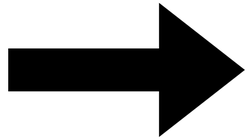
Dynamic Decentralized Functional Encryption

Jérémy Chotard, **Edouard Dufour Sans**, Romain Gay, Duong Hieu Phan, and David Pointcheval

CRYPTO 2020, Monday August 17th 2020

The technological landscape of the early 21st century

- Lots of data.
- Increasing parallel computing power.
- Investments in Machine Learning talent.



\+ Much better software products.

\- Privacy concerns.

Can we protect privacy without sacrificing the benefits of modern data science?

Isn’t that what FHE is for?

- In FHE, a client sends a ciphertext to a server.
- The server obliviously computes on the ciphertext.
- The client gets back the result.
- Multiparty extensions exist.
- But no *non-interactive* way for server to extract intelligence from multiparty data.

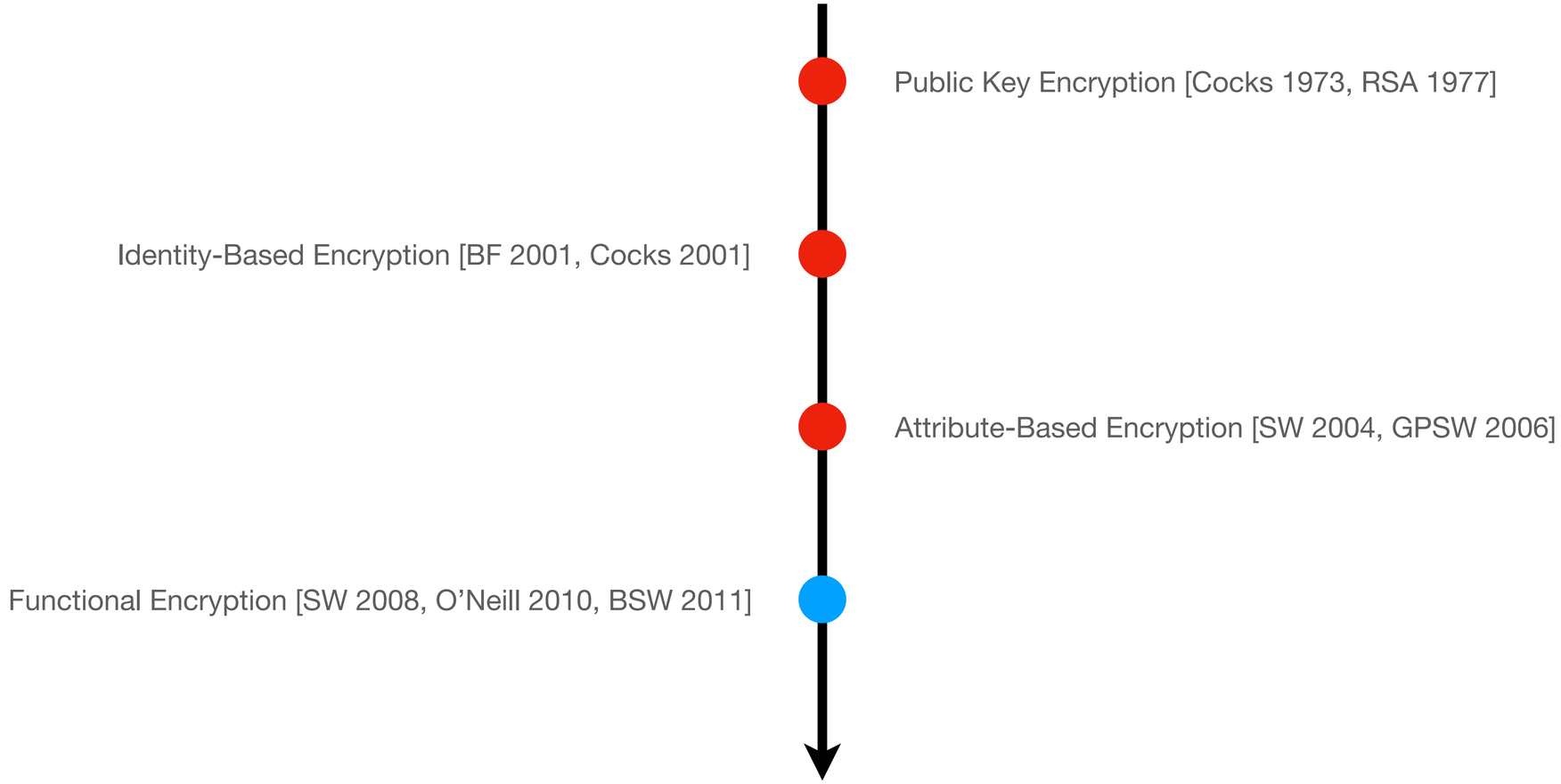
Today's Topic

Allowing a server to aggregate my data with that of other users, *non-interactively*.

