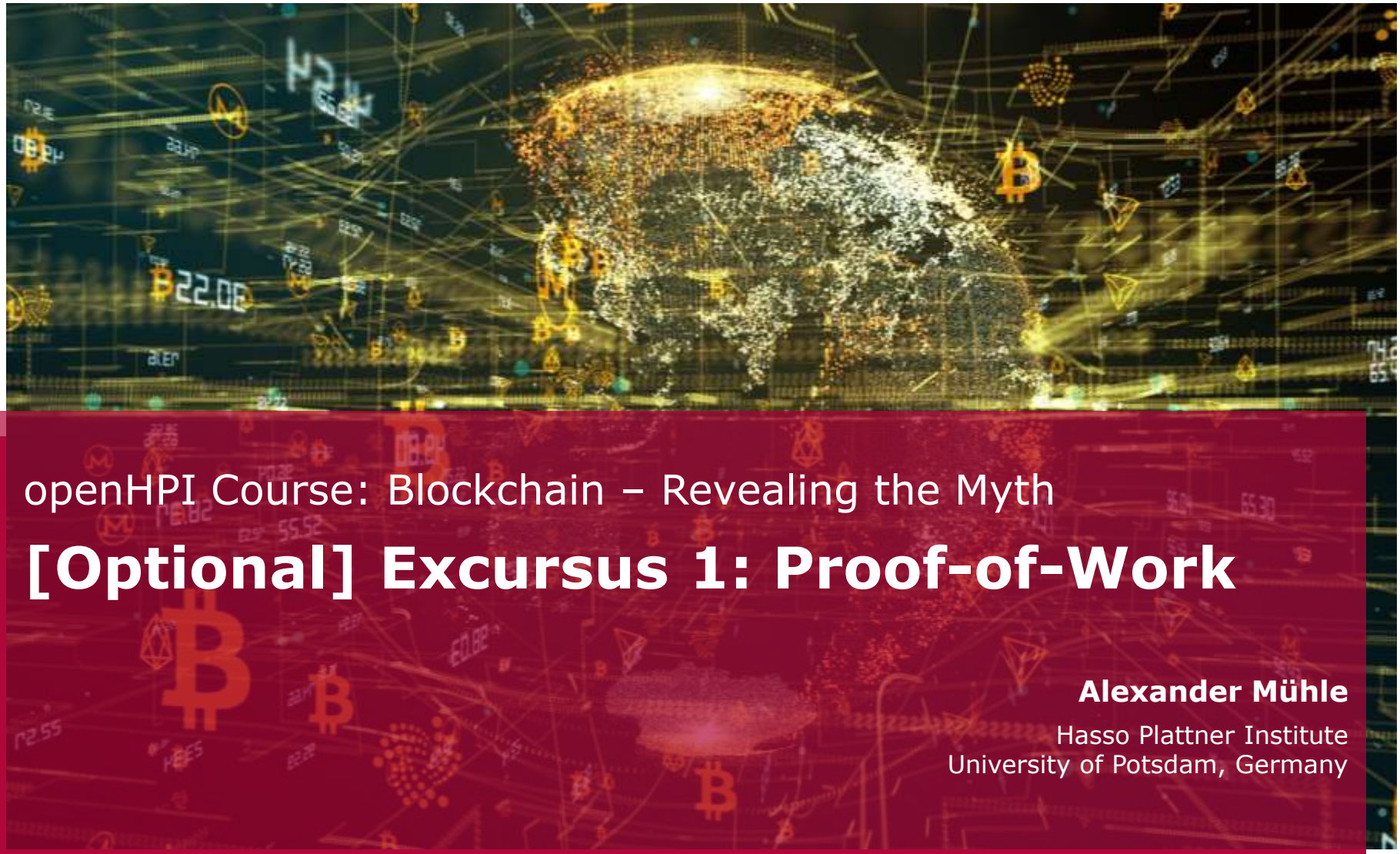


openHPI Course: Blockchain – Revealing the Myth

[Optional] Excursus 1: Proof-of-Work

Alexander Mühle

Hasso Plattner Institute
University of Potsdam, Germany

Sybil Attack

Review

- **Everyone** in our network **can vote** for the **proper order of transactions**
- Therefore, the majority decision is represented by the **majority of votes**

Problem

„You can have as many electronic identities as you have time and energy to create"

- Judith S. Donath

Analogue Voting

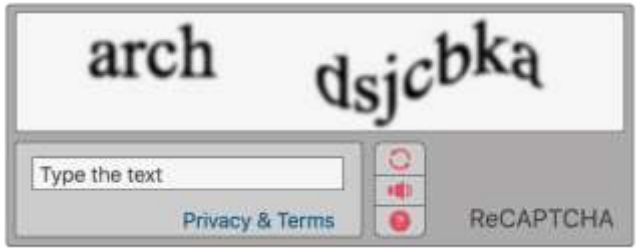
- Participants have to be prevented from creating multiple identities
- Compare with real-life democracies:
  - Passport/ID prevents voting with false name
  - Voter registration lists prevent voting at multiple locations

- Requires a **central entity** (i.e state)!

