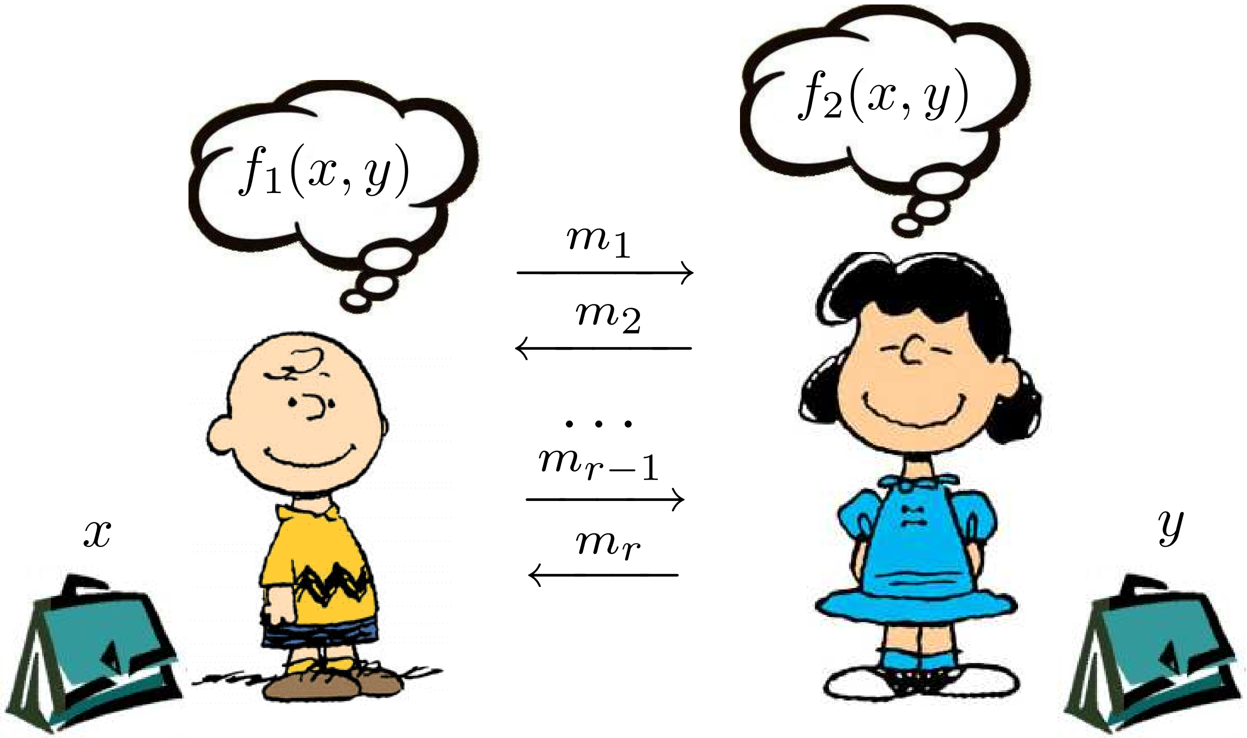
Additive Conditional Disclosure of Secrets

Sven Laur
swen@math.ut.ee

Helsinki University of Technology

Motivation

Consider standard two-party computation protocol.



Standard goals of secure two-party computation

- The inputs and outputs should remain private:
  - Charlie should learn nothing except x and $f_1(x, y)$.
  - Lucy should learn nothing except y and $f_2(x, y)$.

- The outputs should be correct:
  - Charlie should really obtain $f_1(x, y)$.
  - Lucy should really obtain $f_2(x, y)$.

- The protocol should be fair:
  - Charlie and Lucy should <u>both</u> obtain outputs or <u>none</u> of them.

Secure evaluation of intersection cardinality

Charlie		Lucy
Characteristic vector $x = (x_1, x_2, \dots, x_n)$.		Characteristic vector $y = (y_1, y_2, \dots, y_n)$.
Compute $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$.	$\xrightarrow{\mathsf{pk}}$	Store the public key pk.
Form a vector $c = (\mathsf{E}(x_1), \mathsf{E}(x_2), \dots, \mathsf{E}(x_n))$.	$\xrightarrow{c}$	Compute answer $d = c_1^{y_1} c_2^{y_2} \cdots c_n^{y_n} \mathsf{E}(0) = E(x_1 y_1 + x_2 y_2 \cdots + x_n y_n)$.
	$\xleftarrow{d}$	
Output $\mathsf{Dec}(d) = \lvert X \cap Y \rvert$		Output $\perp$

What if Charlie is malicious?