The Agenda

- How does DDFE relate to FE?
- What is DDFE?
- Construction of DSum-DDFE
- Construction of AoNE-DDFE
- Construction of IP-DDFE

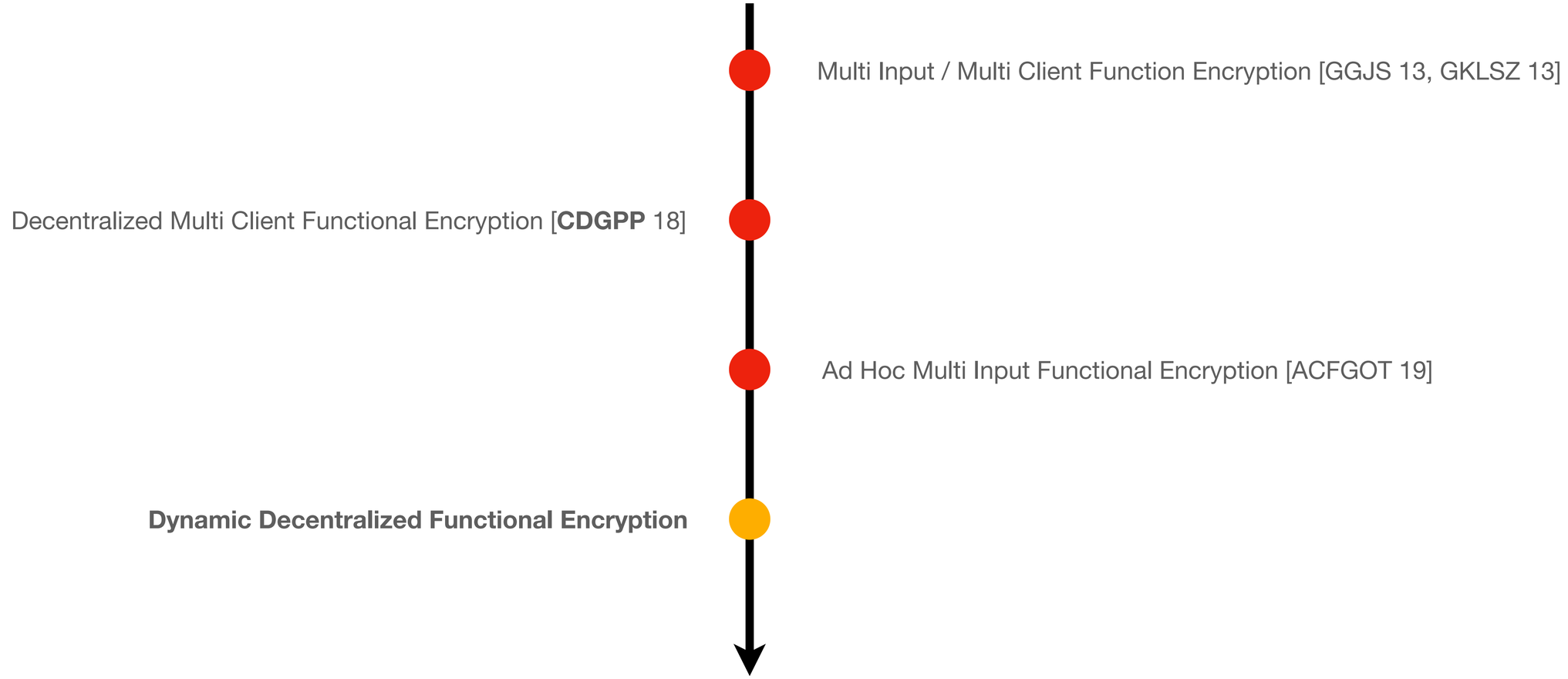
A Brief History of Functional Encryption



Functional Encryption is a framework

- PKE is not a special case of IBE.
  It is a weaker primitive.
- IBE is not a special case of ABE.
  It is a weaker primitive.
- IBE and ABE are special cases of FE.

Functional Encryption for Multiple Users

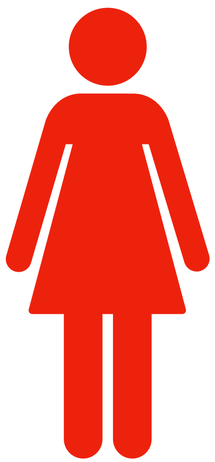


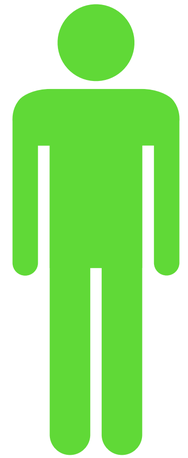
The Agenda

- How does DDFE relate to FE? ✓
- What is DDFE?
- Construction of DSum-DDFE
- Construction of AoNE-DDFE
- Construction of IP-DDFE