Alternatives

Problem: Make it infeasible to vote multiple times without relying on an Identity Provider

Method	Function	Example
CAPTCHAS	Requires human input	Website Registration
Verified Buyer	Requires capital expenditure	eCommerce



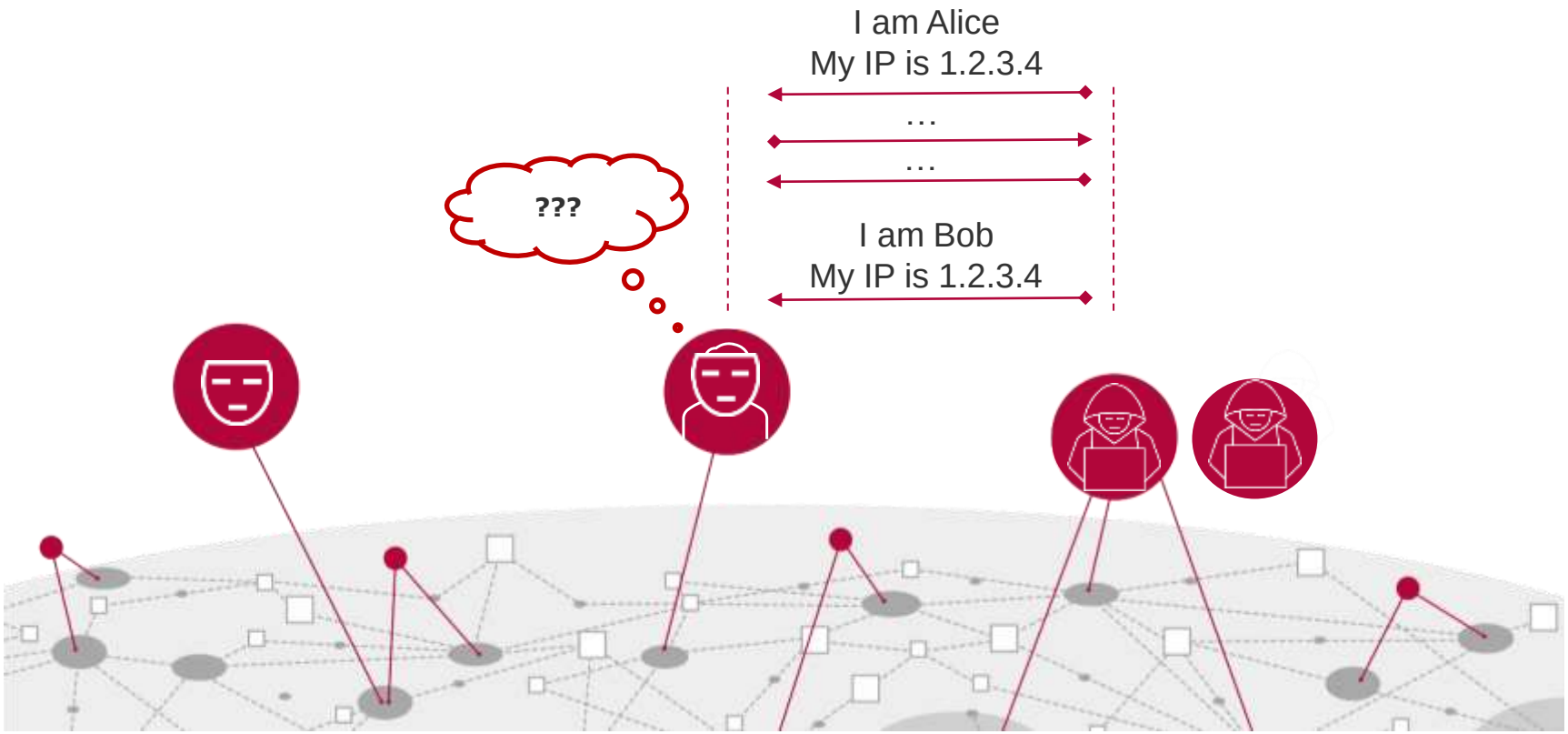
Resource Testing

Hypothesis: An identity which is faked by an attacker has less resources than a real identity

- By „Resource Testing“ a participant it should be possible to distinguish fake and real identities
- Examples of testable resources:
  - Computing power
  - Storage capacity
  - Network bandwidth
  - IP addresses
  - …