If Charlie sends invalid vector

$$c = (\mathsf{E}(1), \mathsf{E}(2), \mathsf{E}(4), \ldots \mathsf{E}(2^n)),$$

then the return value

$$d = \mathsf{E}(1y_1 + 2y_2 + 4y_3 + \cdots + 2^n y_n)$$

and Charlie can reveal

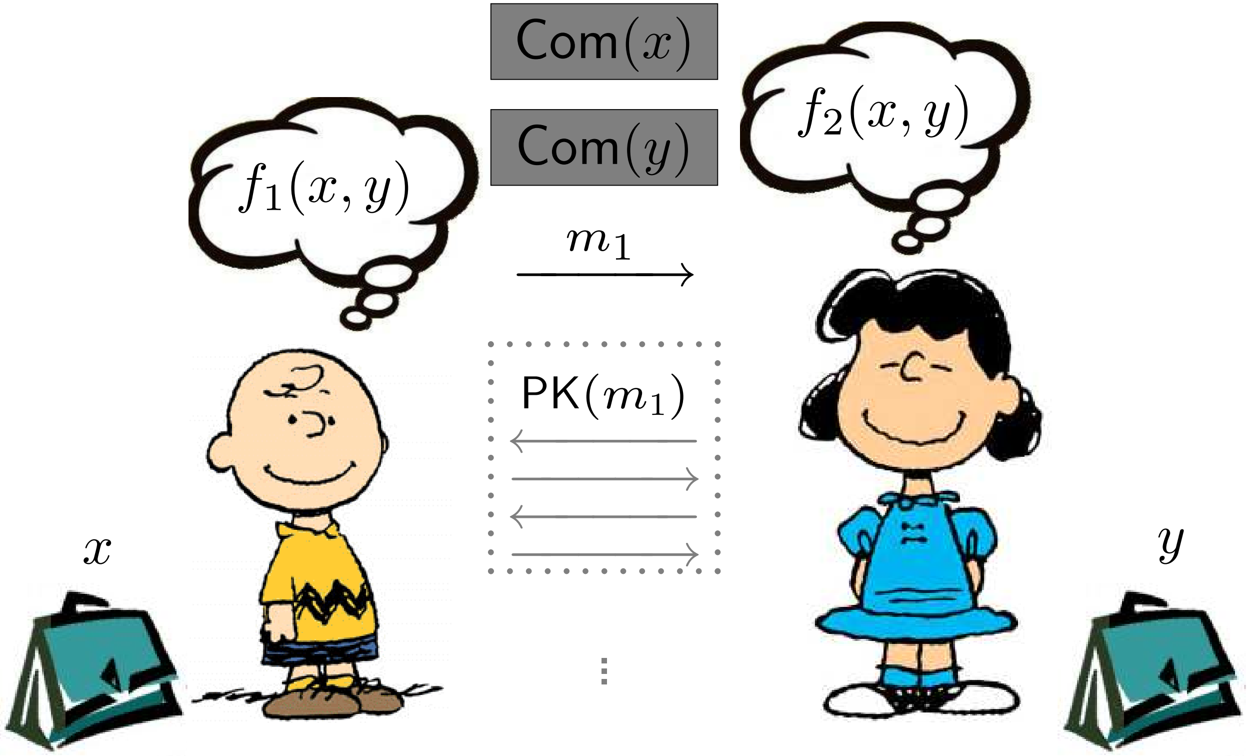
$$\mathsf{Dec}(d) = y_n \ldots y_2 y_1 = y.$$

Standard way to achieve privacy and correctness

1. Device a protocol Π that is secure in *semihonest model:*
   + Both parties follow the protocol,
   – but try to extract additional information

2. Extend the protocol Π by forcing semihonest behaviour:
   + Both parties commit their inputs x and y.
   + For each message m_i of the protocol Π the sender adds a zero-knowledge proof $\mathsf{PK}(m_i)$ that m_i was correctly formed.