DDFE - Informally

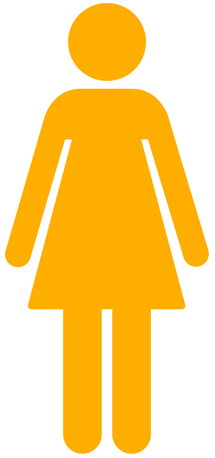
Alice





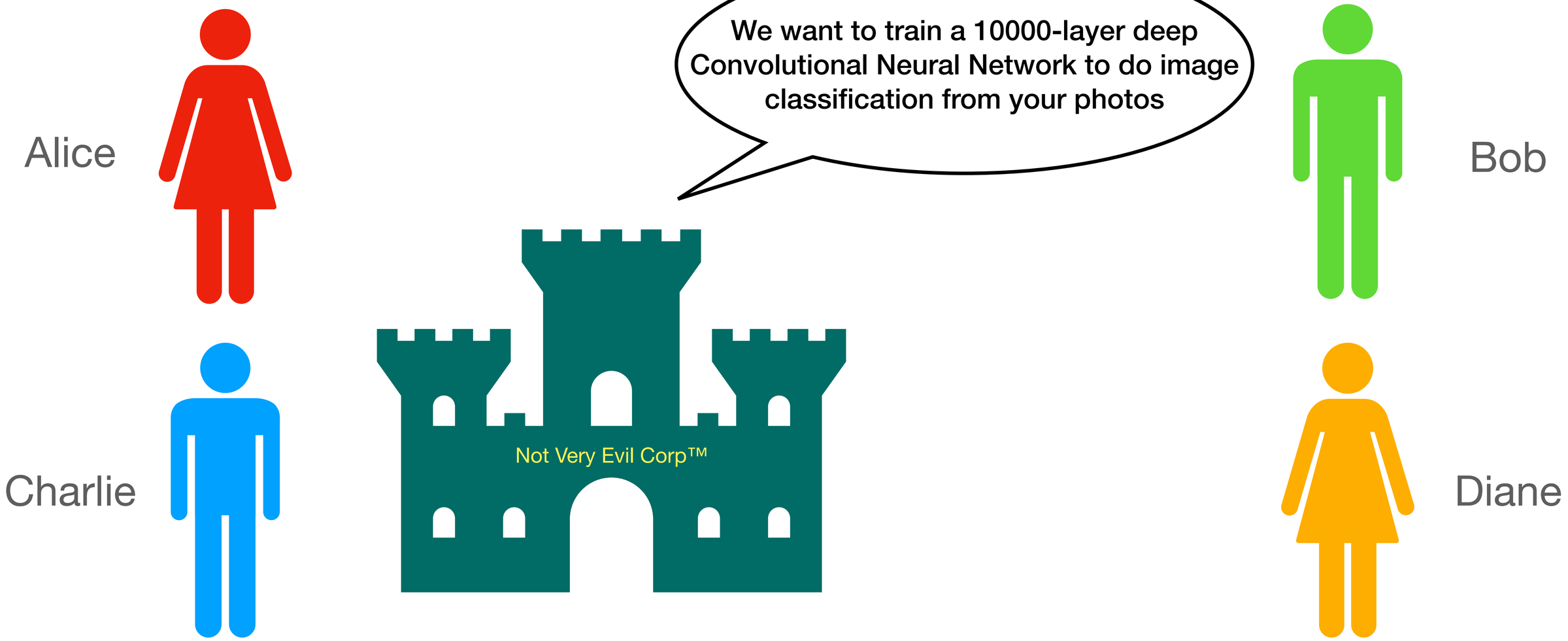
Bob

Charlie



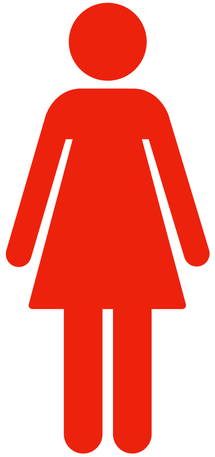
Diane

DDFE - Informally



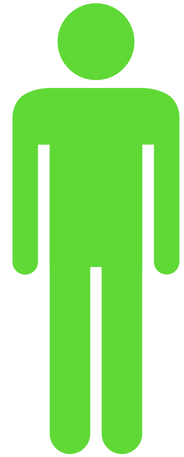
DDFE - Informally

Alice



Ok, but I care about my privacy…

Ok, but I care about my privacy…

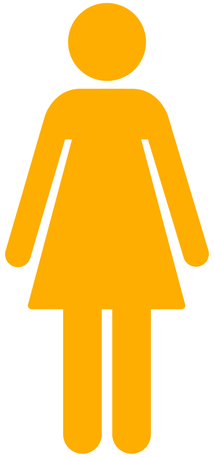


Bob

Charlie

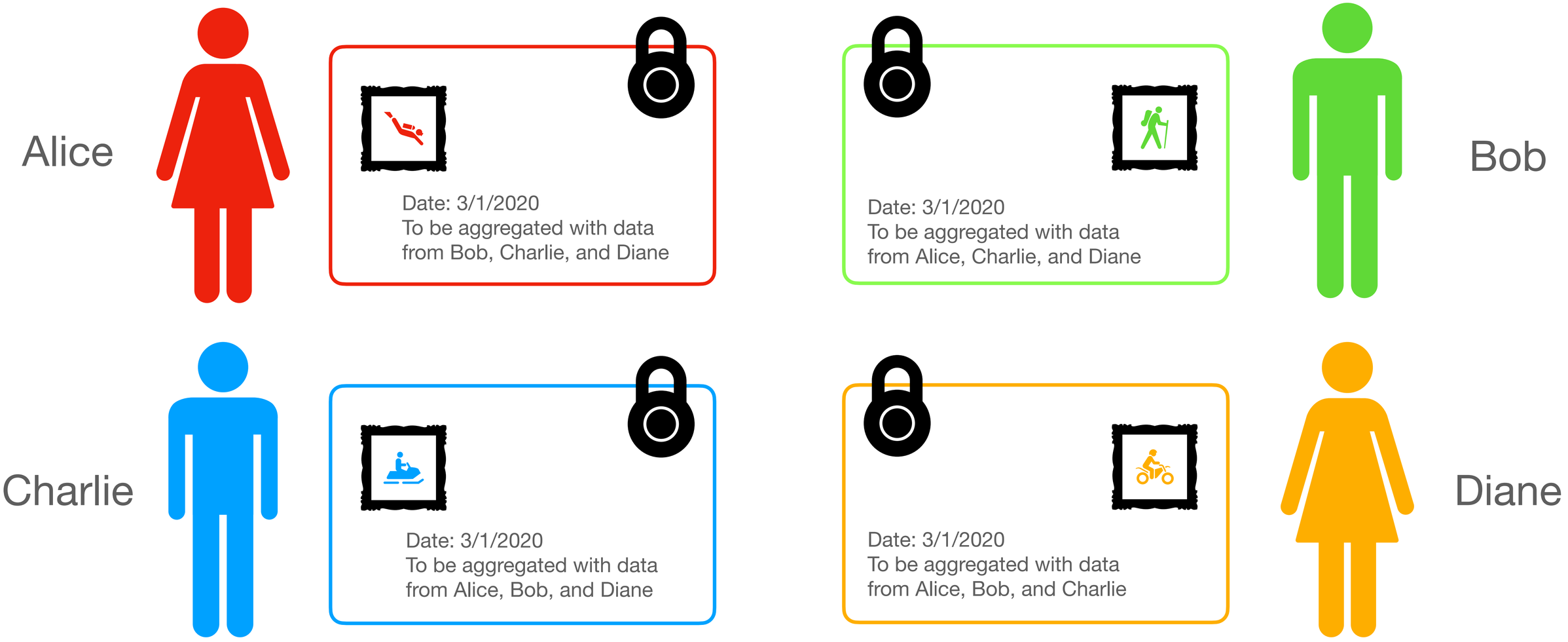
Ok, but I care about my privacy…

Ok, but I care about my privacy…

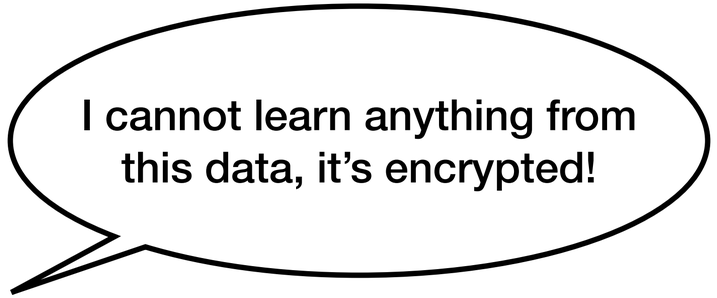


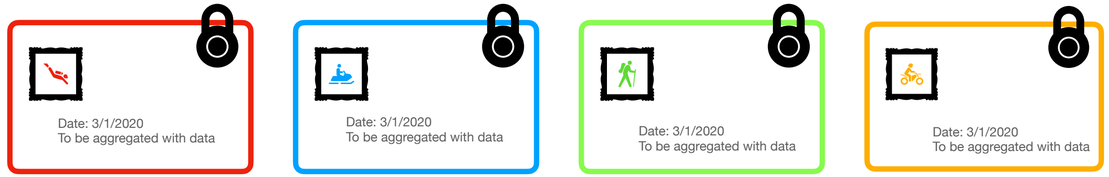