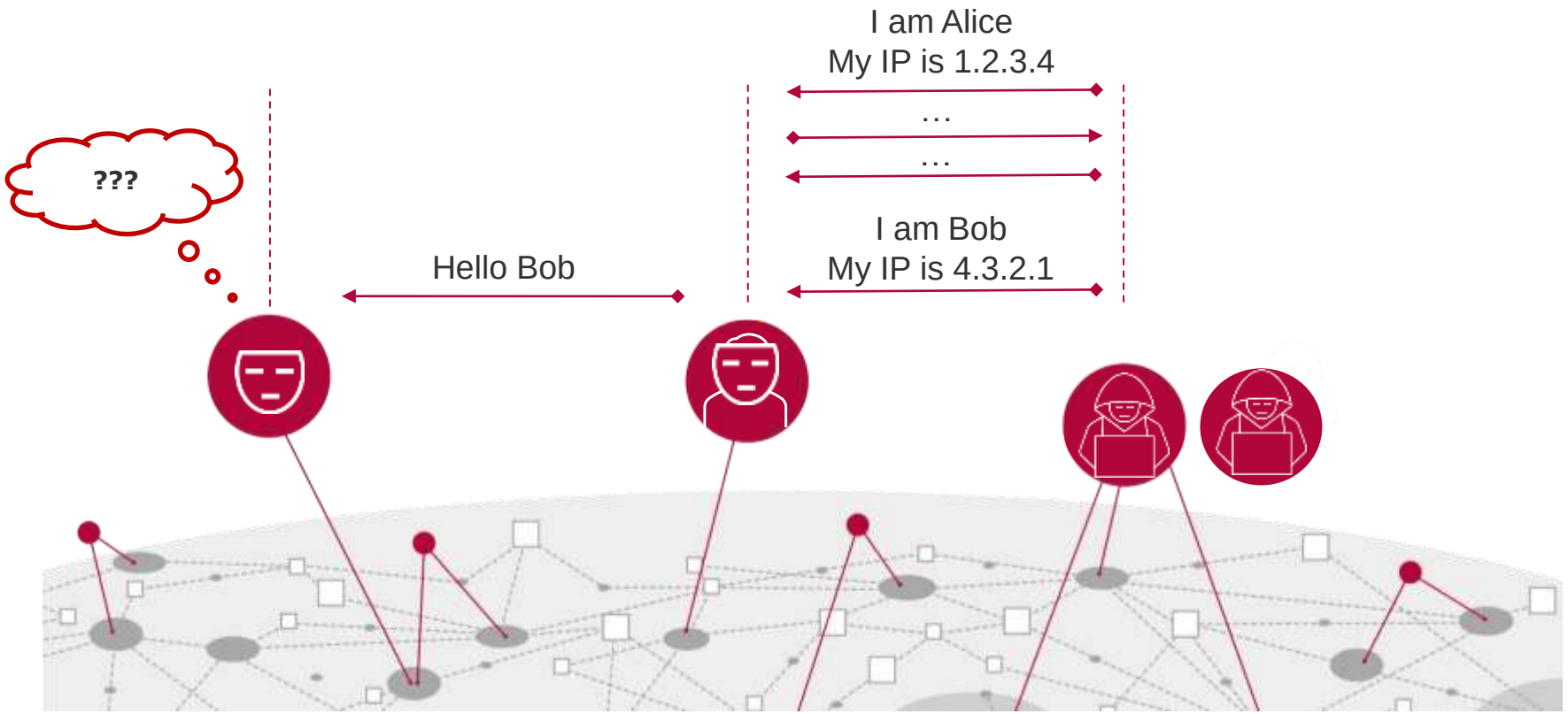
Resource Testing
IP addresses

- Does a participant have their own IP address?
  - IPv4 addresses are expensive (~22€)