Extended protocol

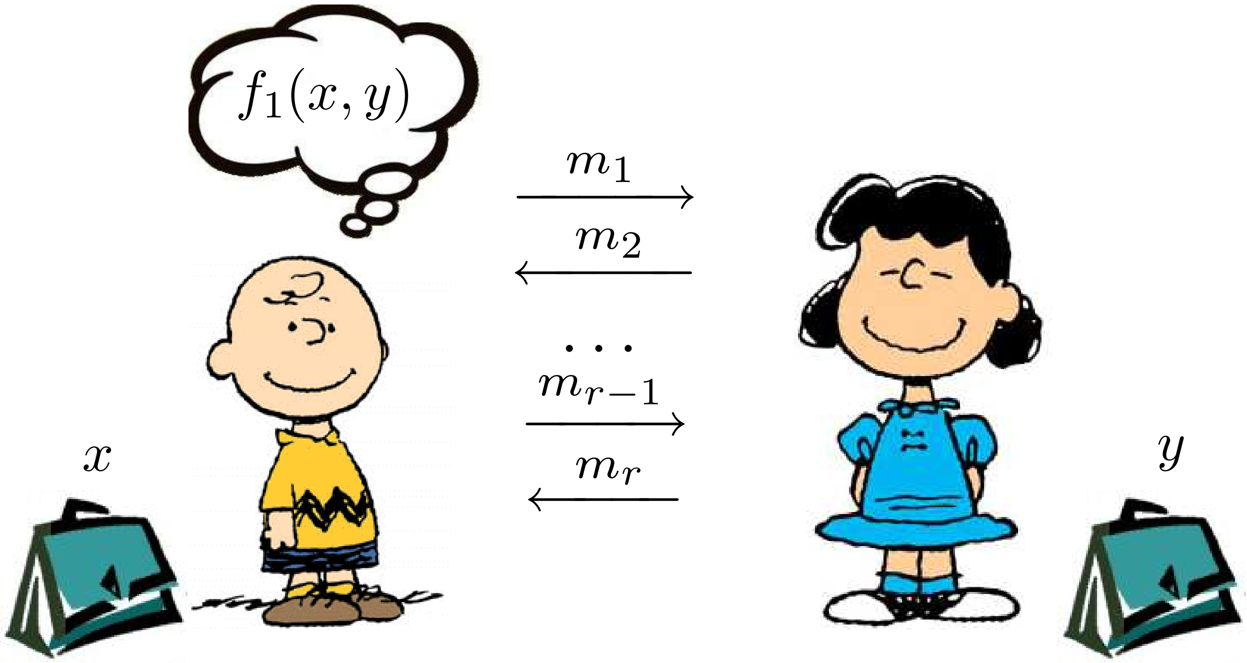


Some properties of extended protocols

- Standard zero-knowledge proofs have at least four rounds:
  - The extended protocol has a large communicational overhead.
  - The extended protocol has a large overhead in rounds.

- We can use non-interactive zero-knowledge proofs (NIZK):
  + Proofs will be relatively short binary strings.
  + The number of rounds do not increase.
  - The security properties of NIZK are essentially unknown.
  - All proofs are valid in the random oracle model.
  - All proofs are valid in the common reference string model.

What if correctness is infeasible?



When correctness requirement is questionable?

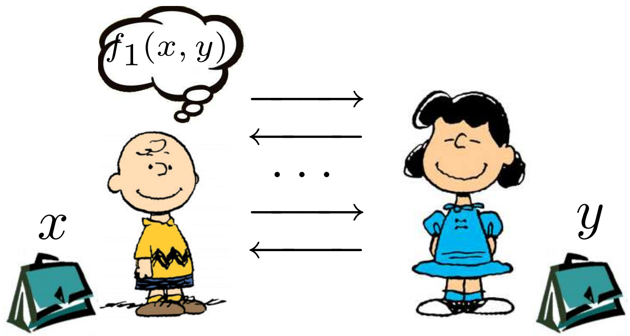
- Lucy's input might be so large that ZK proofs are huge.
- Charlie computes a predicate $P(x, y)$ and there are wild cards

$$\exists y_0 : \quad \forall x \; P(x, y_0) = 0$$
$$\exists y_1 : \quad \forall x \; P(x, y_1) = 1.$$

