Privacy-Preserving Outsourcing by Distributed Verifiable Computation

Meilof Veeningen

Philips Research

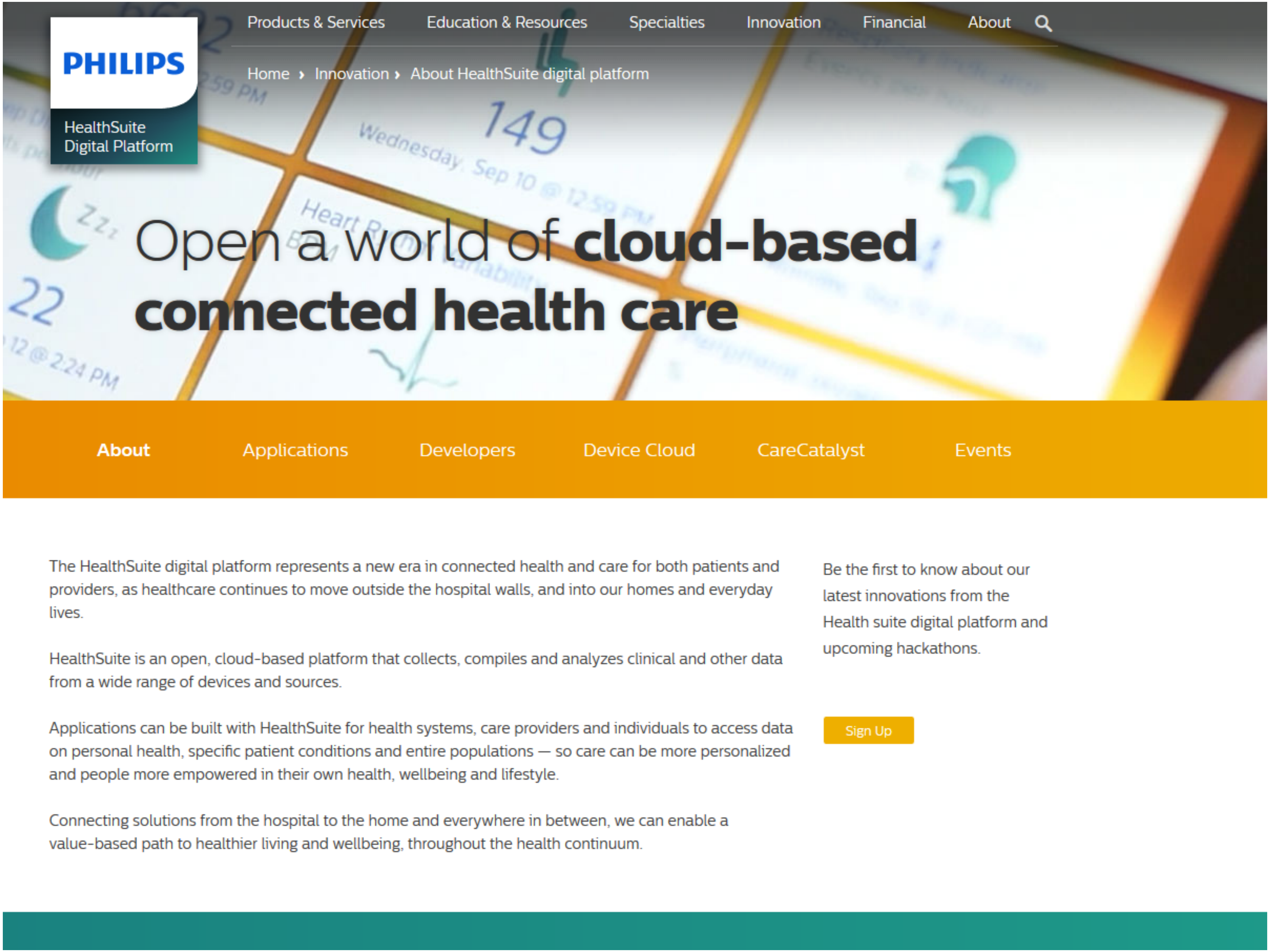
MPC 2016, Aarhus, May 30 2016

PHILIPS

PHILIPS
Series 9000
–
+

CHAMPIONS 2014 - 2015

PHILIPS
PHILIPS



Open a world of **cloud-based connected health care**

The HealthSuite digital platform represents a new era in connected health and care for both patients and providers, as healthcare continues to move outside the hospital walls, and into our homes and everyday lives.

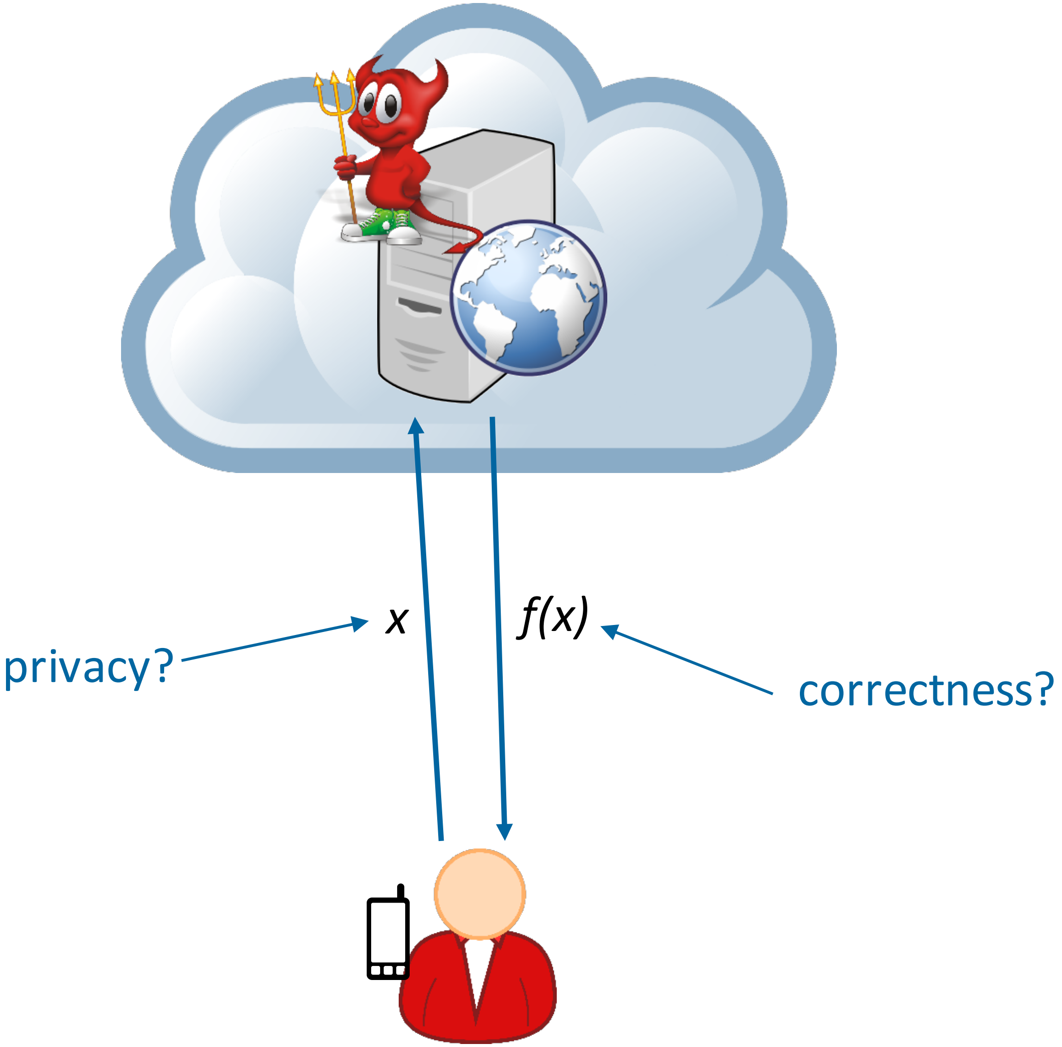
HealthSuite is an open, cloud-based platform that collects, compiles and analyzes clinical and other data from a wide range of devices and sources.

Applications can be built with HealthSuite for health systems, care providers and individuals to access data on personal health, specific patient conditions and entire populations — so care can be more personalized and people more empowered in their own health, wellbeing and lifestyle.