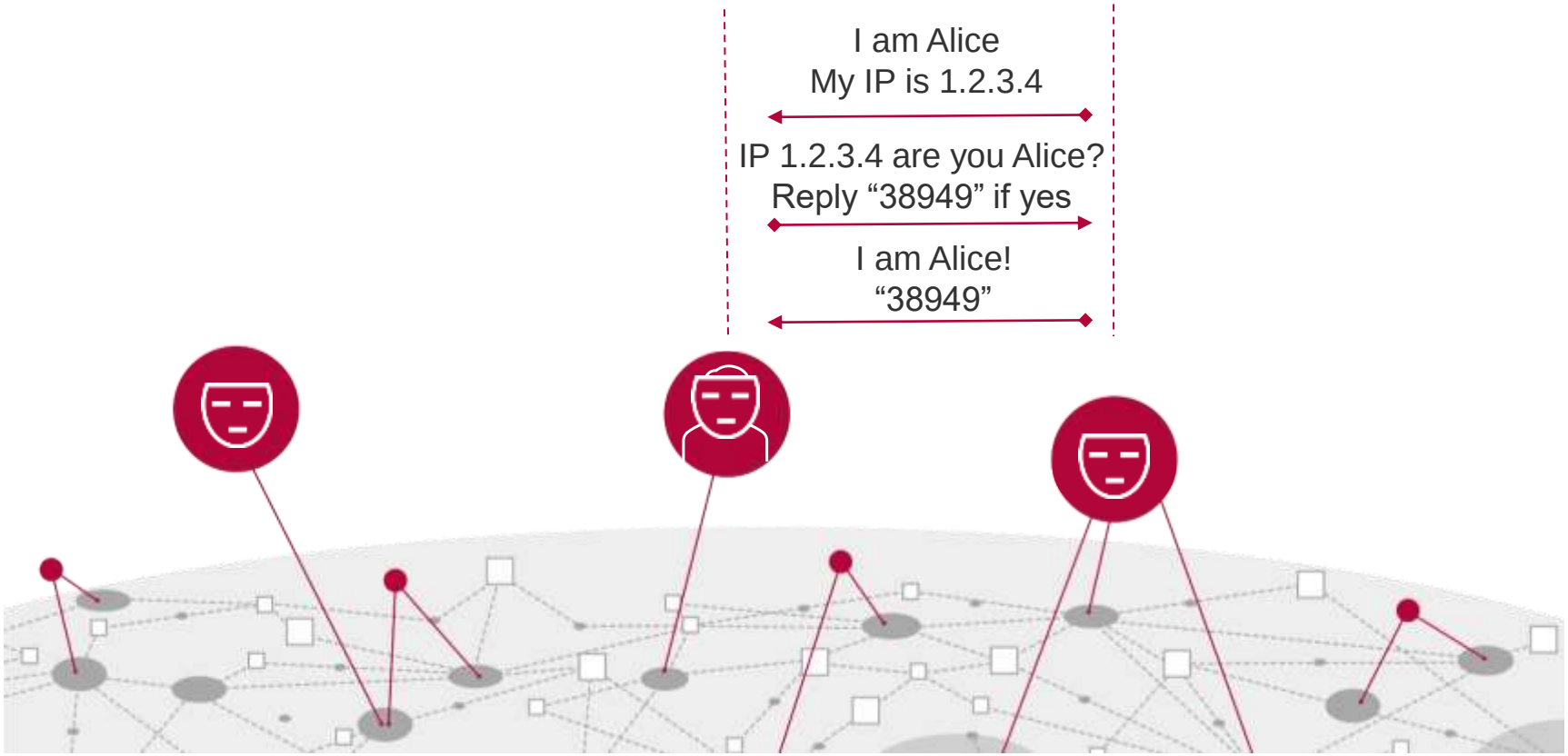
Resource Testing
IP addresses

- Does a participant have their own IP address?
  - IPv4 addresses are expensive (~22€)



