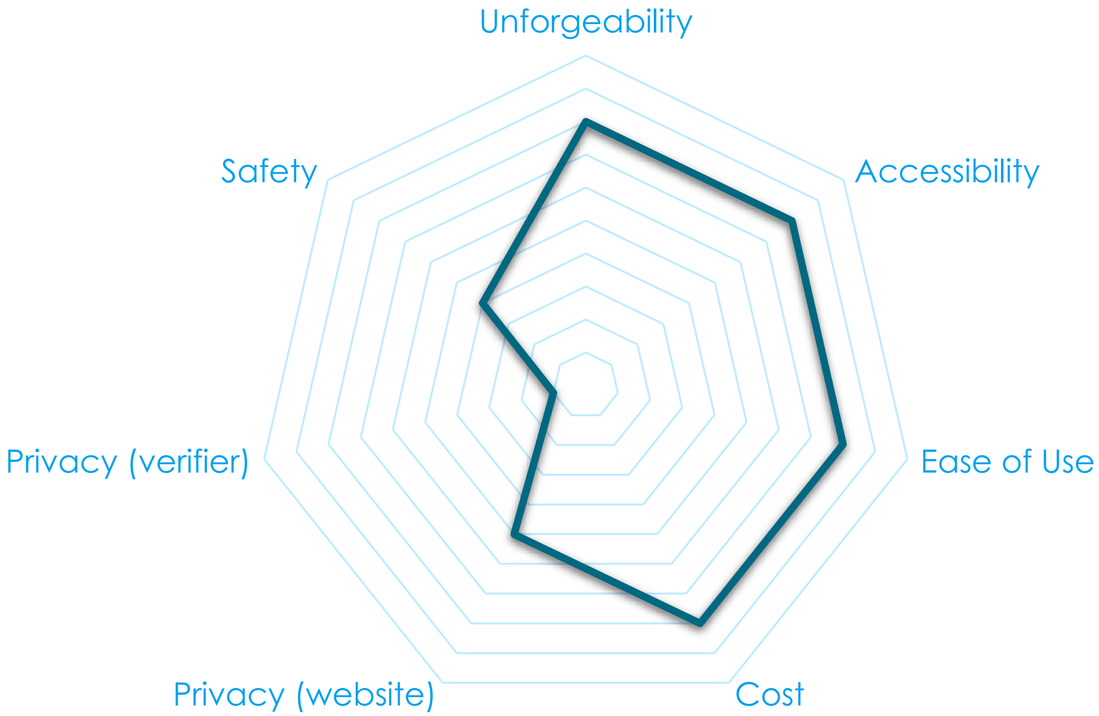
Verification with: A disclaimer



Just a Yes / No box

- **UF:** Everybody can lie, prevent accidental access
- **Acc:** No special expectation
- **Ease of Use:** One click
- **Cost:** Marginal
- **Website:** Learns nothing
- **Verifier:** None
- **Safety:** No information to leak

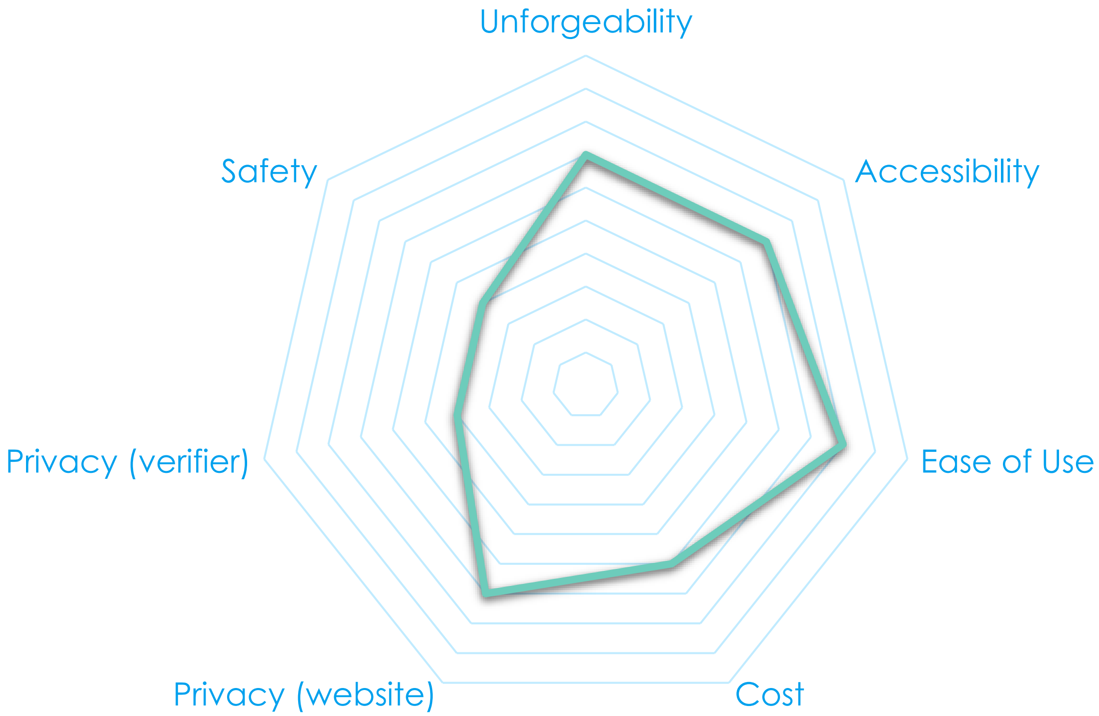
Verification with: A Credit Card



Testing a 0€ payment

- **UF:** Banking System. But 16+
- **Acc:** Need a bank account
- **Ease of Use:** Very easy
- **Cost:** Small processing fee (~0.10€)
- **Website:** Can Learn nothing
- **Verifier:** Learns both the account and the website
- **Safety:** Bad actors can get CC details, risk increases in case of 3rd party ads