Diane

DDFE - Informally



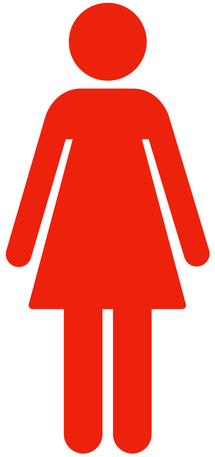
DDFE - Informally





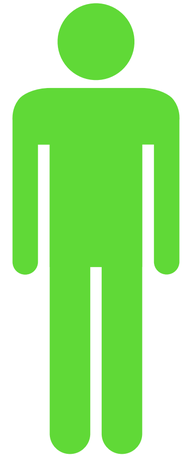
DDFE - Informally

Alice



Allow training of Neural Network on data from Me, Bob, Charlie, Diane

Allow training of Neural Network on data from Alice, Me, Charlie, Diane

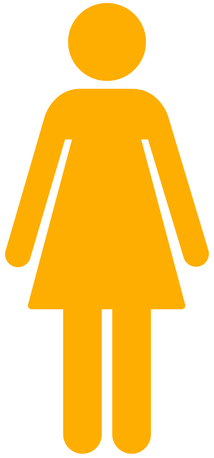


Bob

Charlie

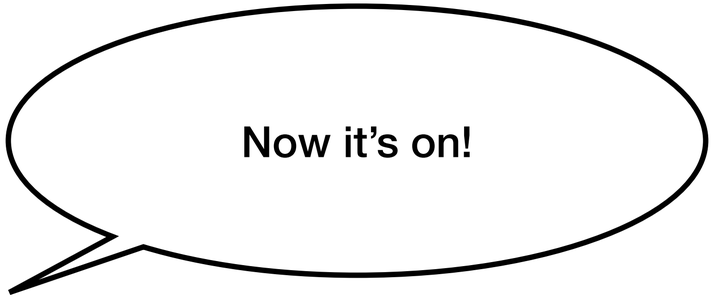
Allow training of Neural Network on data from Alice, Bob, Me, Diane

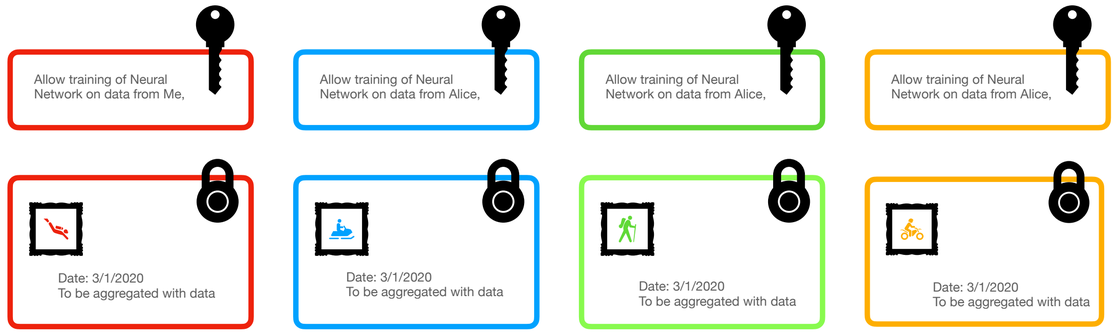
Allow training of Neural Network on data from Alice, Bob, Charlie, Me