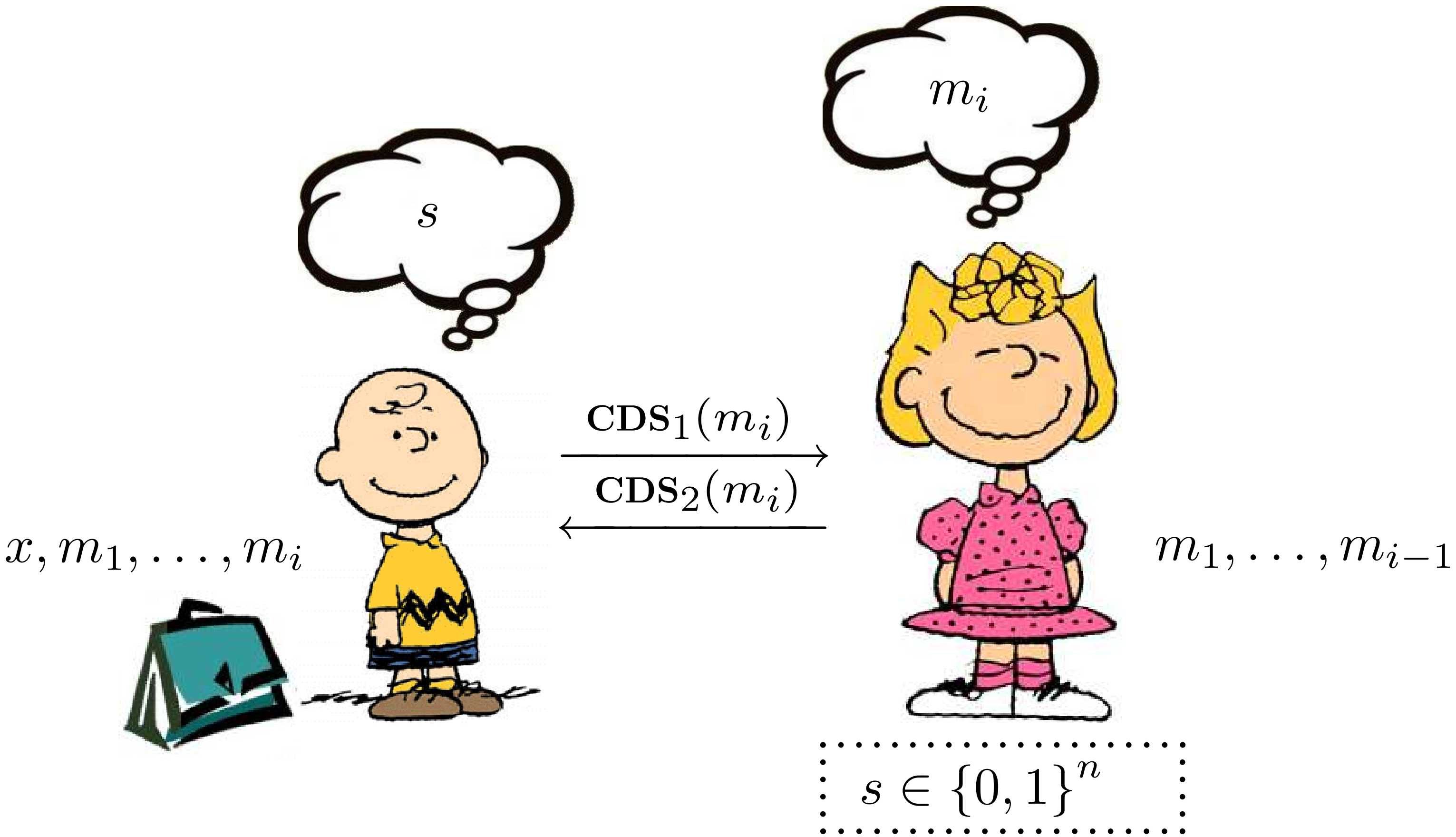
- External reasons force Lucy to act in a semihonest way, for example
  - commercial reputation,
  - laws forced by government organisations.

Informal definition of privacy

- Charlie should learn $f_1(x, y)$ only if
  + input x is in the valid range $\mathcal{X}$;
  + all messages m_i follow protocol specification.
- Charlie should learn nothing if $x \notin \mathcal{X}$ or some m_i is malformed.
- Lucy should learn $f_2(x, y) = \perp$, i.e. nothing.