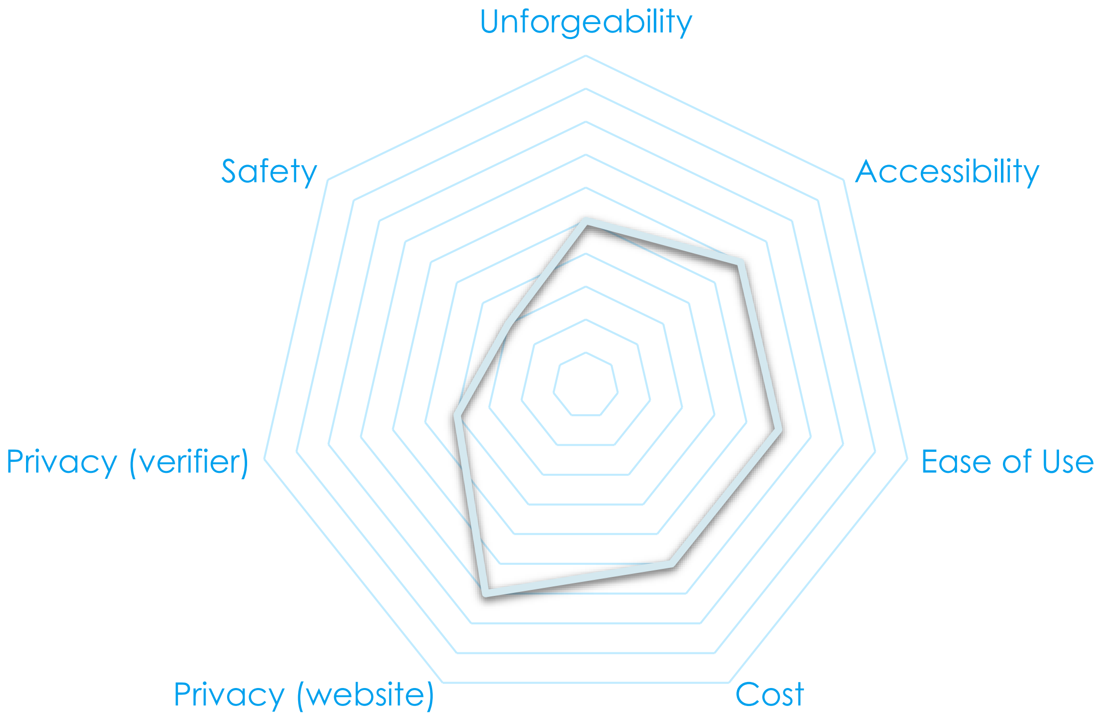
Verification with: Facial Analysis



Using the webcam to estimate the user age

- **UF:** Good for users far from the limit
- **Acc:** Need a webcam
- **Ease of Use:** Careful of racial/gender biases
- **Cost:** ~0.30€
- **Website:** Can lean nothing
- **Verifier:** Learns the website
- **Safety:** Opening a webcam can help phishing

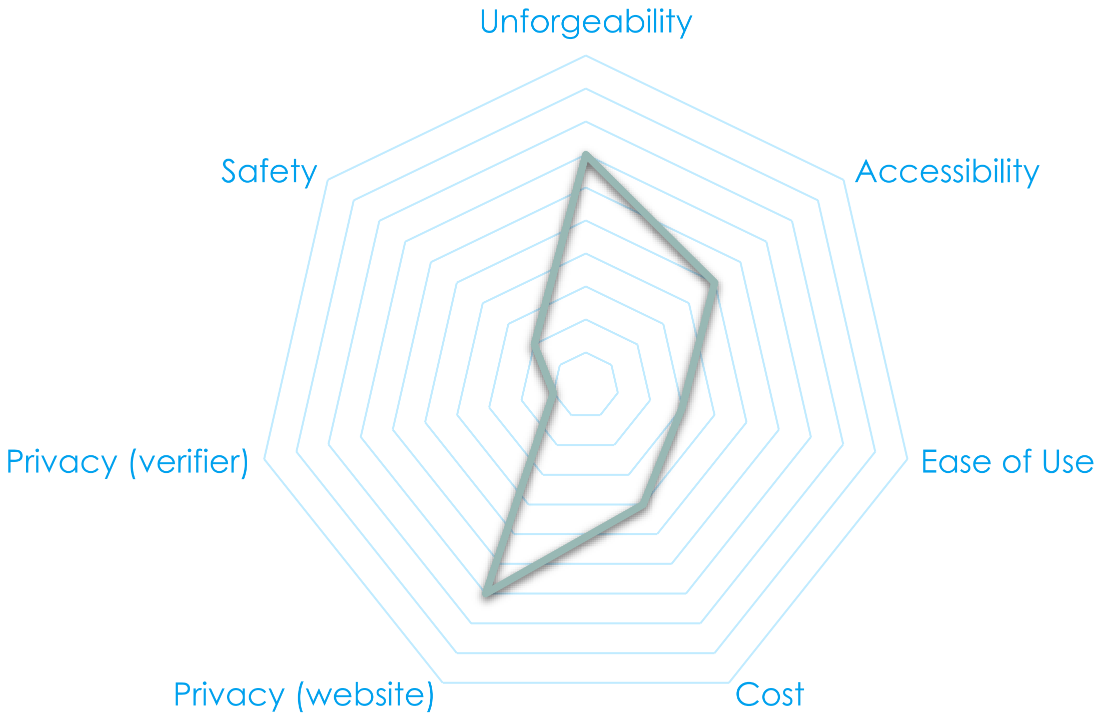
Verification with: An id-card, locally



Local analysis of an ID-Card

- **UF:** ID-card are hard to counterfeit, but lots of scans are available
- **Acc:** Need a *local* id-card
- **Ease of Use:** Need a way to scan the doc
- **Cost:** Marginal
- **Website:** Can learn nothing
- **Verifier:** Learns the website
- **Safety:** Reliant on the fact that the analysis is indeed local