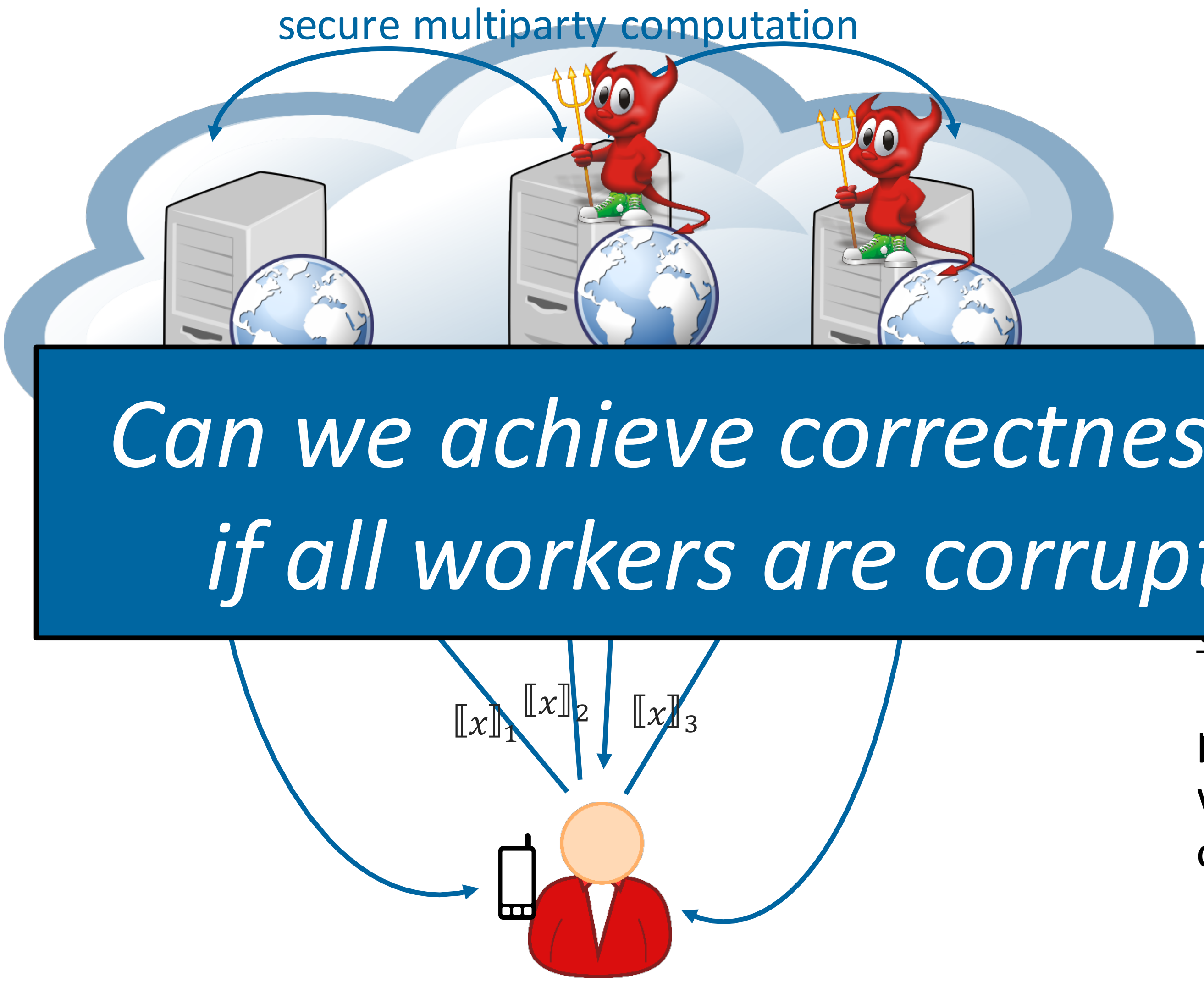
Connecting solutions from the hospital to the home and everywhere in between, we can enable a value-based path to healthier living and wellbeing, throughout the health continuum.

Be the first to know about our latest innovations from the Health suite digital platform and upcoming hackathons.

Outsourcing Computations on Sensitive Data (I)



Outsourcing Computations on Sensitive Data (I)

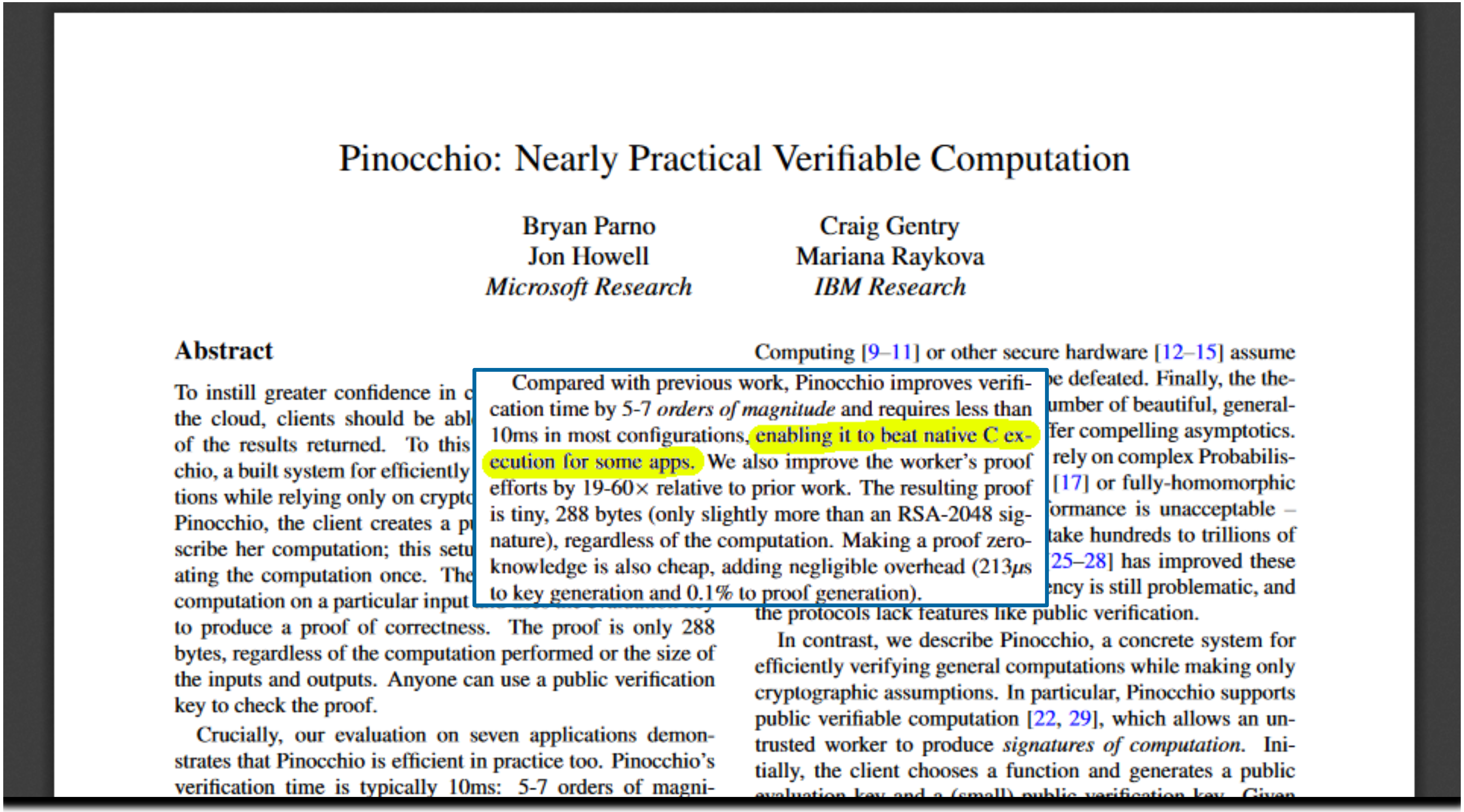


Can we achieve correctness even if all workers are corrupted?

[illegible] (CCSW '14):

privacy and correctness with $n-1$ actively corrupted workers

Outsourcing & Correctness (But No Privacy)



Pinocchio: Nearly Practical Verifiable Computation

Bryan Parno
Jon Howell
Microsoft Research

Craig Gentry
Mariana Raykova
IBM Research

Abstract

To instill greater confidence in c
the cloud, clients should be abl
of the results returned. To this
chio, a built system for efficiently
tions while relying only on crypto
Pinocchio, the client creates a pu
scribe her computation; this setu
ating the computation once. The
computation on a particular input
to produce a proof of correctness. The proof is only 288 bytes, regardless of the computation performed or the size of the inputs and outputs. Anyone can use a public verification key to check the proof.