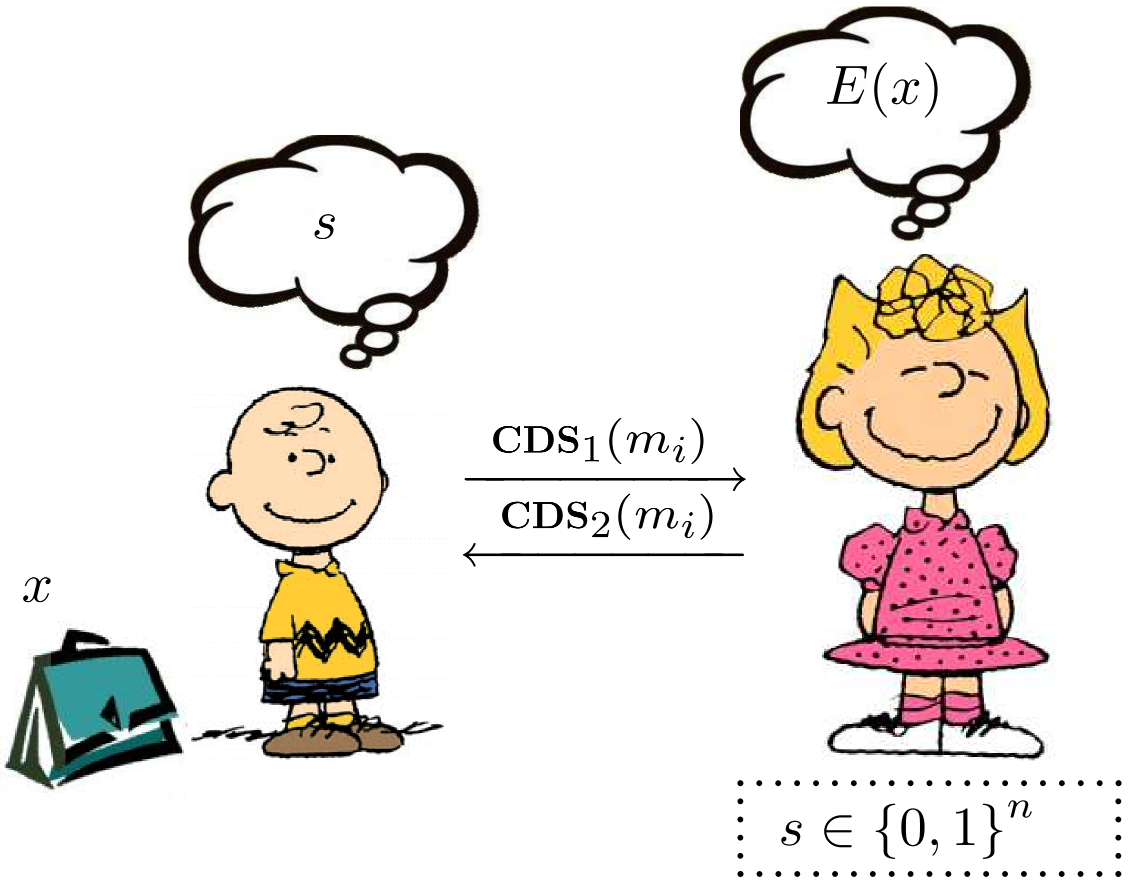


Binding conditional disclosure of secrets (CDS)



Charlie learns secret s only if the message m_i is formed correctly.

Additive conditional disclosure of secrets (ACDS)



Charlie learns secret s only if the input x is in valid set $\mathcal{X}$.

ACDS from oblivious transfer

Consider a keyed list access

Charlie | Sally

$y_i \longrightarrow$

y_1	s
y_2	s
$\vdots$	$\vdots$
y_i	s
$\vdots$	$\vdots$
y_k	s

Charlie invokes oblivious transfer protocol to retrieve:

- $L[y_i] = s$ if $y_i \in \mathcal{X}$,
- $L[y_i] = \perp$ if $y_i \notin \mathcal{X}$.

Simple ACDS protocol

Charlie		Sally
Input x.		Secret s and set of valid values $\mathcal{X} = \{y_1, \ldots, y_k\}$.
Compute $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$.	$\xrightarrow{\mathsf{pk}}$	Store the public key pk.
Send a query $c = \mathsf{E}(x)$	$\xrightarrow{c}$	Compute answers $d_i = (c \cdot \mathsf{E}(-y_i))^{t_i} \cdot \mathsf{E}(s)$
	$\xleftarrow{d_1,\ldots,d_k}$	$= \mathsf{E}(t_i(x - y_i) + s)$
For $x = y_{i_0}$ output $\mathsf{Dec}(d_{i_0}) = s$		Output $\mathsf{E}(x)$

Spectacular failure of homomorphic OT

The message space of Pallier encryption scheme is $\mathbb{Z}_{p \cdot q}$ for primes $p, q \in \mathbb{P}$.