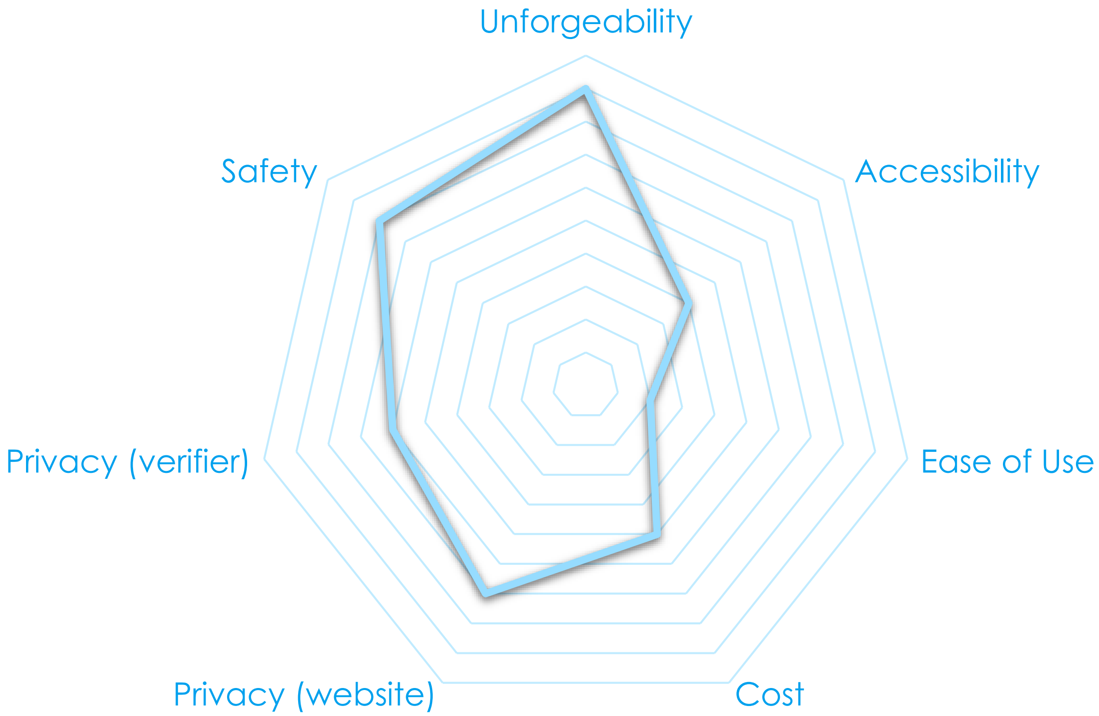
Verification with: A, ID-card, remotely



Sending an ID-card, and pictures remotely

- **UF:** Strenuous process, hard to counterfeit
- **Acc:** Need an id-card, webcam
- **Ease of Use:** Quite long (10-15 min)
- **Cost:** 1-2€
- **Website:** Learns the verifier
- **Verifier:** Has the ID, the website
- **Safety:** Bad actors have copy of id documents, even more dangerous in case of 3rd party ads.

Verification with: E-ID



Let's suppose access to an ID document, compliant with eIDAS (EU digital identity).

- **UF:** As secure as a Credit Card
- **Acc:** Need a reader, and an id-card
- **Ease of Use:** Contactless
- **Cost:** Marginal
- **Website:** Learns nothing
- **Verifier:** Local check
- **Safety:** Bad actors can create flawed apps, but still are limited by the secure design of the id