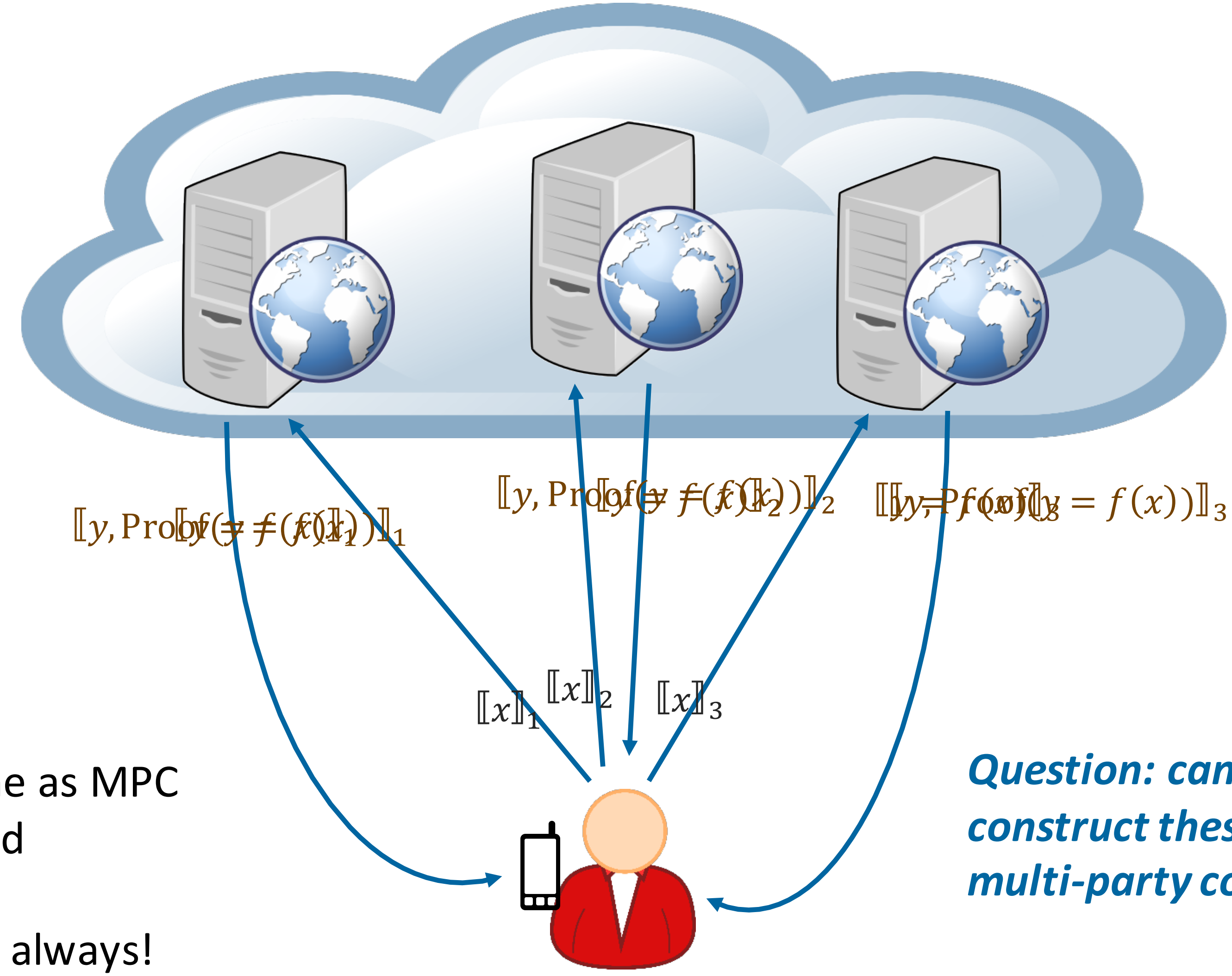
Crucially, our evaluation on seven applications demonstrates that Pinocchio is efficient in practice too. Pinocchio's verification time is typically 10ms: 5-7 orders of magni-

Compared with previous work, Pinocchio improves verification time by 5-7 *orders of magnitude* and requires less than 10ms in most configurations, enabling it to beat native C execution for some apps. We also improve the worker's proof efforts by 19-60× relative to prior work. The resulting proof is tiny, 288 bytes (only slightly more than an RSA-2048 signature), regardless of the computation. Making a proof zero-knowledge is also cheap, adding negligible overhead (213μs to key generation and 0.1% to proof generation).

Computing [9–11] or other secure hardware [12–15] assume
be defeated. Finally, the the-
umber of beautiful, general-
ffer compelling asymptotics.
rely on complex Probabilis-
[17] or fully-homomorphic
ormance is unacceptable –
take hundreds to trillions of
25–28] has improved these
ncy is still problematic, and
the protocols lack features like public verification.

In contrast, we describe Pinocchio, a concrete system for efficiently verifying general computations while making only cryptographic assumptions. In particular, Pinocchio supports public verifiable computation [22, 29], which allows an untrusted worker to produce *signatures of computation*. Initially, the client chooses a function and generates a public evaluation key and a (small) public verification key. Given

Privacy + Correctness: A Generic Construction

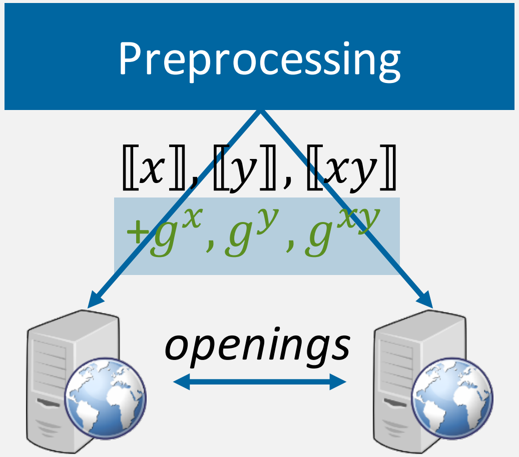

$[\![y, \mathrm{Proof}(y = f(x))]\!]_1$
$[\![y = f(x)]\!]_1$
$[\![y, \mathrm{Proof}(y = f(x))]\!]_2$
$[\![y = f(x)]\!]_2$
$[\![y, \mathrm{Proof}(y = f(x))]\!]_3$
$[\![y = f(x)]\!]_3$
$[\![x]\!]_1$
$[\![x]\!]_2$
$[\![x]\!]_3$

Privacy: same as MPC protocol used

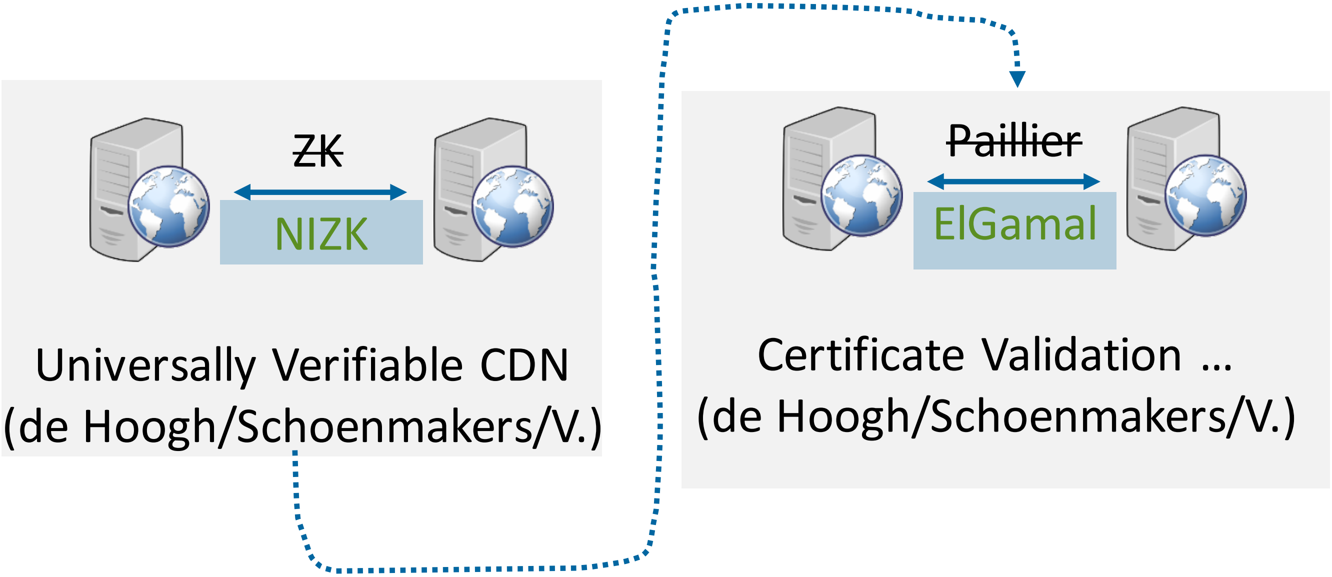
Correctness: always!

Question: can we efficiently construct these proofs with multi-party computation?

Privacy + Correctness: Previous Work



Publicly Auditable SPDZ
(Baum/Damgård/Orlandi)



Universally Verifiable CDN
(de Hoogh/Schoenmakers/V.)

Certificate Validation …
(de Hoogh/Schoenmakers/V.)