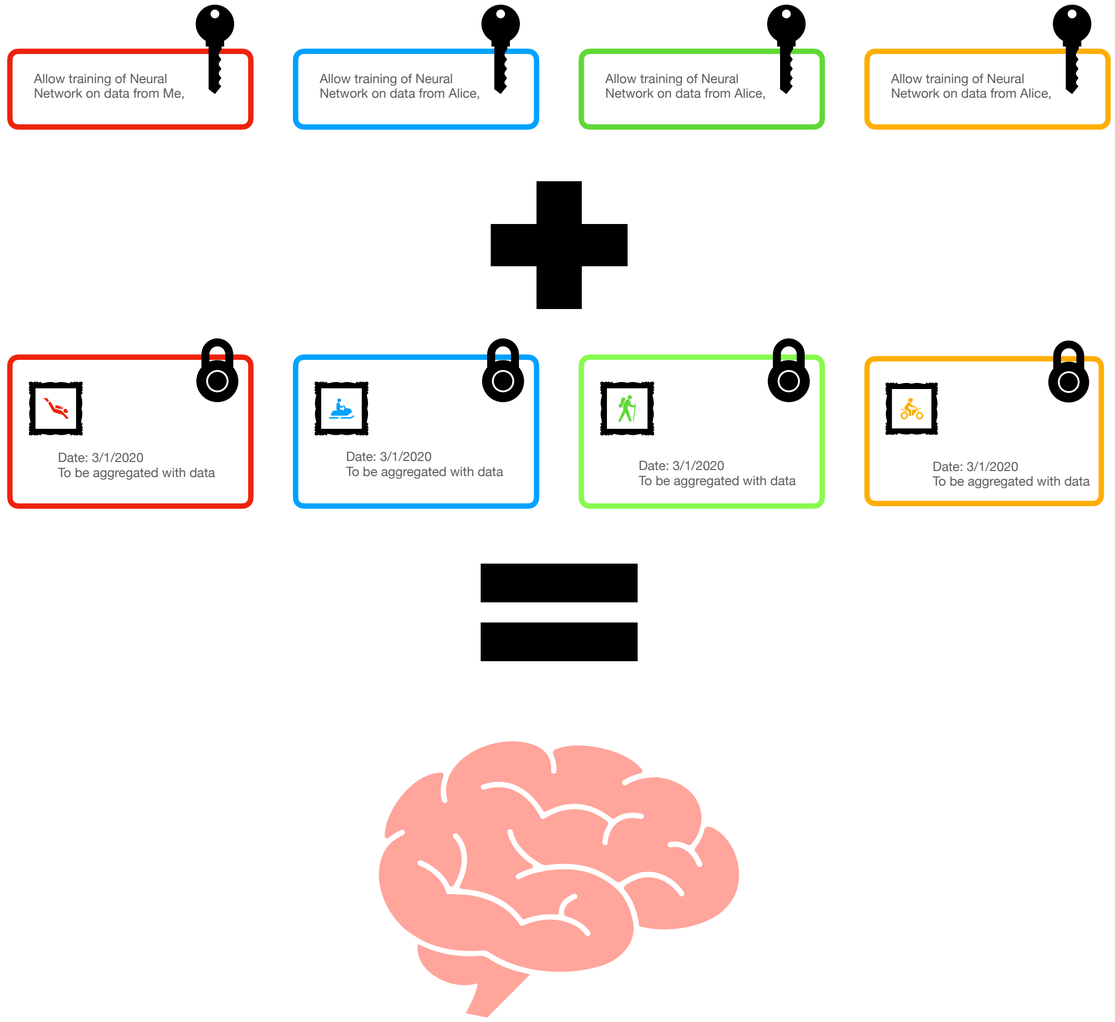
Diane

DDFE - Informally





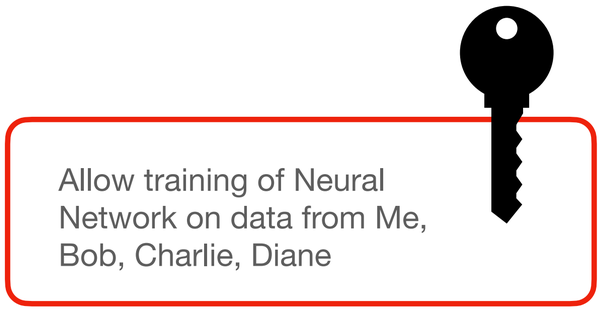
DDFE - Informally



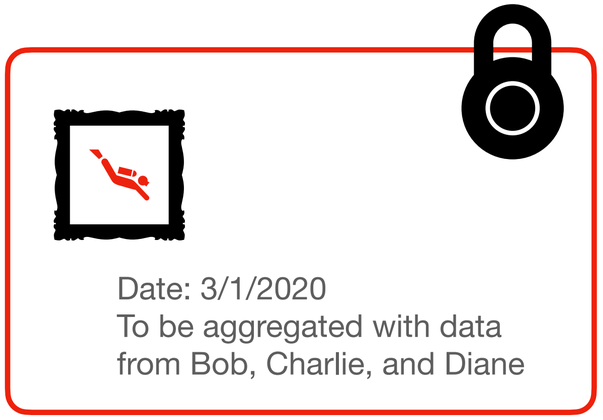
DDFE - Formally

- A Functionality
  $\mathcal{F} : \mathcal{L}(\mathcal{PK} \times \mathcal{K}) \times \mathcal{L}(\mathcal{PK} \times \mathcal{M}) \rightarrow \{0,1\}^*$
- $Setup(\lambda)$: Generate public parameters.
- $pk, sk_{pk} \leftarrow KeyGen()$: Generate my public/private key pair.
- $Encrypt(sk_{pk}, m)$: Generate a ciphertext ct_{pk}.
- $DKeyGen(sk_{pk}, k)$: Generate a functional key $dk_{pk,k}$.
- $Decrypt((dk_{pk,k_{pk}})_{pk \in \mathcal{U}_K}, (ct_{pk})_{pk \in \mathcal{U}_M})$: Evaluate $\mathcal{F}$.

DDFE - Functionality examples



- $\mathcal{K} = \mathcal{S}(\mathcal{PK}) \times \mathcal{C}$
  Set of users and a circuit



- $\mathcal{M} = \mathcal{I}mages \times \mathcal{D}ates \times \mathcal{S}(\mathcal{PK})$
  An image, a date, a set of users

DDFE - Functionality examples

$$\mathcal{F}((pk, (\mathcal{U}, \textbf{NN\_training}))_{pk\in\mathcal{U}}, (pk, (x_{pk}, Date, \mathcal{U}))_{pk\in\mathcal{U}}) = \textbf{NN\_training}((x_{pk})_{pk\in\mathcal{U}})$$

Allow training of Neural Network on data from Me, Bob, Charlie, Diane

Date: 3/1/2020
To be aggregated with data from Bob, Charlie, and Diane

- $\mathcal{U}_M = \mathcal{U}_K = \mathcal{U}$