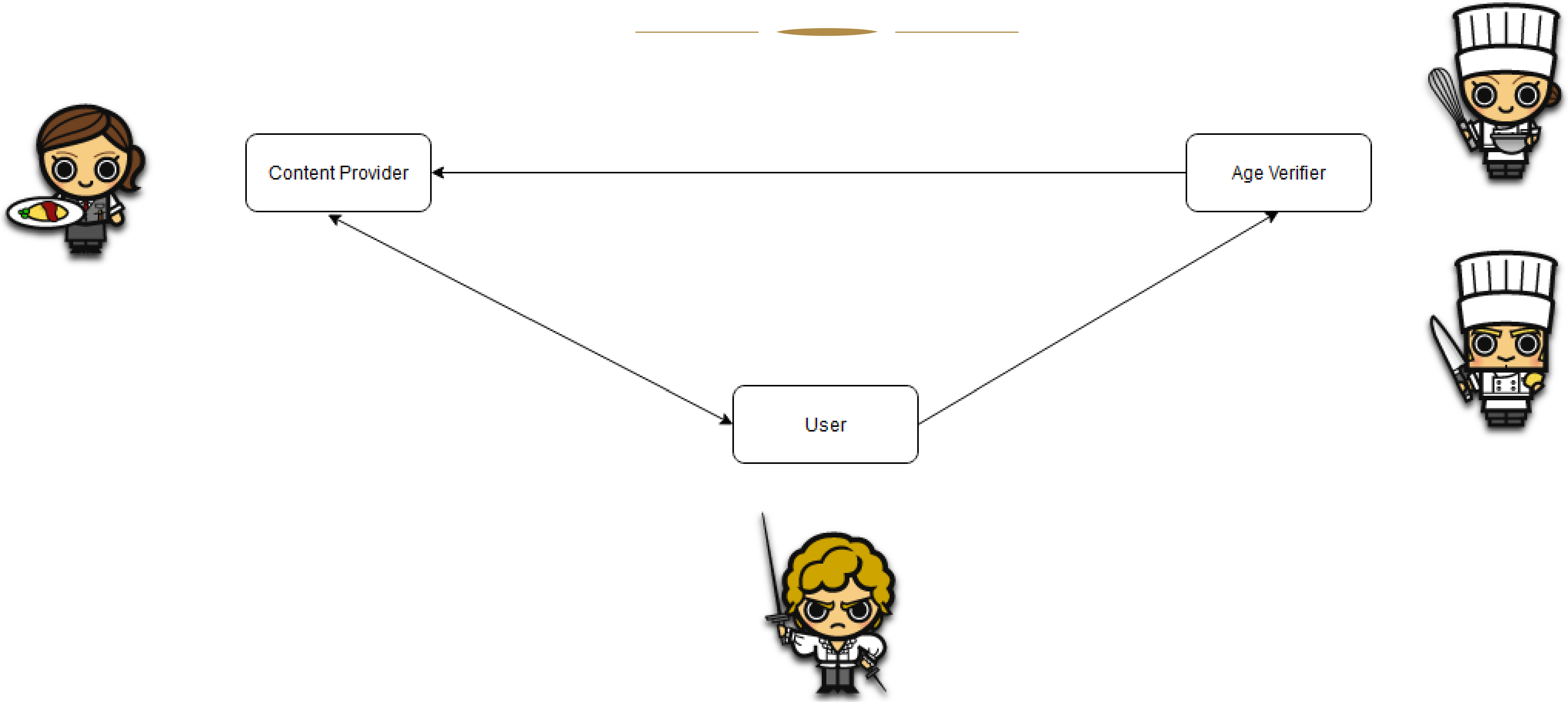
Double-Anonymity to the Rescue

Proof of Concept developed with CNIL (French DPA) and PEReN

- Allows **standardized** protection of privacy
  - Very marginal overhead cost:
    - A yearly certification by regulators
    - A little more costly communication but negligible

- **Does not change** the flow / nature of the verification
  - No interference with the business model / billing
  - No need for dedicated “extra” apps

Workflow of past versions

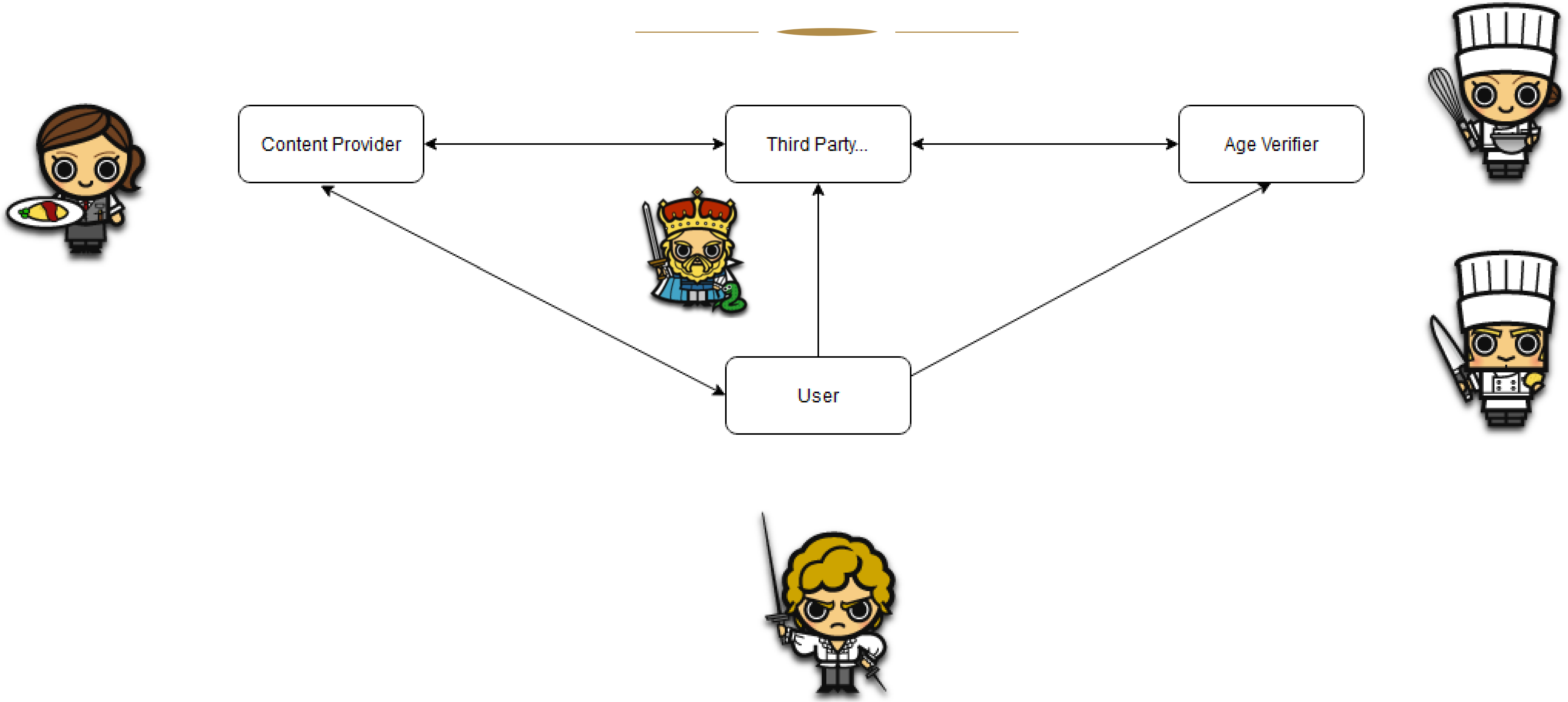


Content Provider

Age Verifier

User

Not a solution



Content Provider

Third Party...