If Charlie sends $E(x)$ such that

$$x \equiv y_1 \mod p \qquad \text{and} \qquad x \equiv y_2 \mod q$$

then

$$\begin{aligned} \mathsf{Dec}(d_1) &\equiv t_1(x - y_1) + s \mod pq &\Rightarrow\qquad \mathsf{Dec}(d_1) &\equiv s \mod p \\ \mathsf{Dec}(d_2) &\equiv t_1(x - y_2) + s \mod pq &\Rightarrow\qquad \mathsf{Dec}(d_2) &\equiv s \mod q \end{aligned}$$

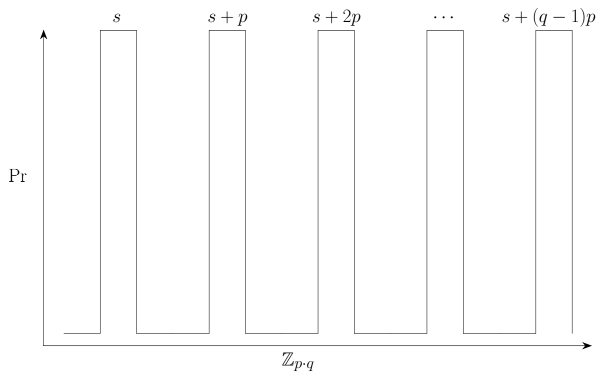
and Charlie can restore secret even if $x \notin \mathcal{X}$.

What is wrong here!?

- If $\gcd(x - y_i, pq) = 1$ then every thing is OK

$$\Pr\left[\mathsf{Dec}(d_i) = t_i(x - y_i) + s = u\right] = \frac{1}{pq}.$$

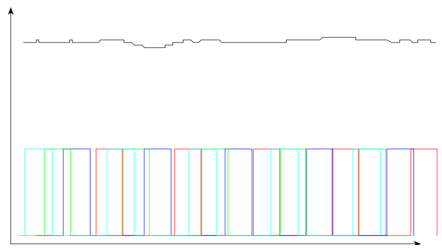
- Otherwise we have a distribution with large steps.



Information-theoretical solution

We choose many different shifts Δ for a single s and send $s + \Delta$ instead.

- Then large bumps cancel out.



- If Δ is such a set that the distribution $\Delta \mod p$ and $\Delta \mod q$ is close to uniform, then

$$t_i(x - y_i) + s + \Delta, \qquad t_i \in \mathbb{Z}_{p \cdot q} \quad x \neq y_i$$

is close to uniform.

Precise construction

- We choose ℓ such that $\frac{m2^\ell}{2\min\{p,q\}} \leq 2^{-\lambda}$, where $k = |\mathcal{X}|$ and $2^{-\lambda}$ is desired security level.
- The message space reduces $s \in \{0,1\}^\ell$.
- The random shifts are

$$\Delta = \left\{0, 2^\ell, 2 \cdot 2^\ell, 3 \cdot 2^\ell, \ldots r \cdot 2^\ell\right\}, \qquad r \cdot 2^\ell < pq < (r+1)2^\ell.$$

- Charlie can restore

$$s \equiv (\mathsf{Dec}(d_{i_0}) \mod pq) \mod 2^\ell \equiv s + \Delta \mod 2^\ell \equiv s \mod 2^\ell.$$

Computationally secure solution

Information theoretical solution has a low throughput.

– We can use roughly 25%–40% of the message space size for the standard Pallier encryption scheme with 512 bit primes.

If we require only computational privacy we can do significantly better.