- $Date$ is the same for all cts
- **NN_training** is the same for all keys

The Agenda

- How does DDFE relate to FE? ✓
- What is DDFE? ✓
- Construction of DSum-DDFE
- Construction of AoNE-DDFE
- Construction of IP-DDFE

DSum-DDFE: The functionality

- Sums over an Abelian Group $\mathbb{A}$.

- $\mathscr{M} = \mathbb{A} \times \mathcal{S}(\mathscr{PK}) \times \{0,1\}^*$
  A group element, a set of users, a label.

- $\mathscr{K} = \emptyset$
  No keys.

- $\mathscr{F}(\epsilon, (pk, (x_{pk}, \mathcal{U}, \ell))_{pk \in \mathcal{U}}) = \sum_{pk \in \mathcal{U}} x_{pk}$

If the user pk can compute a mask $r_{pk,\mathcal{U},\ell} \in \mathbb{A}$ such that

$$\sum_{pk' \in \mathcal{U}} r_{pk',\mathcal{U},\ell} = 0,$$

then they can just publish

$$x_{pk} + r_{pk,\mathcal{U},\ell}$$

Can we sample from

$$\left\{ (r_{pk})_{pk \in \mathcal{U}} \;\middle|\; \sum_{pk \in \mathcal{U}} r_{pk} = 0 \right\}$$

in a decentralized and non-interactive way?

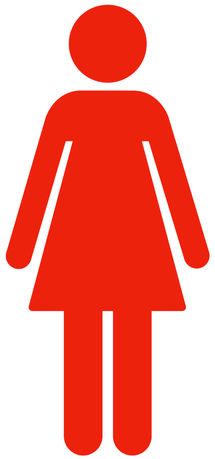
DSum-DDFE: Sum-of-PRFs [Waters in CC09]

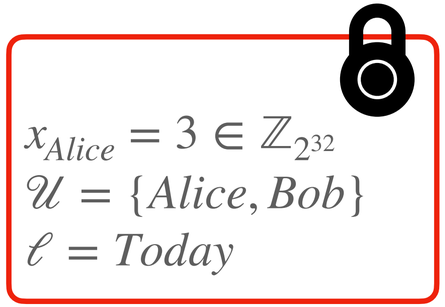
- Computational solution.
- Compute shared randomnesses $K_{pk,pk'}$ via DH.
- Compute $r_{pk,\mathcal{U},\ell}$ as

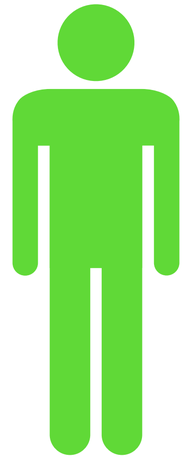
$$\sum_{\substack{pk' \in \mathcal{U} \\ pk' < pk}} F_{K_{pk,pk'}}(\ell) - \sum_{\substack{pk' \in \mathcal{U} \\ pk < pk'}} F_{K_{pk,pk'}}(\ell)$$