Resource Testing
IP addresses

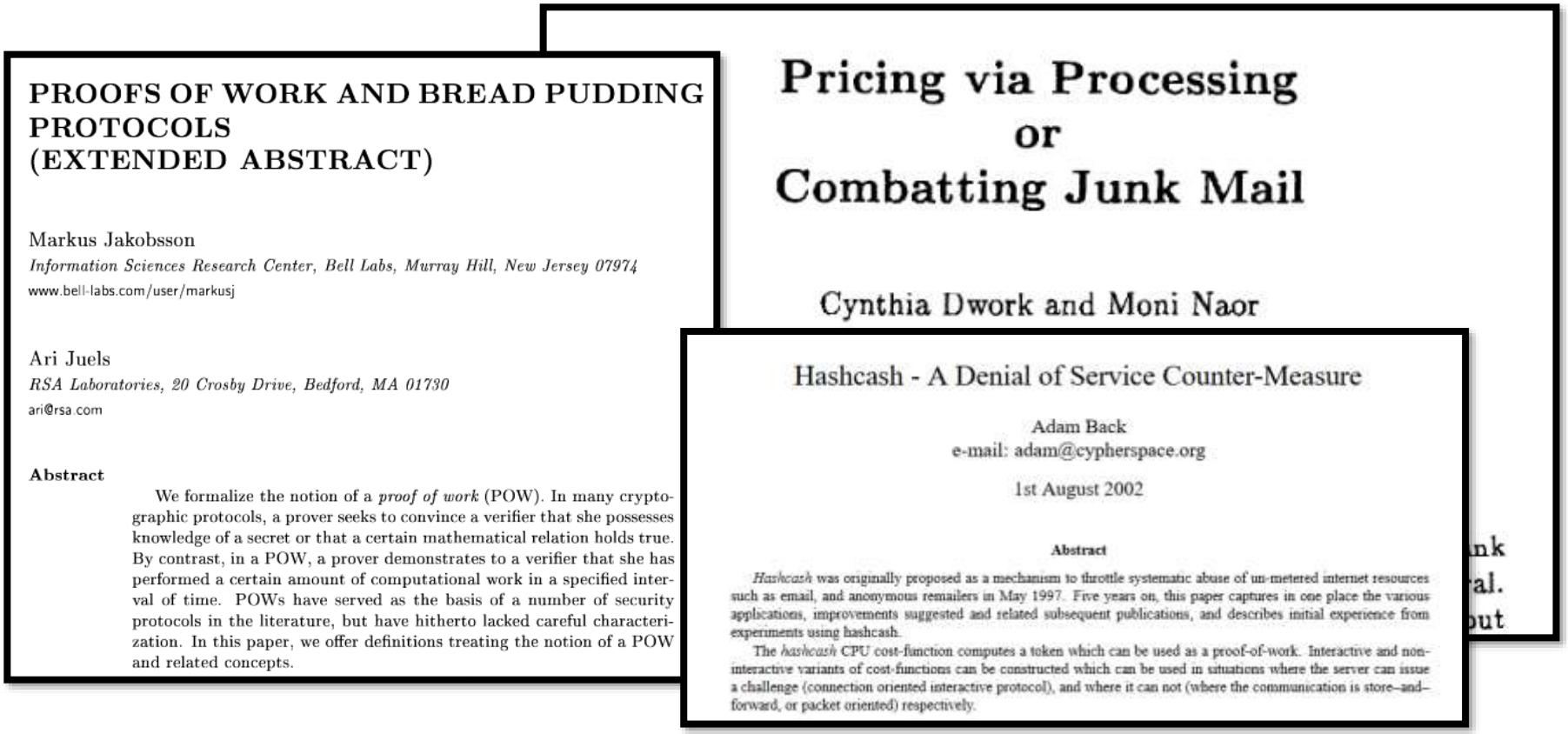
- Challenge-Response Protocol



Resource Testing

Problem: In a decentralised system, who performs the challenge/reponse?

- Resource testing needs to be non-interactive



PROOFS OF WORK AND BREAD PUDDING PROTOCOLS
(EXTENDED ABSTRACT)

Markus Jakobsson
Information Sciences Research Center, Bell Labs, Murray Hill, New Jersey 07974
www.bell-labs.com/user/markusj

Ari Juels
RSA Laboratories, 20 Crosby Drive, Bedford, MA 01730
ari@rsa.com

Abstract

We formalize the notion of a *proof of work* (POW). In many cryptographic protocols, a prover seeks to convince a verifier that she possesses knowledge of a secret or that a certain mathematical relation holds true. By contrast, in a POW, a prover demonstrates to a verifier that she has performed a certain amount of computational work in a specified interval of time. POWs have served as the basis of a number of security protocols in the literature, but have hitherto lacked careful characterization. In this paper, we offer definitions treating the notion of a POW and related concepts.

Pricing via Processing
or
Combatting Junk Mail

Cynthia Dwork and Moni Naor

Hashcash - A Denial of Service Counter-Measure

Adam Back
e-mail: adam@cypherspace.org

1st August 2002

Abstract

Hashcash was originally proposed as a mechanism to throttle systematic abuse of un-metered internet resources such as email, and anonymous remailers in May 1997. Five years on, this paper captures in one place the various applications, improvements suggested and related subsequent publications, and describes initial experience from experiments using hashcash.

The *hashcash* CPU cost-function computes a token which can be used as a proof-of-work. Interactive and non-interactive variants of cost-functions can be constructed which can be used in situations where the server can issue a challenge (connection oriented interactive protocol), and where it can not (where the communication is store–and–forward, or packet oriented) respectively.

Resource Testing

Requirements for computational task

- Hard to calculate
- Easy to verify
- Verifiable by everyone

Solution

- Find a hash that looks a certain way

Confidential Communication
in the Internet

Week 1 Video 8

Resource Testing

Proof-of-Work

Hash function using **message "M"** and **nonce "n"** as input ➔ h(M || n)

Task

Find a nonce "n" so that h(M || n) is below a **target "t"**

- Only known algorithm: "Brute-Force"
- The lower the target the more difficult the task becomes
- Can also be described as finding h(M || n) with k zeros at the front

Resource Testing

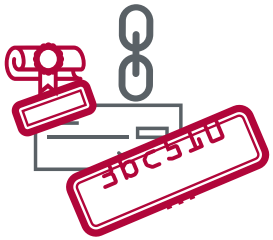
Non-interactive

- Proof of the resource testing through calculation of h(M || n) can be
  - Calculated without interaction
  - Can be verified later without interaction
- **BUT** can be calculated in advance

Challenge

- Adding a non-predictable **challenge "c"** prevents pre-calculation
- **h(M || n || c)**
- **BUT** how to pose a challenge without interaction?
- The **head of the previous block** in the blockchain acts as challenge