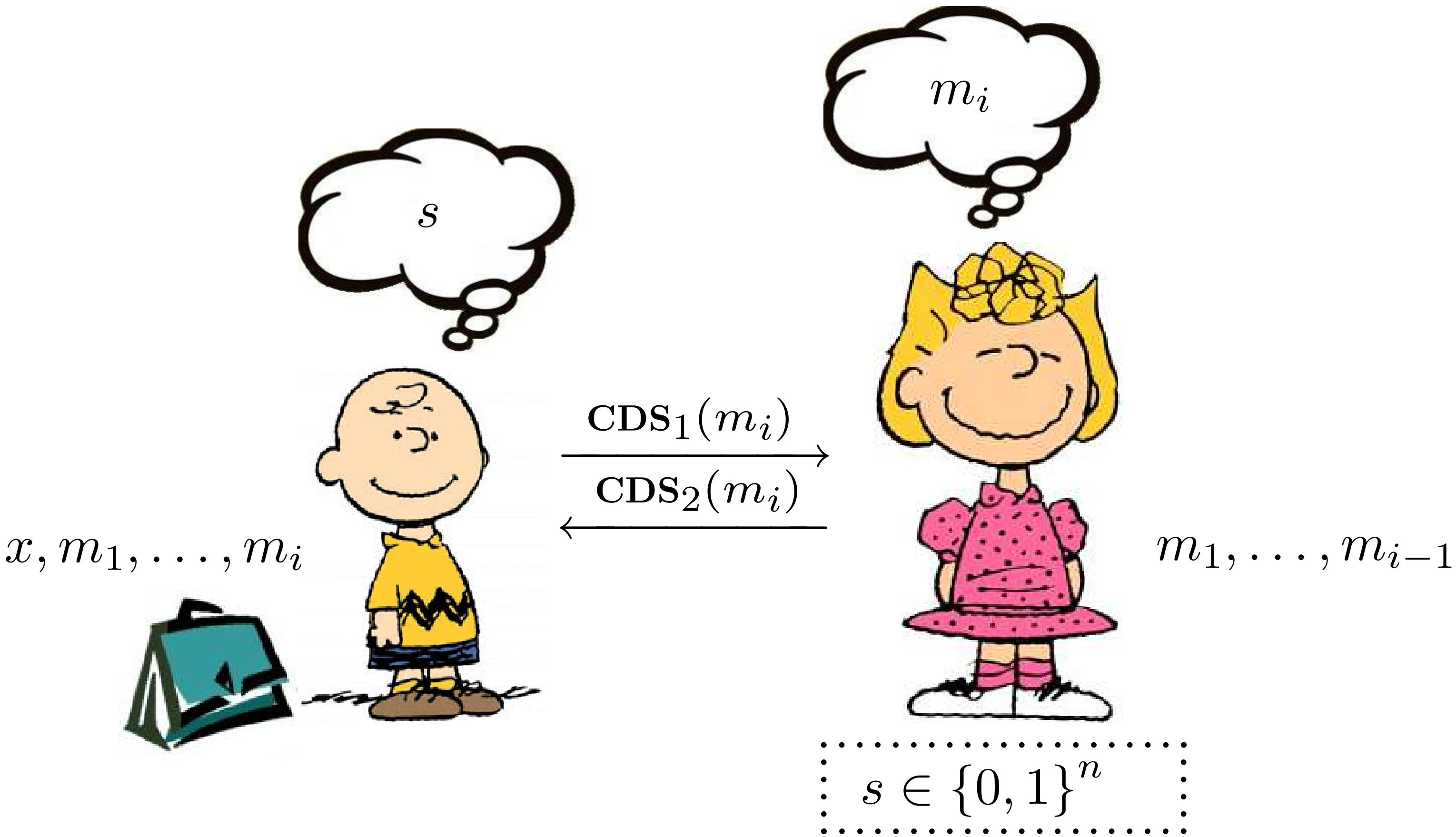
– Trivial solution

$$\mathsf{E}\left(\boxed{\text{IT encoded key } k}\right) \quad \text{and} \quad \boxed{\mathsf{SymEnc}_k(s)}$$

– Can compress it all into a single encryption?