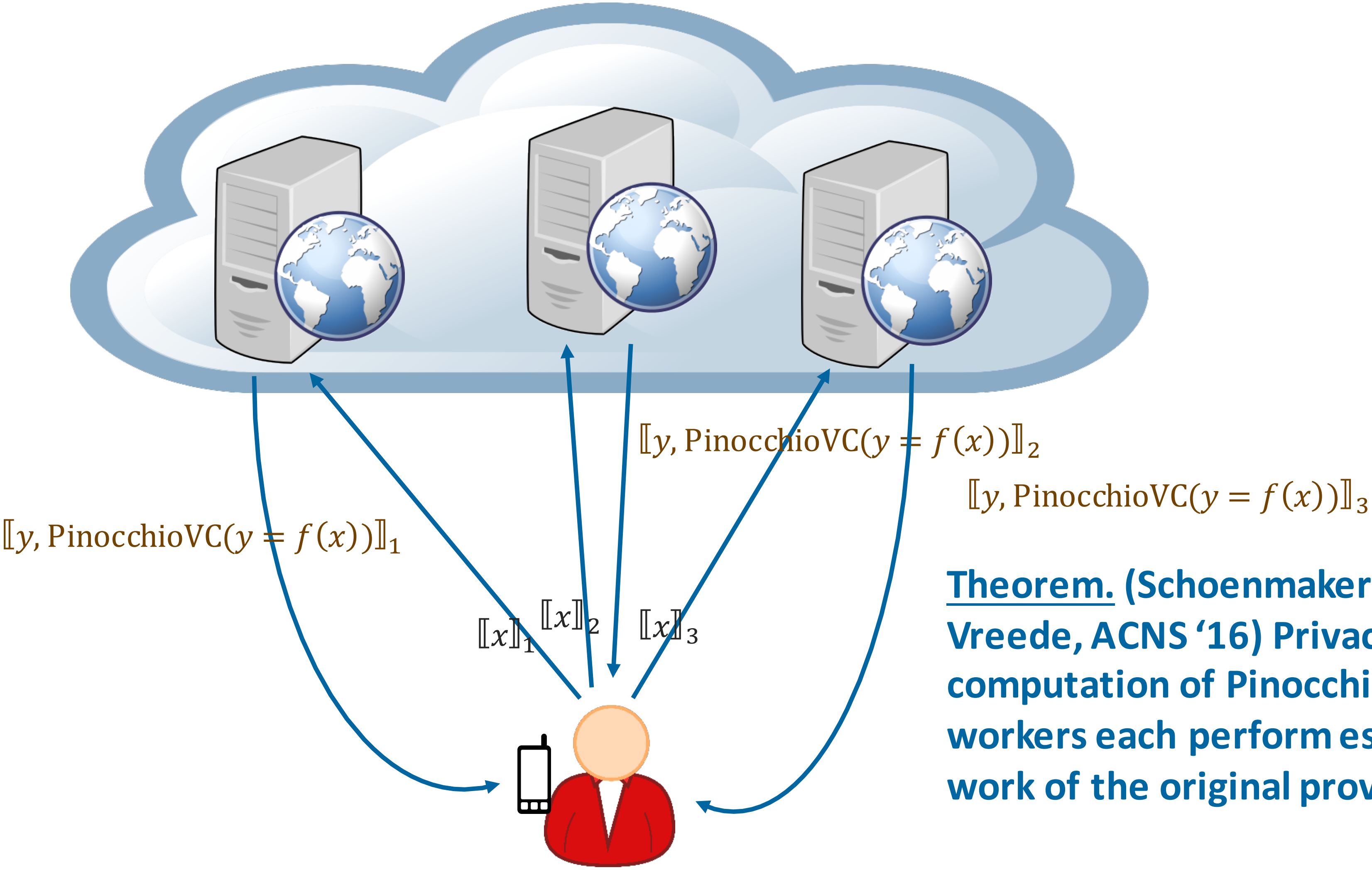
Verification effort scales in computation size!
Reason: existing work takes MPC as starting point!

Privacy + Correctness: Previous Work

- Instead of $[\![y, \mathrm{Proof}(y = f(x))]\!]_2$:
  - Baum/Damgård/Orlandi: SPDZ + Pedersen commitments = SPDZ’
  - de Hoogh/Schoenmakers/Veeningen: CDN + non-interactive proofs = CDN’
  - de Hoogh/Schoenmakers/Veeningen: CDN’ + ElGamal encryption = CDN’’

- Because of MPC starting point, no efficient verification!

Today: $[\![y, \mathrm{Proof}(y = f(x))]\!]$ can be efficient!


$[\![y, \mathrm{PinocchioVC}(y = f(x))]\!]_2$
$[\![y, \mathrm{PinocchioVC}(y = f(x))]\!]_3$
$[\![y, \mathrm{PinocchioVC}(y = f(x))]\!]_1$
$[\![x]\!]_1$
$[\![x]\!]_2$
$[\![x]\!]_3$

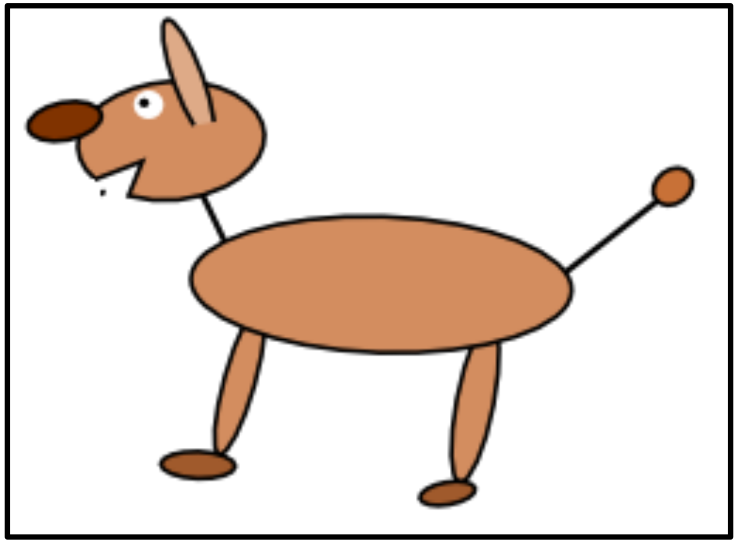
Theorem. (Schoenmakers/V/de Vreede, ACNS '16) Privacy-preserving computation of Pinocchio VC: three workers each perform essentially the work of the original prover.

Corollary. Verifiable Multi-Party Computation with constant-time verification!

Outline

- Secret sharing MPC
- Pinocchio VC

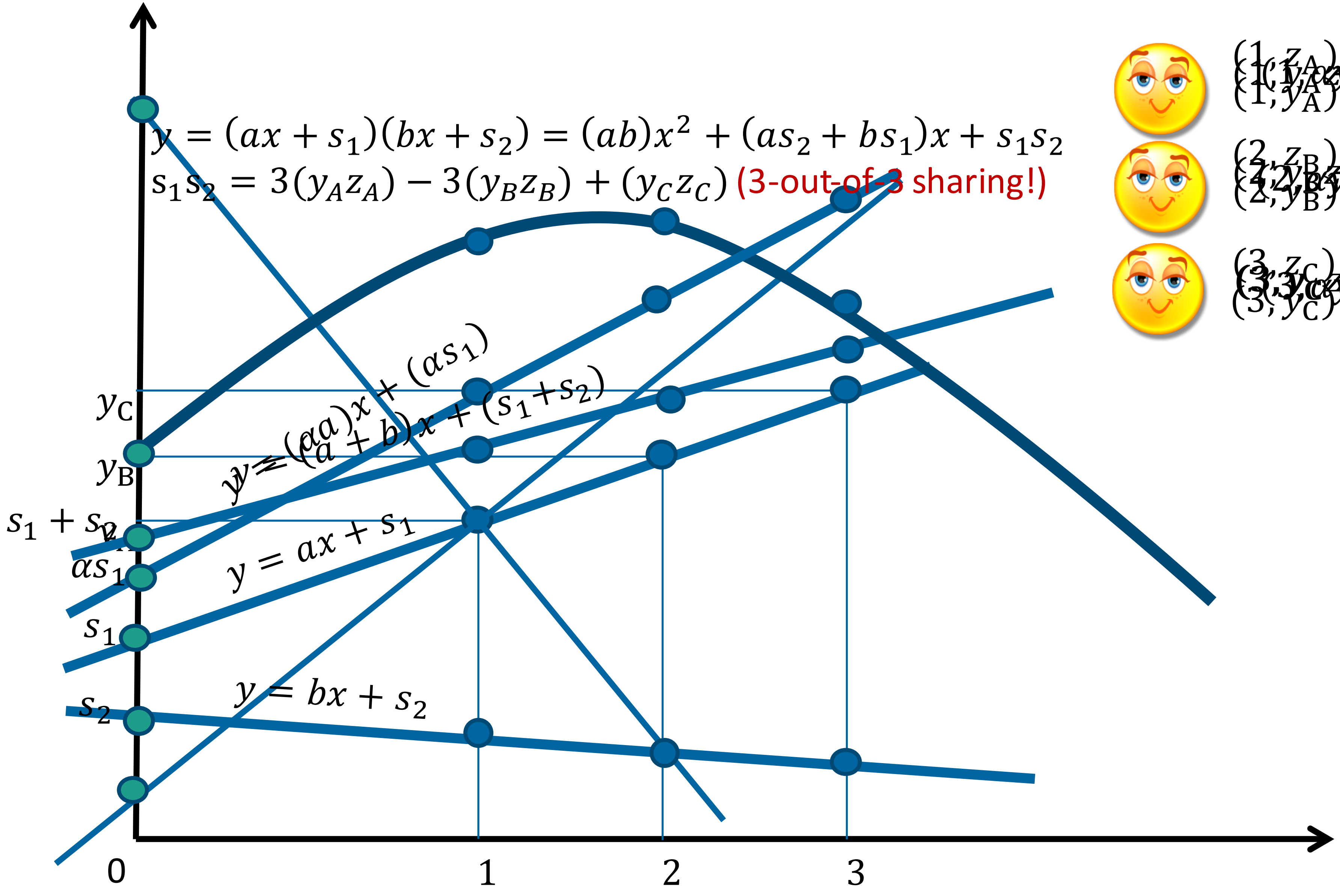
- Secret sharing MPC + Pinocchio VC



Secret sharing MPC

Animation: Sebastiaan de Hoogh

Shamir secret sharing (2-out-of-3)



$y = (ax + s_1)(bx + s_2) = (ab)x^2 + (as_2 + bs_1)x + s_1 s_2$

$s_1 s_2 = 3(y_A z_A) - 3(y_B z_B) + (y_C z_C)$ (3-out-of-3 sharing!)