Resource Testing
IP addresses

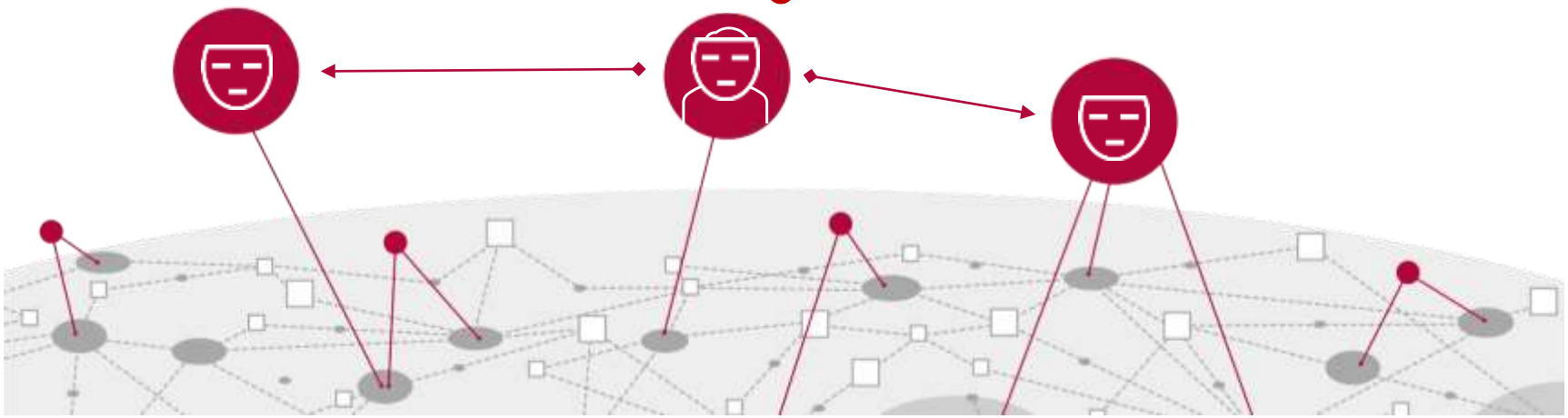


Target: 0000098…

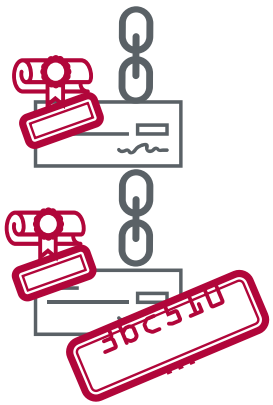
H(“I am Alice” || 1 || 3bc510…)
c04269e12ee5d0329ea902792c3aa903