DSum-DDFE: Technical Difficulties

Alice





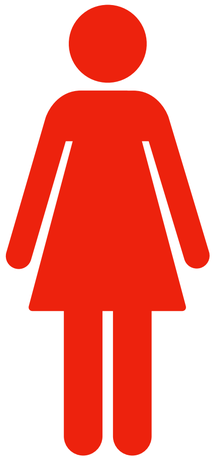


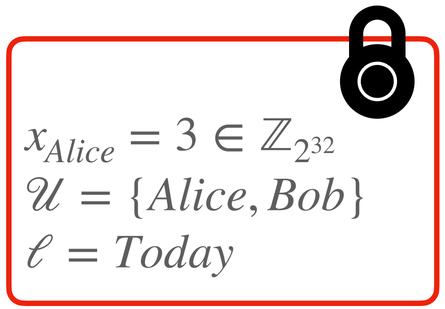
Bob

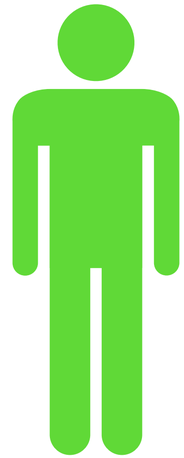
Charlie

DSum-DDFE: Technical Difficulties

Alice

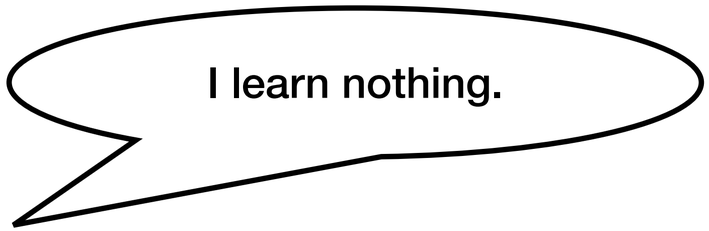






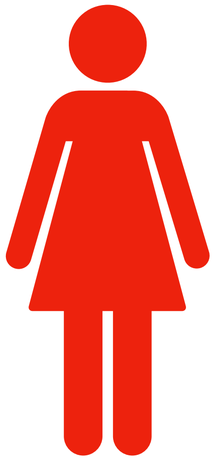
Bob

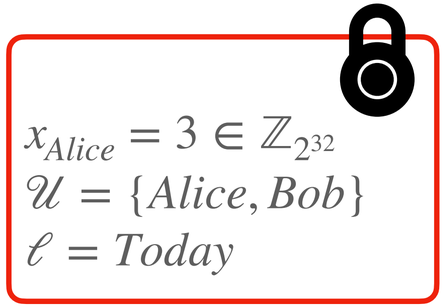
Charlie

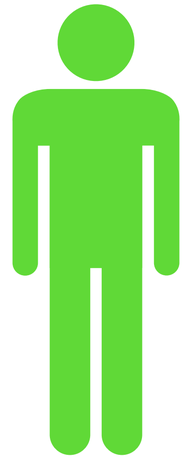


DSum-DDFE: Technical Difficulties

Alice





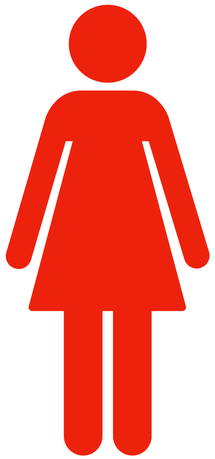


Bob

Charlie

DSum-DDFE: Technical Difficulties

Alice



$x_{Alice} = 3 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$

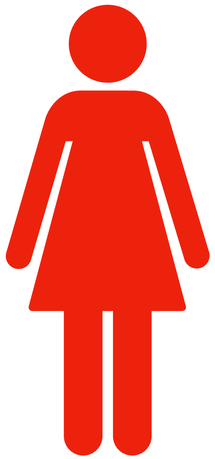
$x_{Bob} = 5 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$