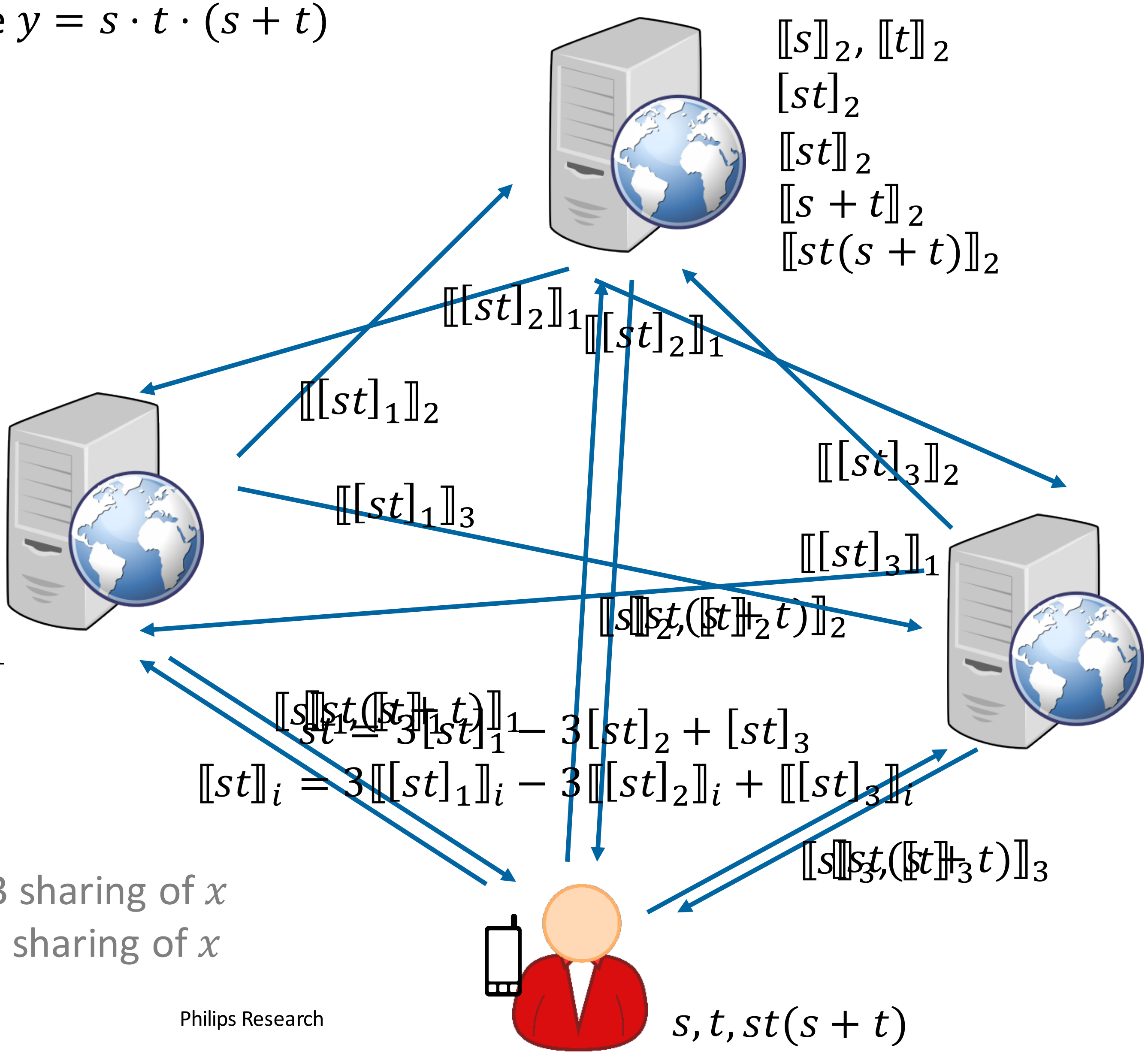
$(1, z_A)$
$(1, y_A z_A)$
$(1, \alpha z_A + y_A)$
$(1, y_A)$

$(2, z_B)$
$(2, y_B z_B)$
$(2, \alpha z_B + y_B)$
$(2, y_B)$

$(3, z_C)$
$(3, y_C z_C)$
$(3, \alpha z_C + y_C)$
$(3, y_C)$

MPC based on Shamir secret sharing

Goal: compute $y = s \cdot t \cdot (s + t)$



$[\![s]\!]_2, [\![t]\!]_2$

$[st]_2$

$[\![st]\!]_2$

$[\![s+t]\!]_2$

$[\![st(s+t)]\!]_2$

$[\![s]\!]_1, [\![t]\!]_1$

$[st]_1$

$[\![st]\!]_1$

$[\![s+t]\!]_1$

$[\![st(s+t)]\!]_1$

$[\![s]\!]_3, [\![t]\!]_3$

$[st]_3$

$[\![st]\!]_3$

$[\![s+t]\!]_3$

$[\![st(s+t)]\!]_3$

$st = 3[st]_1 - 3[st]_2 + [st]_3$

$[\![st]\!]_i = 3[\![[st]_1]\!]_i - 3[\![[st]_2]\!]_i + [\![[st]_3]\!]_i$

$s, t, st(s+t)$

$[\![x]\!]$: 2-out-of-3 sharing of x

$[x]$: 3-out-of-3 sharing of x

Pinocchio VC

Pinocchio: Quadratic Arithmetic Programs

Prove that committed $\vec{x}$ satisfies equations

$$(V \cdot \vec{x}) * (W \cdot \vec{x}) = (Y \cdot \vec{x})$$

"quadratic arithmetic program" (QAP)

Example: $y = s \cdot t \cdot (s + t)$ if and only if:

$$\exists z : \begin{cases} s & \cdot & t & = & z \\ z & \cdot & (s+t) & = & y \end{cases}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} s \\ t \\ z \\ y \end{pmatrix} * \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} s \\ t \\ z \\ y \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s \\ t \\ z \\ y \end{pmatrix}$$

E.g.: $(s\ t\ y\ z) = (3\ 2\ 6\ 30)$ is a solution

Pinocchio: From QAP to SNARK (I)

Prove that committed $\vec{x}$ satisfies equations $(V \cdot \vec{x}) * (W \cdot \vec{x}) = (Y \cdot \vec{x})$.

Define $V_i(\xi), W_i(\xi), Y_i(\xi)$ by “columnwise Lagrange interpolation”

$$\begin{array}{r}\text{value at 1}\\ \text{value at 2}\end{array}\begin{pmatrix}1&0&0&0\\0&0&1&0\end{pmatrix}\cdot\begin{pmatrix}s\\t\\z\\y\end{pmatrix} * \begin{pmatrix}0&1&0&0\\1&1&0&0\end{pmatrix}\cdot\begin{pmatrix}s\\t\\z\\y\end{pmatrix} = \begin{pmatrix}0&0&1&0\\0&0&0&1\end{pmatrix}\cdot\begin{pmatrix}s\\t\\z\\y\end{pmatrix}$$

$V_1(1) = 1, V_1(2) = 0$

$V_1(\xi) = 2 - \xi$

$W_2(1) = 1, W_2(2) = 1$

$W_2(\xi) = 1$

…

Consider polynomial $P_{\vec{x}}(\xi) = (V_1(\xi)s + V_2(\xi)t + \cdots) \cdot (W_1(\xi)s + \cdots) - (Y_1(\xi)s + \cdots)$:

- In $\xi = 1$: $P_{\vec{x}}(1) = (V_1(1)s + V_2(1)t + \cdots) \cdot (W_1(1)s + \cdots) - (Y_1(1)s + \cdots) = s \cdot t - z$
- In $\xi = 2$: $P_{\vec{x}}(2) = (V_1(1)s + V_2(1)t + \cdots) \cdot (W_1(1)s + \cdots) - (Y_1(1)s + \cdots) = z \cdot (s + t) - y$