Age Verifier

User

PETs for authentication: Group Signatures

Signatures are a well known primitive: An entity authenticates a message so that everybody can check that it had been approved.

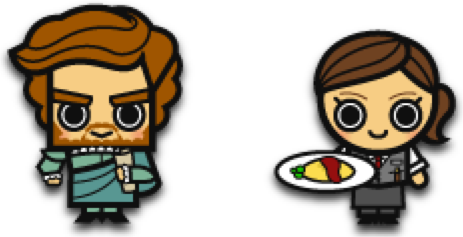
Problem: If a bank authenticates an ID? Then the website learns you are a client of said bank…

Group signatures allow to sign “anonymously” for the group. So anyone can check that an ID was approved by someone in the group without knowing by whom.

A special authority (Opener) can revoke this anonymity in case of misbehavior (approving underage access).

Group Signatures: Security (Anonymity)

```
Experiment $\mathsf{Exp}^{\mathsf{anon}-b}_{\mathsf{GS},\mathcal{A}}(\mathfrak{K})$
1. $(\mathsf{pk}, \mathsf{msk}, \mathsf{skO}) \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
2. $(m, i_0, i_1) \leftarrow \mathcal{A}(\mathtt{FIND}, \mathsf{pk}, \mathsf{msk} : \mathsf{joinP}, \mathsf{corrupt}, \mathsf{sign})$
3. $\sigma \leftarrow \mathsf{Sign}(\mathsf{pk}, i_b, m, \mathsf{sk}[i])$
4. $b' \leftarrow \mathcal{A}(\mathtt{GUESS}, \sigma : \mathsf{joinP}, \mathsf{corrupt}, \mathsf{sign})$
5. IF $i_0 \notin \mathsf{HU}$ OR $i_1 \notin \mathsf{HU}$ RETURN $0$
6. RETURN $b'$
```



Experiment:

1) Generate keys

2) The adversary is the authority and so has (msk) and can corrupt users. After a while, it selects 2 honest user and a target message.

3) We pick one, and sign the message in its name

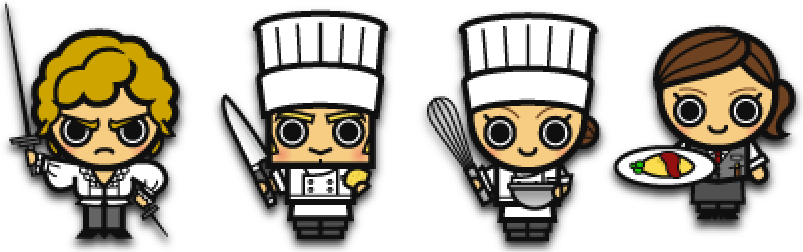
4) The adversary tries to guess who we picked

5) If it didn't corrupt one of the user, then we test its answer.

Group Signatures: Security (Unforgeability)

```
(a) Experiment $\mathsf{Exp}^{\mathsf{tr}}_{\mathsf{GS},\mathcal{A}}(\mathfrak{K})$
1. $(\mathsf{pk}, \mathsf{msk}, \mathsf{skO}) \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
2. $(m, \sigma) \leftarrow \mathcal{A}(\mathsf{pk} : \mathsf{joinA}, \mathsf{joinP}, \mathsf{corrupt}, \mathsf{sign}, \mathsf{open})$
3. IF $\mathsf{Verif}(\mathsf{pk}, m, \sigma) = 0$, RETURN 0
4. IF $\exists j \notin \mathsf{CU} \cup \mathcal{S}[m]$,
         $\mathsf{Open}(\mathsf{pk}, m, \sigma, \mathsf{skO}) = (j, \Pi)$
     RETURN 1
5. ELSE RETURN 0
```

A signature should point to a key known by the signer



```
(b) Experiment $\mathsf{Exp}^{\mathsf{nf}}_{\mathsf{GS},\mathcal{A}}(\mathfrak{K})$
1. $(\mathsf{pk}, \mathsf{msk}, \mathsf{skO}) \leftarrow \mathsf{Setup}(1^{\mathfrak{K}})$
2. $(m, \sigma) \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{msk}, \mathsf{skO} : \mathsf{joinP}, \mathsf{corrupt}, \mathsf{sign})$
3. IF $\mathsf{Verif}(\mathsf{pk}, m, \sigma) = 0$ RETURN 0
4. IF $\exists i \in \mathsf{HU} \setminus \mathcal{S}[m]$,
         $\mathsf{Open}(\mathsf{pk}, m, \sigma, \mathsf{skO}) = (i, \Pi)$
     RETURN 1
5. ELSE RETURN 0
```

Not even an authority should be able to incriminate an honest user

Group Signature, it's just a Zero knowledge proof on a Zero Knowledge proof

A digital signature is the ZKPK of a secret key associated for a specific tag (message)

In other words given sk,vk, you generate P:(sk, <vk,m>)

A group signature consists in proving for a tag (message) m, that you know a secret key (usk) associated with a public key (uvk) that has been signed by a manager.