H(“I am Alice” || 2 || 3bc510…)
10296043d21fa3d1f4da45fa17d642c5

…

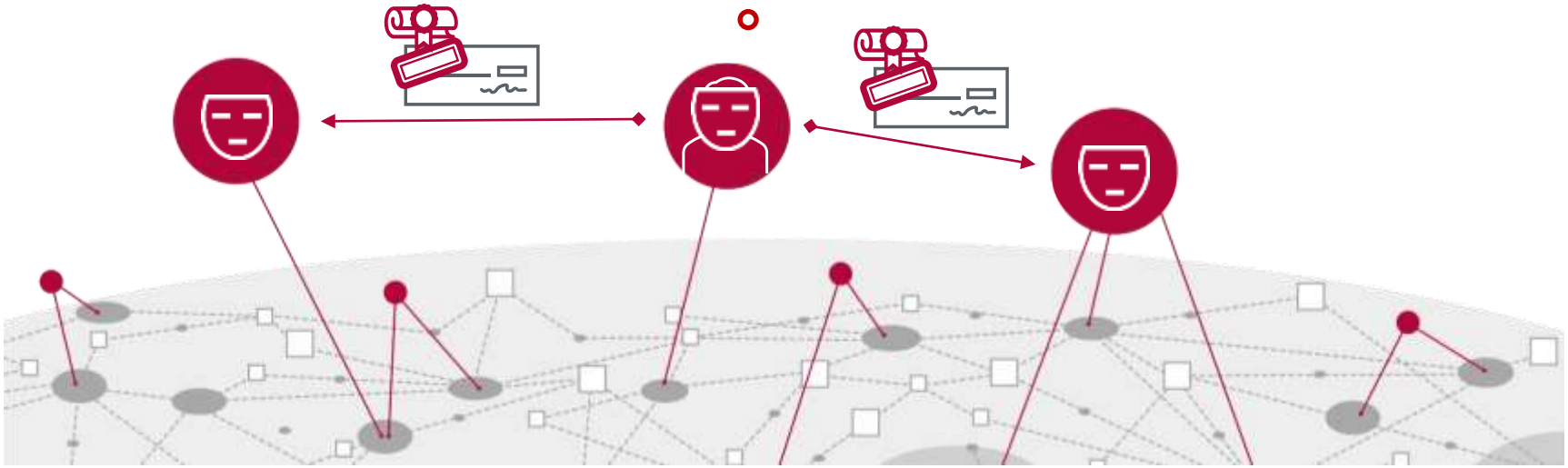


Resource Testing
IP addresses

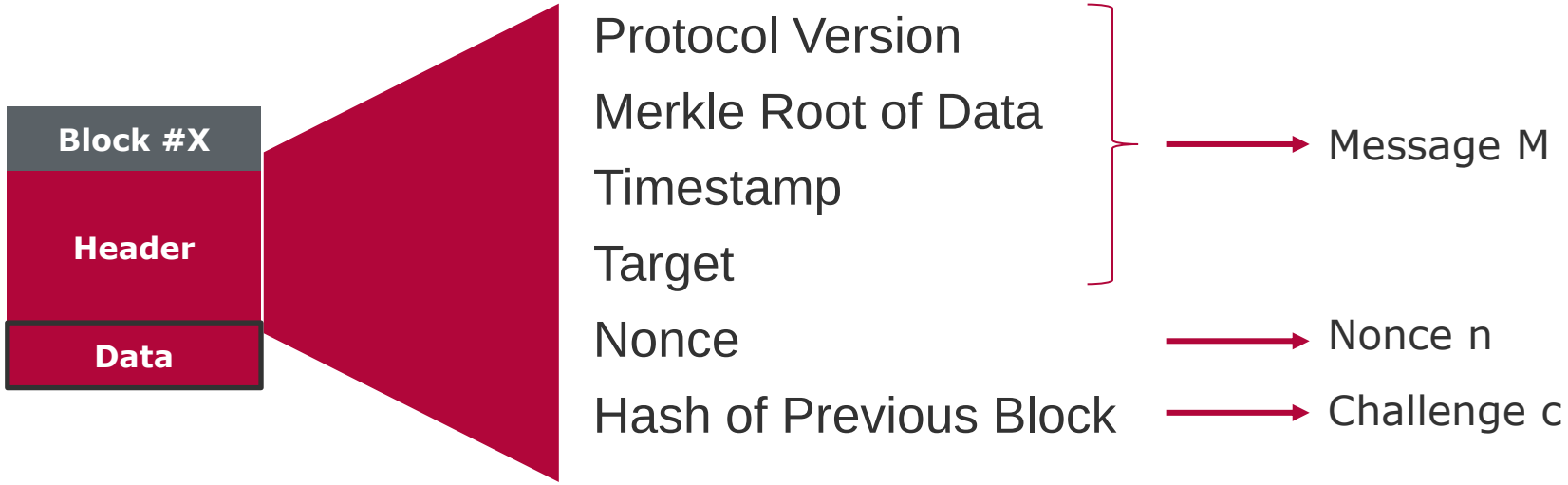


H(“I am Alice” || 89435792 || 3bc510…)
0000068db428fc8374183d8a8b3485f

Target: 0000098…



Bitcoin
Proof-of-Work



- Hash Function used: SHA-256

Resource Testing
Computing Power

CPU

- **C**entral **P**rocessing **U**nit
- Not competitive anymore
- **~100 – 24.000 Hash/Second**