So $(V \cdot \vec{x}) * (W \cdot \vec{x}) = (Y \cdot \vec{x})$
if and only if $P_{\vec{x}}(1) = P_{\vec{x}}(2) = 0$
if and only if $(\xi - 1) \cdot (\xi - 2) \mid P(\xi)$
if and only if there exists $h(\xi)$: $(\xi - 1) \cdot (\xi - 2) \cdot h(\xi) = P_{\vec{x}}(\xi)$

Pinocchio: From QAP to SNARK (II)

Example.

value at 1 →

value at 2 →

$$\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{pmatrix}\cdot\begin{pmatrix}s\\ t\\ z\\ y\end{pmatrix} * \begin{pmatrix}0 & 1 & 0 & 0\\ 1 & 1 & 0 & 0\end{pmatrix}\cdot\begin{pmatrix}s\\ t\\ z\\ y\end{pmatrix} = \begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\cdot\begin{pmatrix}s\\ t\\ z\\ y\end{pmatrix}$$

$V_1(\xi) = Y_3(\xi) = 2 - \xi$

$V_2(\xi) = V_4(\xi) = W_3(\xi) = W_4(\xi) = Y_1(\xi) = Y_2(\xi) = 0$

$V_3(\xi) = W_1(\xi) = Y_4(\xi) = \xi - 1$

$W_2(\xi) = 1$

Claim: $(s\ t\ z\ y)$ is solution *iff* there exists $h(\xi)$ such that

$$(\xi - 1)(\xi - 2)h(\xi) = \left(sV_1(\xi) + tV_2(\xi) + zV_3(\xi) + yV_4(\xi)\right)\cdot\left(sW_1(\xi) + tW_2(\xi) + zW_3(\xi) + yW_4(\xi)\right) - \left(sY_1(\xi) + tY_2(\xi) + zY_3(\xi) + yY_4(\xi)\right)$$

Claim: $(3\ 2\ 6\ 30)$ is solution *iff* there exists $h(\xi)$ such that

$$(\xi - 1)(\xi - 2)h(\xi) = \left(3V_1(\xi) + 2V_2(\xi) + 6V_3(\xi) + 30V_4(\xi)\right)\cdot\left(3W_1(\xi) + 2W_2(\xi) + 6W_3(\xi) + 30W_4(\xi)\right) - \left(3Y_1(\xi) + 2Y_2(\xi) + 6Y_3(\xi) + 30Y_4(\xi)\right)$$

PHILIPS

Pinocchio: From QAP to SNARK (III)

Lemma ⇒ $(3\ 2\ 6\ 30)$ is solution *iff* there exists $h(\xi)$ such that

$$(\xi-1)(\xi-2)h(\xi)=9\xi^2-27\xi+18$$

$$\begin{array}{llll} \boxed{\xi^2}-3\xi+2 & / & \boxed{9\xi^2}-27\xi+18 & \backslash\quad 9 \\ & & 9\,(\xi^2-3\xi+2) & \\ \cline{3-3} & & \hfill 0 & - \end{array}$$

$h(\xi)=9$ ✓

Pinocchio: From QAP to SNARK (IV)

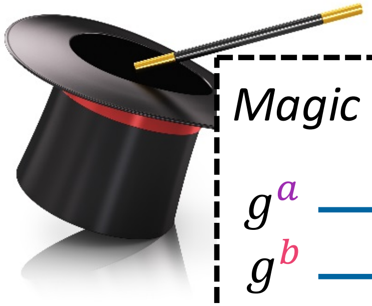
Ξ: random, unknown

evaluation key:
$g, g^{\Xi}, g^{\Xi^2}, \ldots$

evaluation/verification key:
$g^{V_i(\Xi)}, g^{W_i(\Xi)}, g^{Y_i(\Xi)}$

Prove: $\underbrace{(\Xi - 1)\cdot \ldots \cdot(\Xi - d)}_{\substack{\text{verification key:}\\ g^{(\Xi-1)\cdot\ldots\cdot(\Xi-d)}}} \cdot \underbrace{h(\Xi)}_{\substack{\text{prover:}\\ g^{h(\Xi)}}} = \underbrace{(V_1(\Xi)x_1+\cdots)}_{\substack{\text{prover/verifier:}\\ g^{V_1(\Xi)x_1+\cdots}}} \cdot \underbrace{(W_1(\Xi)x_1+\cdots)}_{\substack{\text{prover/verifier:}\\ g^{W_1(\Xi)x_1+\cdots}}} - \underbrace{(Y_1(\Xi)x_1+\cdots)}_{\substack{\text{prover/verifier:}\\ g^{Y_1(\Xi)x_1+\cdots}}} \cdot 1$

verifier: $e\left(g^{(\Xi-1)\cdot\ldots\cdot(\Xi-d)}, g^{h(\Xi)}\right) = e\left(g^{V_1(\Xi)x_1+\cdots}, g^{W_1(\Xi)x_1+\cdots}\right)\cdot e\left(g^{Y_1(\Xi)x_1+\cdots}, g\right)^{-1}$?



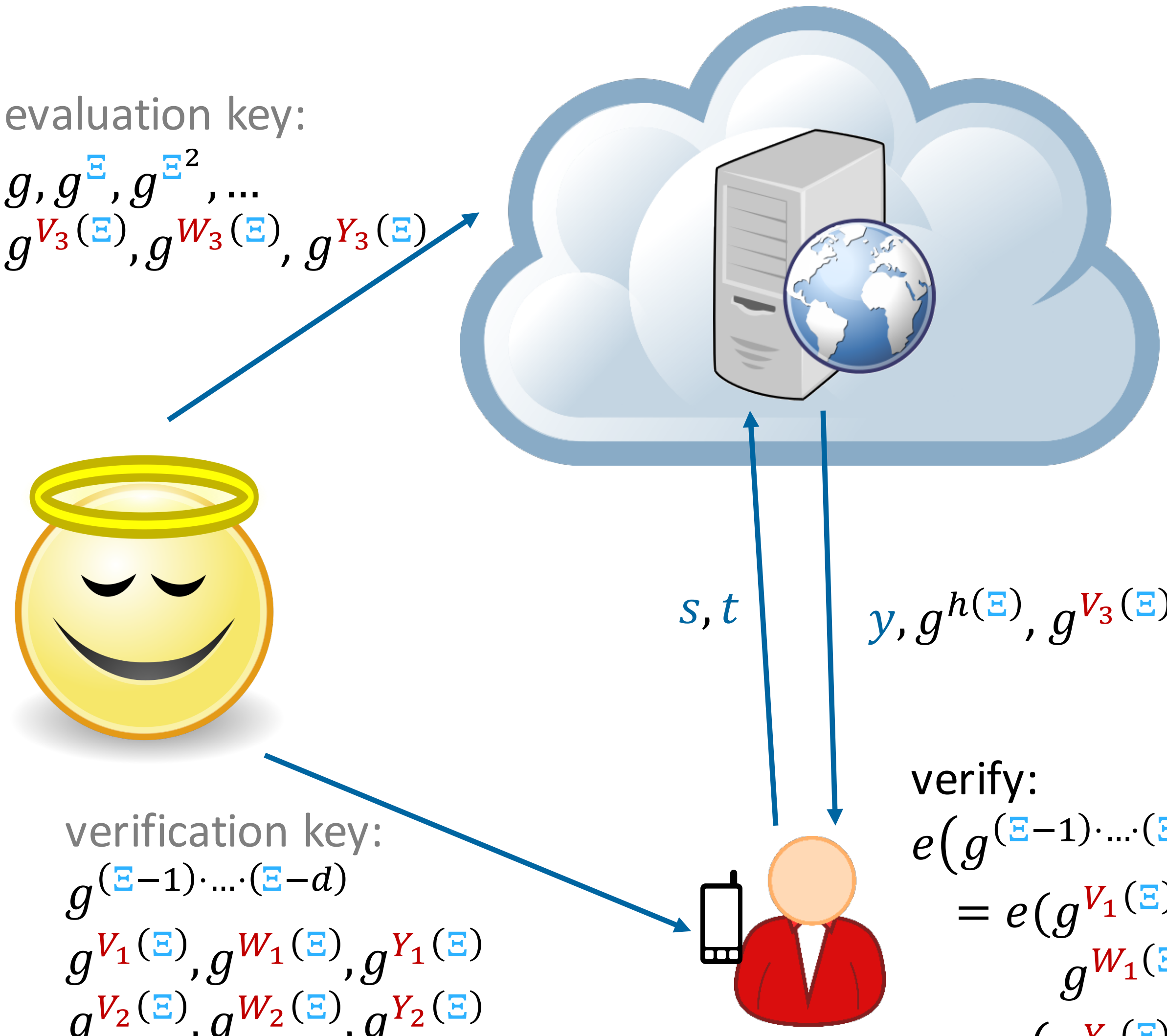
Magic crypto tool: pairing

$$\begin{tikzcd}
g^a \arrow[r] & \boxed{e} \arrow[r] & e(g^a,g^b) = e(g^c,g^d) & \boxed{e} \arrow[l] & g^c \arrow[l] \\
g^b \arrow[ru] & & & & g^d \arrow[lu]
\end{tikzcd}$$

iff

$a \cdot b = c \cdot d$

Pinocchio: From QAP to SNARK (V)



evaluation key:

$g, g^{\Xi}, g^{\Xi^2}, \ldots$

$g^{V_3(\Xi)}, g^{W_3(\Xi)}, g^{Y_3(\Xi)}$

- evaluate function: get z, y
- compute $g^{V_3(\Xi)z}, g^{W_3(\Xi)z}, g^{Y_3(\Xi)z}$
- compute $h(\xi) = \frac{V(\xi)W(\xi)-Y(\xi)}{(\xi-1)\cdot\ldots\cdot(\xi-d)}$
- compute $g^{h(\Xi)}$

s, t

$y, g^{h(\Xi)}, g^{V_3(\Xi)z}, g^{W_3(\Xi)z}, g^{Y_3(\Xi)z}$

verification key:

$g^{(\Xi-1)\cdot\ldots\cdot(\Xi-d)}$

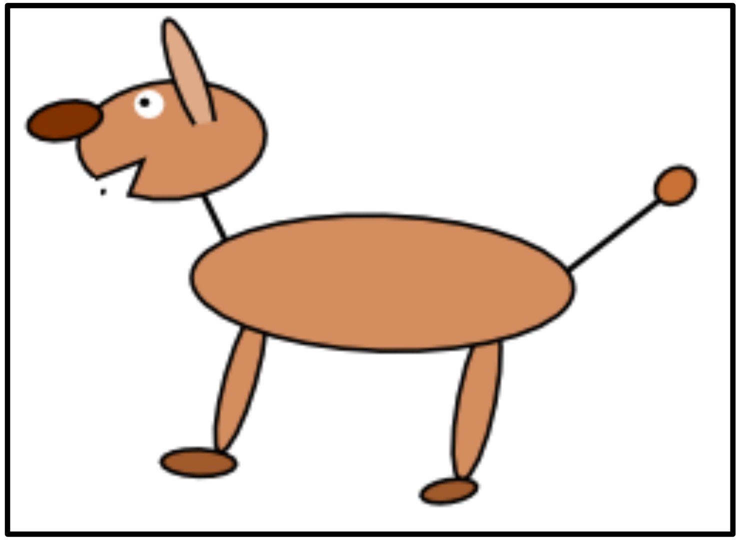
$g^{V_1(\Xi)}, g^{W_1(\Xi)}, g^{Y_1(\Xi)}$

$g^{V_2(\Xi)}, g^{W_2(\Xi)}, g^{Y_2(\Xi)}$

$g^{V_4(\Xi)}, g^{W_4(\Xi)}, g^{Y_4(\Xi)}$

verify:

$$e\left(g^{(\Xi-1)\cdot\ldots\cdot(\Xi-d)}, g^{h(\Xi)}\right)$$
$$= e(g^{V_1(\Xi)s+V_2(\Xi)t+V_4(\Xi)y} \cdot g^{V_3(\Xi)z},$$
$$g^{W_1(\Xi)s+W_2(\Xi)t+W_4(\Xi)y} \cdot g^{W_3(\Xi)z}) \cdot$$
$$e\left(g^{Y_1(\Xi)s+Y_2(\Xi)t+Y_4(\Xi)y} \cdot g^{Y_3(\Xi)z}, g\right)^{-1}$$



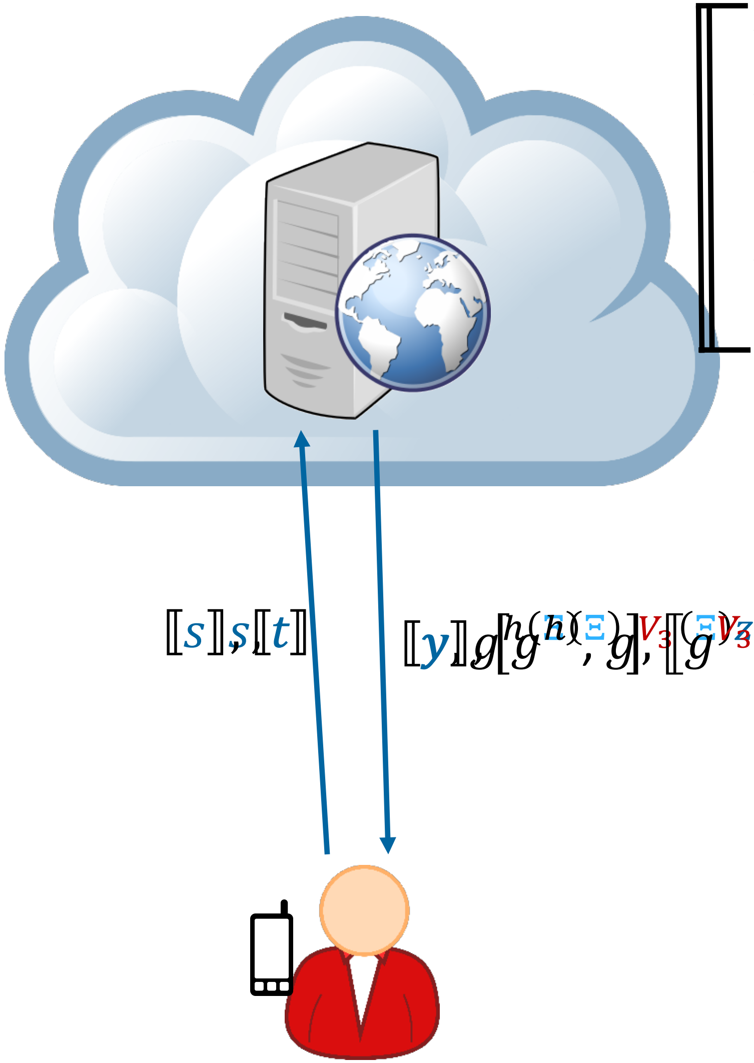
Secret sharing MPC

+

Pinocchio VC

Trinocchio: Distributing the Pinocchio System (I)

- evaluate function: get z, y
- compute $g^{V_3(\Xi)z}, g^{W_3(\Xi)z}, g^{Y_3(\Xi)z}$
- compute $h(\xi) = \frac{V(\xi)W(\xi)-Y(\xi)}{(\xi-1)\cdot\ldots\cdot(\xi-d)}$
- compute $g^{h(\Xi)}$



$[\![s]\!], s, [\![t]\!]$

$[\![y]\!], g^{h(\Xi)}, [\![g^{h(\Xi)}, g^{V_3(\Xi)z}]\!], [\![g^{V_3(\Xi)z}, g^{W_3(\Xi)z}]\!], [\![g^{W_3(\Xi)z}, g^{Y_3(\Xi)z}]\!], [\![g^{Y_3(\Xi)z}]\!]$

PHILIPS

Trinocchio: Distributing the Pinocchio System (II)

```
⟦prove$(g, g^{\Xi}, g^{\Xi^2}, \ldots, g^{V_3(\Xi)}, g^{W_3(\Xi)}, g^{Y_3(\Xi)}, s, t)$:
    $z, y = f(s, t)$
    $g^{V_3(\Xi)z} = \exp(g^{V_3(\Xi)}, z)$
    $g^{W_3(\Xi)z} = \exp(g^{W_3(\Xi)}, z)$
    $g^{Y_3(\Xi)z} = \exp(g^{Y_3(\Xi)}, z)$
    $n(\xi) = (V_1(\xi)s + V_2(\xi)t + V_3(\xi)z + V_4(\xi)y) * (W_1(\xi)s + \cdots) - (Y_1(\xi)s + \cdots)$
    $h(\xi) = \frac{n(\xi)}{(\xi-1)\cdot\ldots\cdot(\xi-d)}$
    $g^{h(\Xi)} = \exp(g, h_0) \cdot \exp(g^{\Xi}, h_1) \cdot \ldots \cdot \exp(g^{\Xi^{d-1}}, h_{d-1})$
    return $y, g^{h(\Xi)}, g^{V_3(\Xi)z}, g^{W_3(\Xi)z}, g^{Y_3(\Xi)z}$⟧
```

Trinocchio: Distributing the Pinocchio System (II)

$\mathrm{prove}(g, g^{\Xi}, g^{\Xi^2}, \ldots, g^{V_3(\Xi)}, g^{W_3(\Xi)}, g^{Y_3(\Xi)}, \llbracket s \rrbracket, \llbracket t \rrbracket):$

$\boxed{\llbracket z \rrbracket, \llbracket y \rrbracket = f(\llbracket s \rrbracket, \llbracket t \rrbracket)}$

$\llbracket g^{V_3(\Xi)z} \rrbracket = \exp(g^{V_3(\Xi)}, \llbracket z \rrbracket)$

$\llbracket g^{W_3(\Xi)z} \rrbracket = \exp(g^{W_3(\Xi)}, \llbracket z \rrbracket)$

$\llbracket g^{Y_3(\Xi)z} \rrbracket = \exp(g^{Y_3(\Xi)}, \llbracket z \rrbracket)$