So that there exist some P:(sk, <vk,uvk>), and some Q:(usk,<uvk,m>).

You don't want to give away uvk for anonymity so just a Q:(usk,<P,vk,m>).

Very high overview of the PoC

A meta-authority certifies the verifiers

➡ Allow verifiers to prove they operate within some legislative framework

When a user accesses a website, they receive a challenge

➡ Separation + Unicity

They forward this challenge to a verifier of their choice

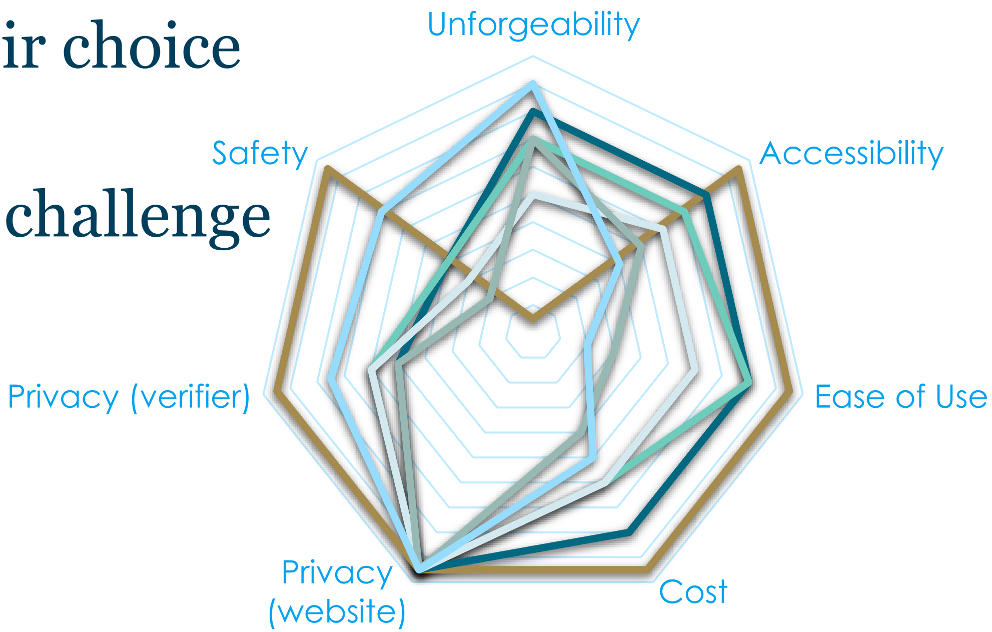
➡ Separation + Choice

The verifier signs (for the meta-authority) the challenge

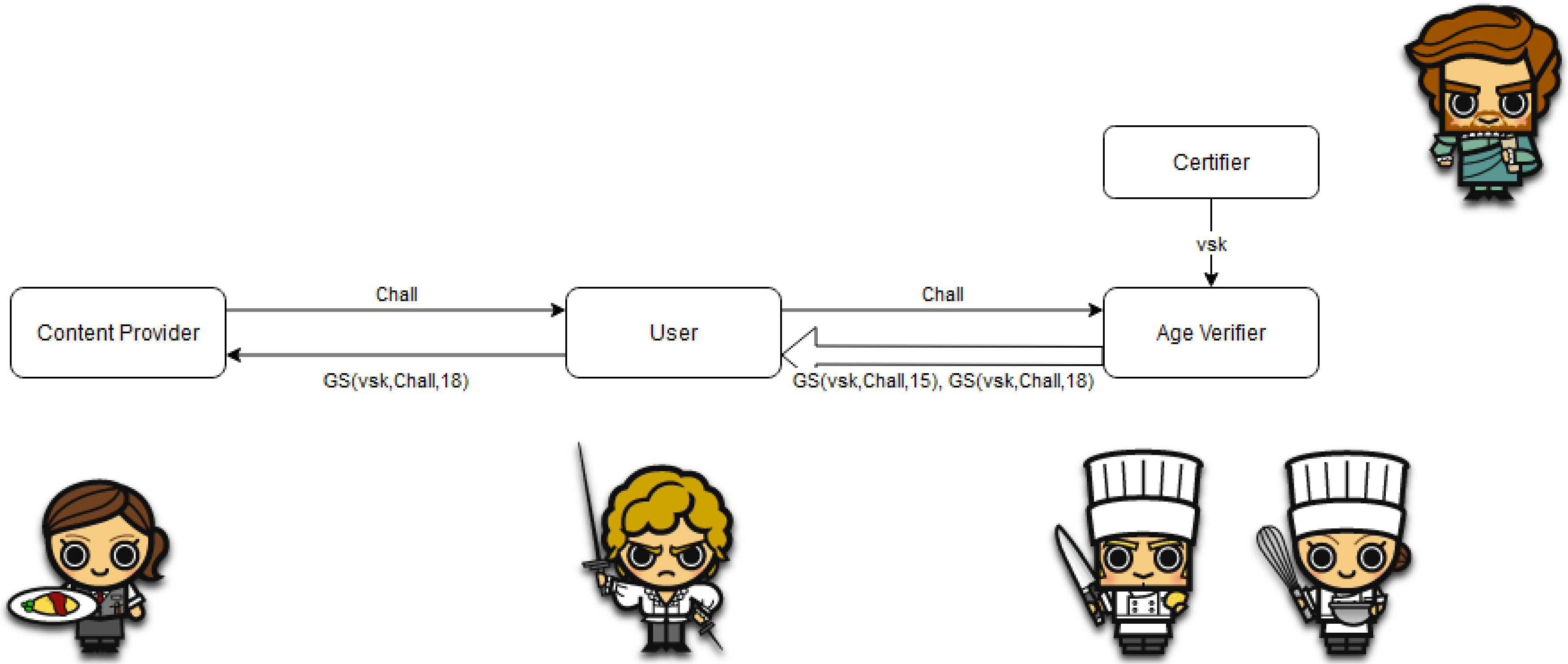
➡ Anonymity of the verifier + Unforgeability