Bob

Charlie

DSum-DDFE: Technical Difficulties

Alice



$x_{Alice} = 3 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$

$x_{Bob} = 5 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$

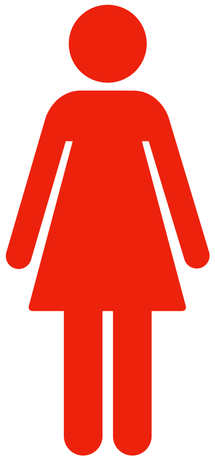
Bob

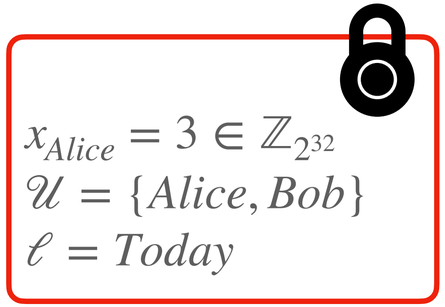
Charlie

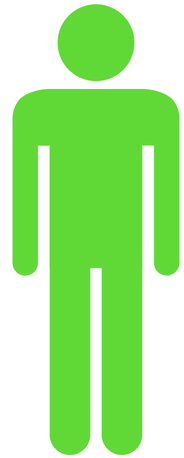
I learn that
$x_{Alice} + x_{Bob} = 8.$

DSum-DDFE: Technical Difficulties

Alice





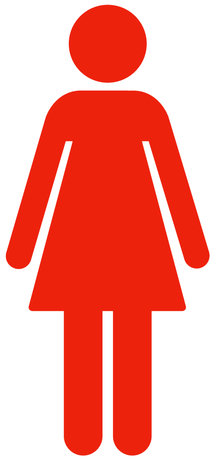


Bob

Charlie

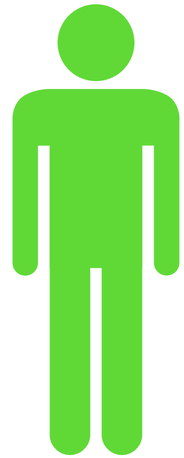
DSum-DDFE: Technical Difficulties

Alice



$x_{Alice} = 3 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$

$x'_{Alice} = 20 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$

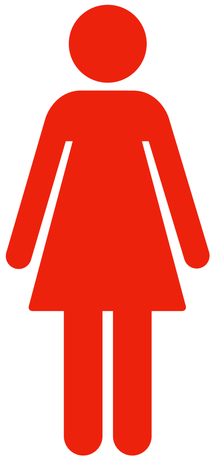


Bob

Charlie

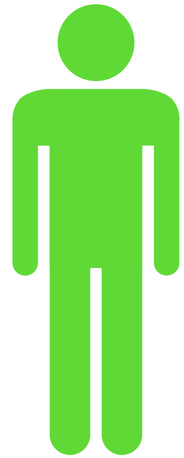
DSum-DDFE: Technical Difficulties

Alice



$x_{Alice} = 3 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$

$x'_{Alice} = 20 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$



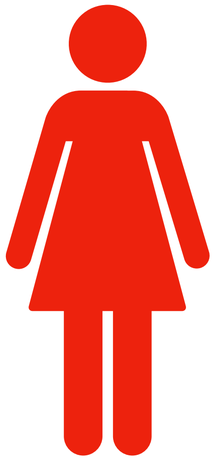
Bob

Charlie

I should learn nothing.

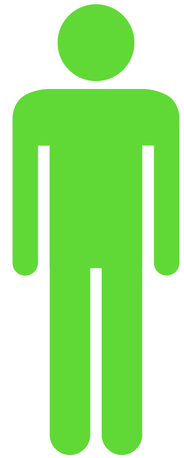
DSum-DDFE: Technical Difficulties

Alice



$x_{Alice} = 3 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$

$x'_{Alice} = 20 \in \mathbb{Z}_{2^{32}}$
$\mathcal{U} = \{Alice, Bob\}$
$\ell = Today$



Bob

Charlie

I learn that

$$x'_{Alice} + r_{Alice,\mathcal{U},\ell} - x_{Alice} - r_{Alice,\mathcal{U},\ell} = x'_{Alice} - x_{Alice} = 17$$

All-or-Nothing Encapsulation: The functionality

- $\mathscr{M} = \{0,1\}^L \times \mathcal{S}(\mathscr{PK}) \times \{0,1\}^*$
  L bits of data, a set of users, a label.

- $\mathscr{K} = \emptyset$
  No keys.

- $\mathscr{F}(\epsilon, (pk, (x_{pk}, \mathcal{U}, \ell))_{pk \in \mathcal{U}}) = (pk, x_{pk})_{pk}$

All-or-Nothing Encapsulation solves the problem of an adversary abusing linear structure without getting enough ciphertexts for the Finalize condition to kick in.

All-or-Nothing Encapsulation from IBE

$|\mathcal{U}|$ layers of IBE encryption on identity ℓ + my key for identity ℓ

All-or-Nothing Encapsulation from [BF01] has succinct ciphertexts [Paper]

The Agenda

- How does DDFE relate to FE? ✓
- What is DDFE? ✓
- Construction of DSum-DDFE ✓
- Construction of AoNE-DDFE ✓
- Construction of IP-DDFE

Inner Product DDFE: The functionality

- Inner Products over $\mathbb{Z}_p$.

- $\mathscr{M} = \mathbb{Z}_p \times \mathcal{S}(\mathcal{PK}) \times \{0,1\}^*$
  A scalar, a set of users, a label.

- $\mathscr{K} = \{(pk, y_{pk})_{pk \in \mathcal{U}} \,|\, \mathcal{U} \in \mathcal{S}(\mathcal{PK})\}$
  Weights over a set of users

- $\mathscr{F}((pk, (pk', y_{pk'})_{pk' \in \mathcal{U}})_{pk \in \mathcal{U}}, (pk, (x_{pk}, \mathcal{U}, \ell))_{pk \in \mathcal{U}}) = \sum_{pk \in \mathcal{U}} x_{pk} y_{pk}$

Inner Product MCFE: Basic idea [CDGPP 18]

- $KeyGen()$: secret key $s \leftarrow \mathbb{Z}_p$
- $Encrypt(s, (x, \mathcal{U}, \ell))$: $g^x \cdot \mathscr{H}(\mathcal{U} \,||\, \ell)^s$
- $DKeyGen((s_{pk})_{pk\in\mathcal{U}}, (y_{pk'}, pk')_{pk'\in\mathcal{U}})$: $\sum_{pk\in\mathcal{U}} s_{pk} y_{pk}$
- $Decrypt(dk, (pk, c_{pk})_{pk\in\mathcal{U}})$:

$$\prod_{pk\in\mathcal{U}} c_{pk}^{y_{pk}} \Big/ \mathscr{H}(\ell)^{dk} = \prod_{pk\in\mathcal{U}} \left(g^{x_{pk}} \cdot \mathscr{H}(\mathcal{U} \,||\, \ell)\right)^{y_{pk}} \Big/ \mathscr{H}(\ell)^{\sum_{pk\in\mathcal{U}} s_{pk} y_{pk}}$$

$$= g^{\sum_{pk\in\mathcal{U}} x_{pk} y_{pk}}$$

How do we distribute key generation?

How do we distribute key generation?

The key is a sum of the $y_{pk}s_{pk}$, just use DSum!

How do we protect against repeated queries?

How do we protect against repeated queries?

Same as DSum, with AoNE!

Going from scalar messages $\mathbb{Z}_p$ to vector messages $\mathbb{Z}_p^d$ requires IPFE and another use of AoNE [Paper].

Recap: Our contributions