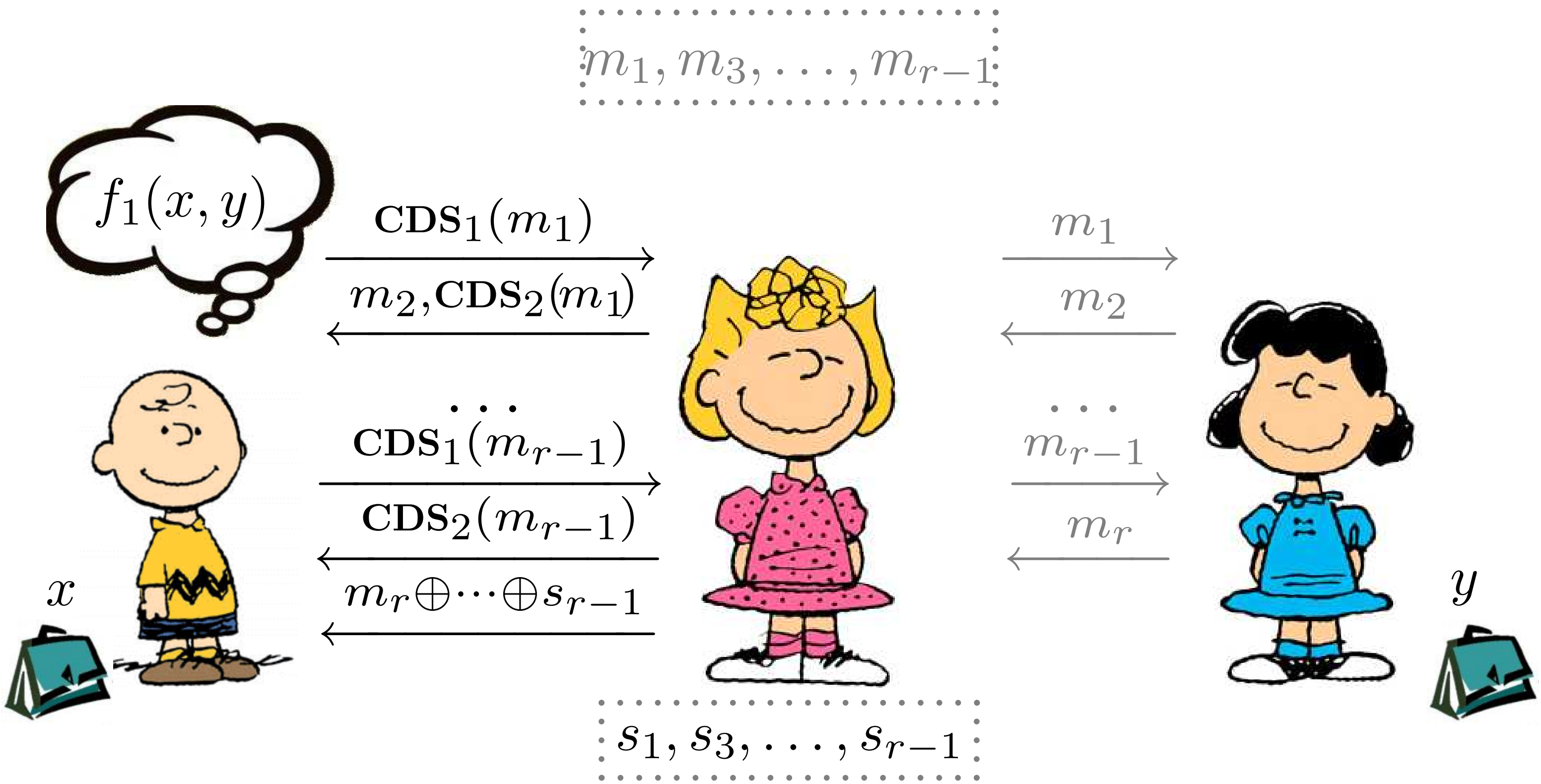
Cleverly encoded 128 bit key k	$\mathsf{SymEnc}_k(s)$

Now recall the idea of CDS



Charlie learns secret s only if the message m_i is formed correctly.

Privacy through binding CDS



Formal specification

In the semihonest protocol Π Charlie sends messages $m_1, m_3, \ldots, m_{r-1}$.

Secure transformation

- For each odd message m_i Charlie and Lucy execute a binding CDS scheme such that
  - Charlie obtains a secret s_i iff m_i is valid;
  - Lucy can compute message m_i from protocol transcript.
- Lucy uses restored m_i and follows the original protocol Π.
- Lucy sends $m_r \oplus s_1 \oplus \cdots \oplus s_{r-1}$ as last message.
- Charlie can restore m_r iff $m_1, m_3, \ldots, m_{r-1}$ were correctly formed.

Alternative viewpoint to padding schemes in ACDS

- We used special kind of padding scheme to prevent malicious behaviour.
- Plaintext awareness transformations use also padding that fix a very restricted input format.
- Actually, the constructed padding schemes achieve plain-text awareness under very restricted conditions. Adversary is allowed to:
  - do homomorphic operations;
  - choose a random cryptogram;
  - choose a random cryptogram of p;
  - choose a random cryptogram of q;