The user forwards the signature

➡ Separation + Anonymity of the user



Workflow of the naïve prototype



Certifier

vsk

Content Provider

Chall

User

Chall

Age Verifier

GS(vsk,Chall,18)

GS(vsk,Chall,15), GS(vsk,Chall,18)

From a technical point of view

Well-known cryptography

- Zero-Knowledge Proofs of Knowledge
- Group signature

Everything can be verified by anyone

- No step relies on a trusted / uncheckable computation
- User can add enthropy when not trusting the authority

Website does not have any secret

- It can be completely hidden
- A malicious website is not stronger than an outsider

General characteristics

The API is **compatible** with every age verification system

- No technical limitation, the legislator can pick those he finds suitable

The code is open, online for nearly two years

- Public audits are good, suggestions / evolutions are welcome
- Proposing a digital common is important…

A **modular** tool

- Possibility to integrate a mechanism to bill the verification to the platforms
- Various trust level can exist for verifiers depending on the context

PETs for anonymous billing: Batch Threshold Opening

Age verifier wants to bill websites for the verification. However, a direct billing would break part of the anonymity…

A tax service could get a digest of tokens received by a website every x months, and do a threshold batch opening (like in e-voting). Without seeing individual tokens, they could say that N tokens come from A, M from B, …

Canari tokens would allow to ensure than a website is not hiding answers…

PETs for anonymous billing: Batch Threshold Opening

Real world use:

- Used in e-voting solutions (Fr, Sw, No, …)
  - Allow to compute a tally without doing a 1 by 1 opening
- Used in some cryptocurrency (Zcash)
  - Batch verification is way less expensive (in time/computation) than 1 by 1

Shamir Secret Sharing

Imagine you have a secret s, you want to share between n persons.
Pick random $z_1, \ldots, z_{n-1}$, set $z_n = s - \sum_{1}^{n-1} z_i$.

If one person is missing, s is perfectly hidden. But… if one opener is missing the vote is unusable.

Polynomial interpolation to the rescue!
Assume, you want to share the secret between n persons, and at least t of them should be present to recover s.

Pick a random polynomial P of degree t-1, such that P(0)=s.
for each user i, give them P(i).
With t values, one can interpolate and recover P(0).

PETs for a better anonymity : Blind (Group) Signature

As is, the authority learns the nonce it is signing

Problem: Even if the nonce contains no information, it creates a unique identifier if the website and the authority colludes

Blind Signatures allow an entity to sign a message without learning its content. The client would have to do a (tiny) computation locally to transform the blind signature into a proper signature before sending it to the website.

Warning: Without using a VPN, the IP would still constitute an identifier…

PETs for a better anonymity : Blind (Group) Signature

- E-voting:

  Allow to have a certified ballot…

- E-Cash:

  Withdraw cash from the bank without linking it to a spending

Blind Signature, it's just a Zero knowledge proof on a Zero Knowledge proof

(Fischlin 06)

User: Sends a commitment C of m

Server: Signs the commitment C

User: Proves he knows a Signature valid under the server verification key on a commitment of a message.

In other words:

User: <C>

Server: Q(sk,<vk,C>)

User: R(Q,C,<m>)

a Blind Group Signature… is ZK^3

You just prove that you know a message signed under a key, whose public key has been signed by another key…

P:(sk, <vk,uvk>),

Q:(usk,<P,vk,m>),

R:(m,Q,P, <vk>)

(We can do way better… but at least ZK is getting slowly standardized…)