$\llbracket n(\xi) \rrbracket = \llbracket (V_1(\xi)\llbracket s \rrbracket + V_2(\xi)\llbracket t \rrbracket + V_3(\xi)\llbracket z \rrbracket + V_4(\xi)\llbracket y \rrbracket) * (W_1(\xi)\llbracket s \rrbracket + \cdots) \rrbracket$
$- (Y_1(\xi)\llbracket s \rrbracket + \cdots)$

$\llbracket h(\xi) \rrbracket = \left\llbracket \frac{\llbracket n(\xi) \rrbracket}{(\xi - 1) \cdot \ldots \cdot (\xi - d)} \right\rrbracket$

$\left\llbracket g^{h(\Xi)} \right\rrbracket = \exp(g, \llbracket h_0 \rrbracket) \cdot \exp(g^{\Xi}, \llbracket h_1 \rrbracket) \cdots \exp(g^{\Xi^{d-1}}, \llbracket h_{d-1} \rrbracket)$

$\mathrm{return}\ \llbracket y \rrbracket, \llbracket g^{h(\Xi)} \rrbracket, \llbracket g^{V_3(\Xi)z} \rrbracket, \llbracket g^{W_3(\Xi)z} \rrbracket, \llbracket g^{Y_3(\Xi)z} \rrbracket$

MPC computation of f gives internal wire values "for free"

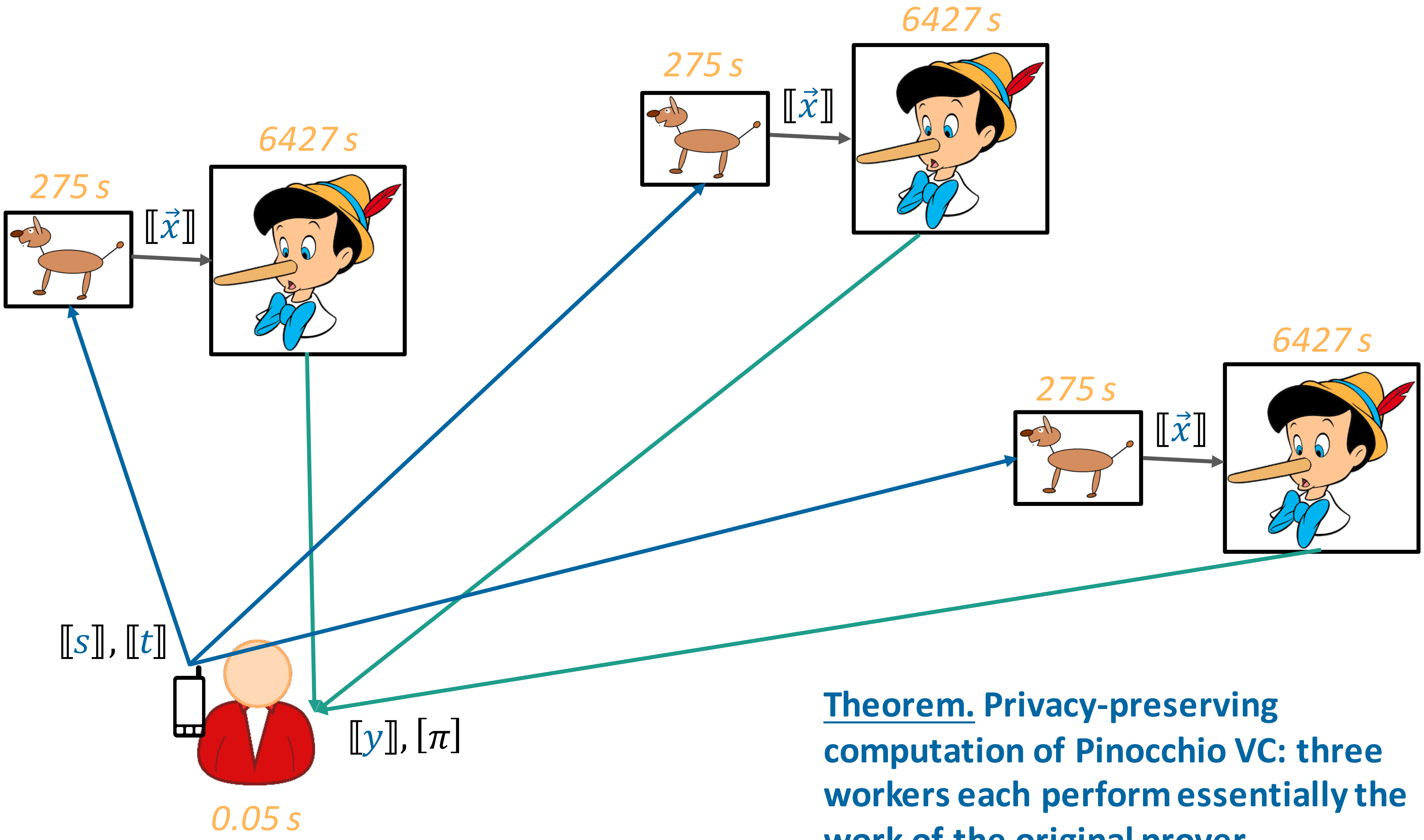
Shamir reconstruction "in the exponent"

Only step in which the workers communicate!

Products of 2-out-of-3 shares give 3-out-of-3 shares

Division by public polynomial is linear!

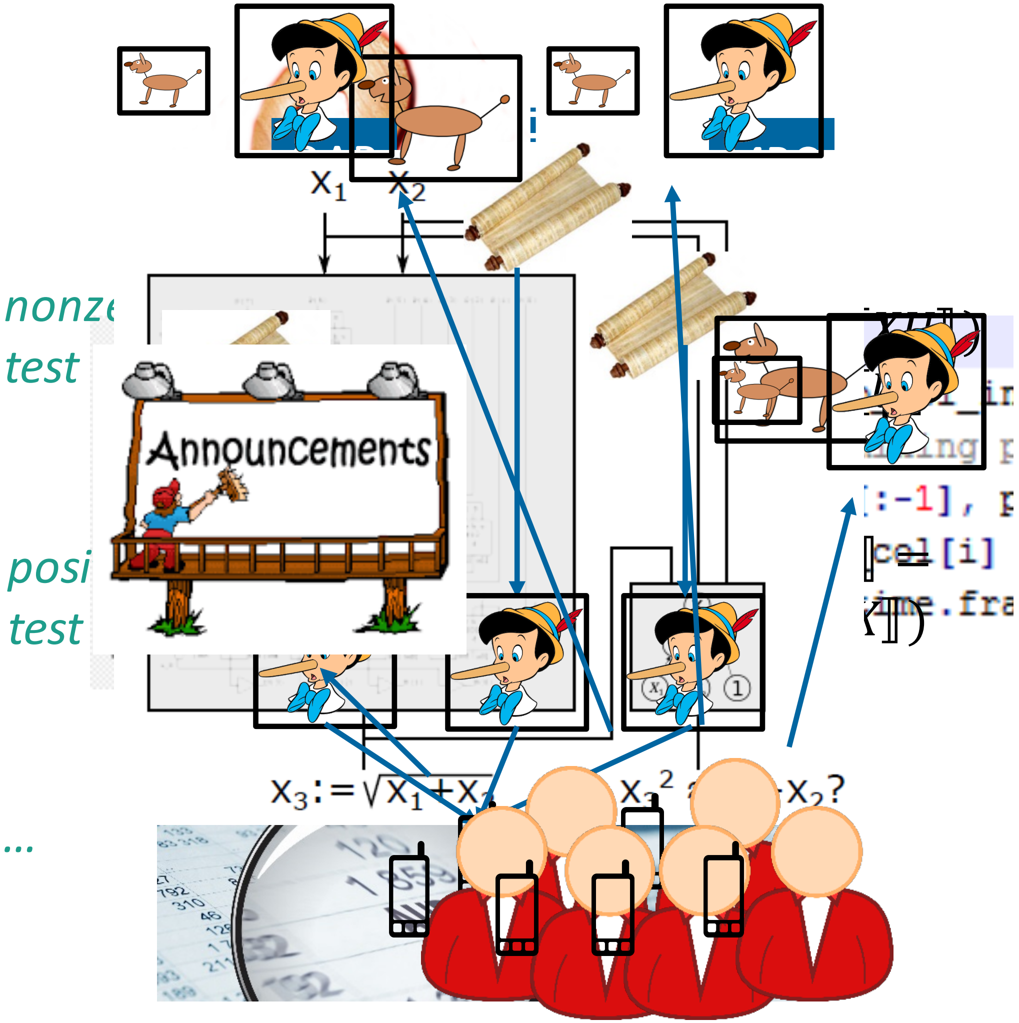
Trinocchio: Distributing the Pinocchio System (III)



Theorem. **Privacy-preserving computation of Pinocchio VC: three workers each perform essentially the work of the original prover.**

Extensions / Future Directions

- Multiple inputters
- Auditable MPC
- Verifiability by certificate validation
- QAPs + MPC for particular tasks?
  - Zero testing
  - Comparison
  - …
- Easily programmable distributed verifiable computation



PHILIPS