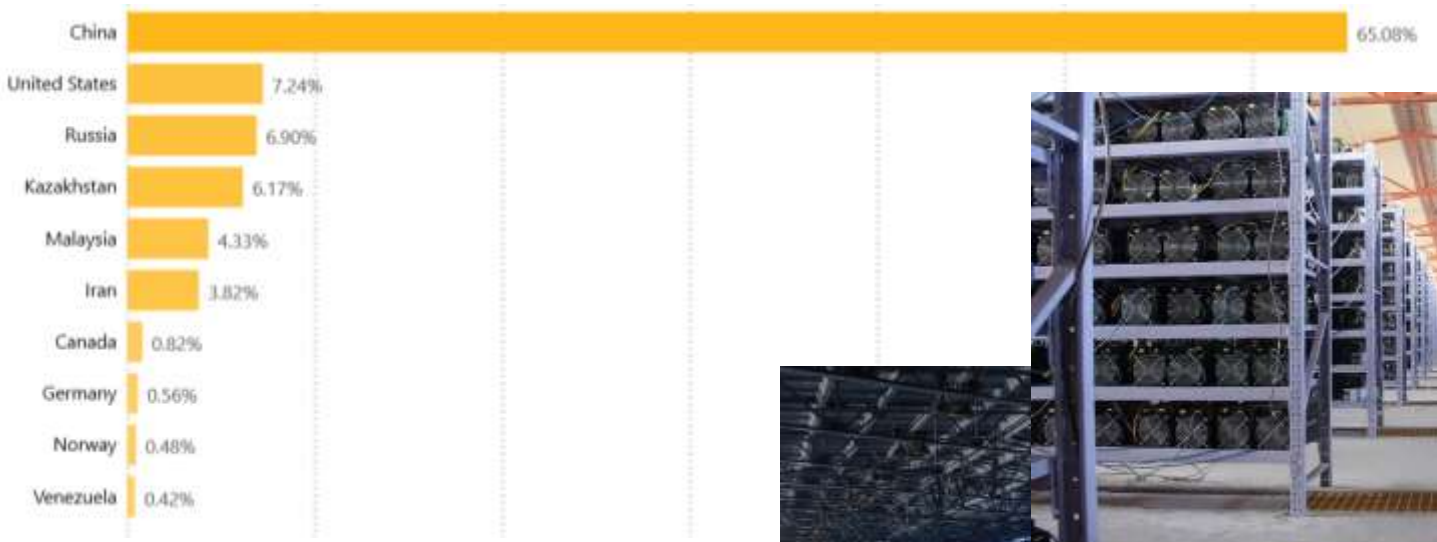
GPU

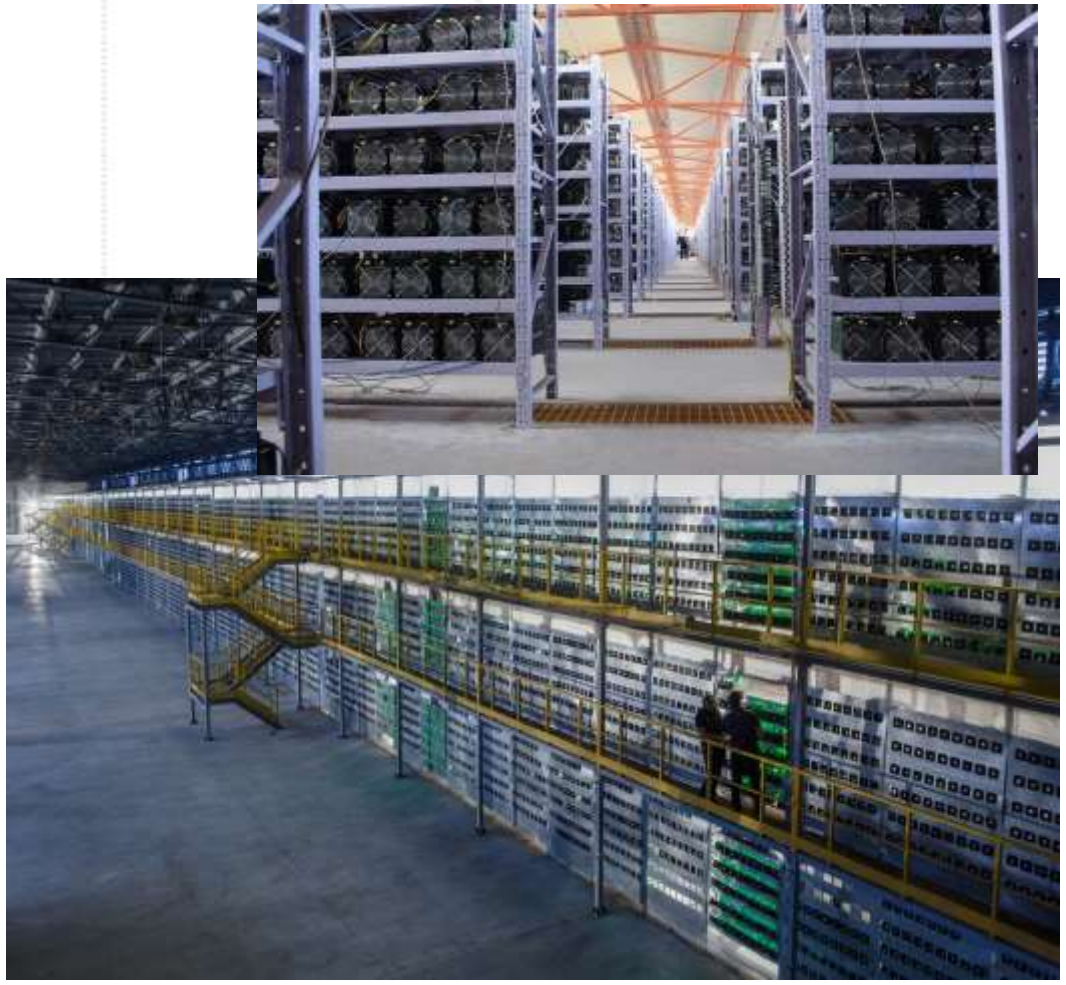
- **G**raphics **P**rocessing **U**nit
- A lot of processing units (high parallelism)
- 10.000 – 50.000.000 Hash/Second

ASIC

- **A**pplication **S**pecific **I**ntegrated **C**ircuits
- 10^{12} (Trillion) Hash/Second

Resource Testing
Hash Power Distribution





Resource Testing
Correlation of Price and Computing Power