PETs against Sub-Groups: Steppable Group Signature

As is, a user can not check that a token was generated honestly

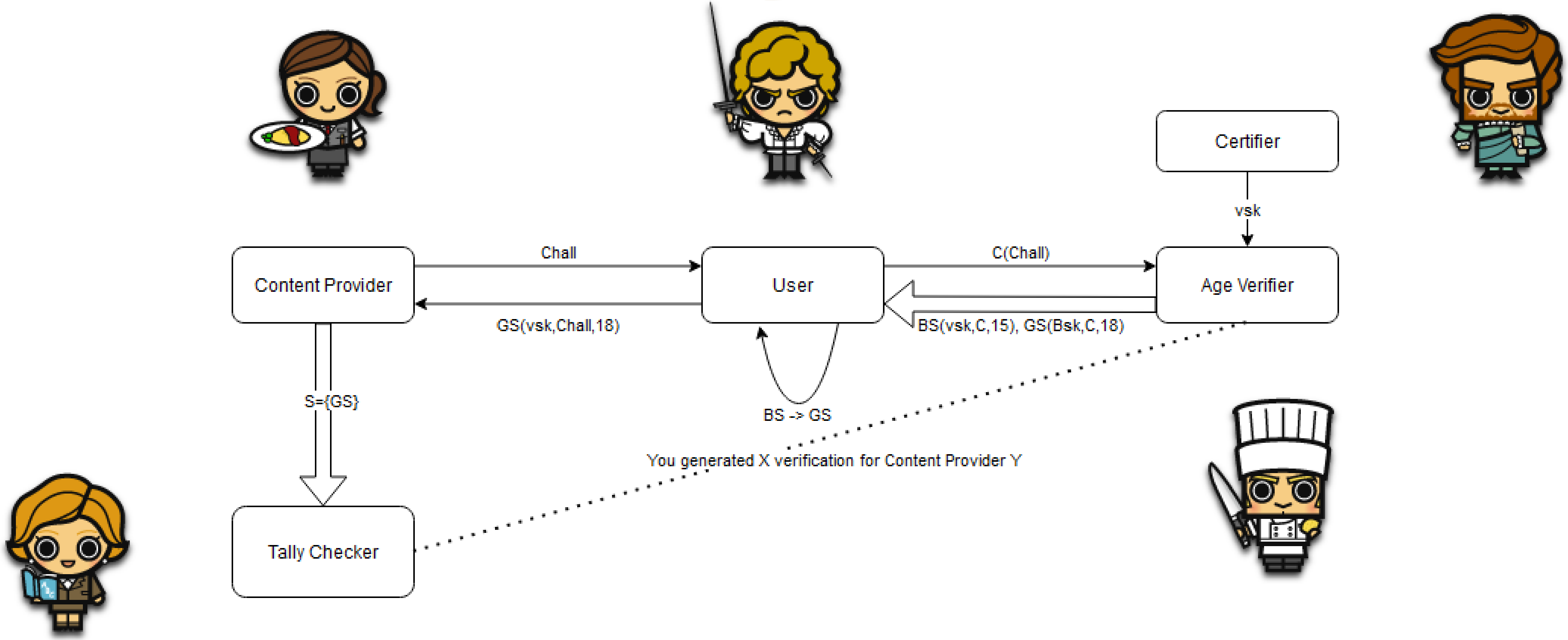
Problem: If a malicious authority has 2 certified keys, it could use one against a specific user… And so when billed would know which website Alice has consulted…

Steppable Signatures would allow an authority to generate a group signature and prove the user it used the good secret key… increasing the trust for a marginal cost.

A posteriori Group Signature: This primitive would allow to generate a pre non-anonymous signature verifiable by the user, that he would later on transform into a classical group signature

Trade-off: The first one is less costly for the user… but often implementation "forget" to do the required verification which cost 100 of millions of dollars to various cryptocurrency holder…

Everything at once



Certifier

vsk

Content Provider

Chall

User

C(Chall)

Age Verifier

GS(vsk,Chall,18)

BS(vsk,C,15), GS(Bsk,C,18)

S={GS}

BS -> GS

You generated X verification for Content Provider Y

Tally Checker

In term of crypto we're done

A combination of 3 zero-knowledge proofs leads to a more efficient, secure and privacy solution than current solutions.

Politic and public adoption make some new constraints appear.

(Academic optimal !=Adopted)

And so?

Digital Fracture?

- People accessing a website already have a computer at hand
- Careful with tourists, they don’t have a local identity…

And VPN?

- Independent from our solution… If the website site can’t know it has to apply local legislation…
- Need a global solution
- DON’T FORBID VPN!!!!!!!!!!! (!!!)

Anonymity?

- In term of GDPR, this is just Pseudonymity (IP is not hidden)
- The API does not weaken privacy compared to a classical access

Adversarial Approach

Age verification as a mean to **limit** access is **poorly** perceived by the public

- People might circumvent it directly (VPN …)
- They might **share** token for access

Critical information is **processed** to do age verification

- Very bad actors could gather it directly -> Need for **accreditations**
- Approved actors can be hacked, so data **minimization** is the safe route

For a **better** reception by the public, it should also be used to **grant perks**

- Senior discounts / benefits
- Children exclusive groups / discounts

Let’s wrap up

Age Verification has many forms so it is hard to strike the **right** balance

- **Ease of use** is very important. “On a well-known French platform”
  - 5 min delay -> only 1.7% of the (voluntary) users continuing through the process
  - 13% for CC / Facial Analysis
- A dedicated app can already be problematic
- **Balance** to be found between verification at account creation / periodical vs at every connection

It is **possible** to do a GDPR compliant solution

- A digital ID (**eIDAS** with some extra requirement) would help to have selective disclosure of attributes and so age verification.

A **global** approach, with **local** variants is possible with our design

Policies recommendation

Age Verification is **not an easy** problem to solve

Making clear recommendations is hard

It’s a **balance** between efficiency and privacy intrusion

If a solution is local, then a **VPN** will circumvent it

A gov study shows that in France **40%** of (15-24) have used one

Tech **alone** cannot be the answer (besides banning internet…) (!!!)

- **Education/Tech literacy** is important, both for the children and the parents

Enforcing existing laws is already hard, making more stringent laws without **treating** the reason why the actual ones are not applied is patching a wooden leg…

A tech academic in a political environment

CONS:

A constantly shifting specification list…

Achieving the public goal is not always what is expected…

Easier to make everybody unhappy… than satisfying anyone

PROS:

Providing a working **less terrible** solution can help improve policies

You get to meet lots of interesting people

Real impact in a very short term

Take away

- **Real World problems** that can be naively solved using basic tools.
- Some Real World data lead to innovation / new design. Nothing was really fancy, but it makes you think your security experiments differently.
- You can’t assume perfect secure channels like in an academic paper

- The lack of proper standardized Zero Knowledge (and/or Group Signature) is really an issue for political adoption. (Standards sadly matter…)

- Getting THE solution to the problem does not mean it will be adopted… Policy makers are not deterministic oracle…

- Accessibility, usability are hard to formalize but are **important**!
- Formalizing law is complicated, but useful

Merci

olivier.blazy@polytechnique.edu

https://github.com/LINCnil/SigGroup

ÉCOLE POLYTECHNIQUE

INSTITUT POLYTECHNIQUE DE PARIS