Georgia Tech College of Computing

Securely Outsourcing Garbled Circuit Evaluation

USENIX Security Symposium 2013

Henry "Hank" Carter
Patrick Traynor

Benjamin Mood
Kevin Butler

SMC on mobile devices

- Mobile devices loaded with private and context-sensitive information and applications that use this information
- Secure Multiparty Computation (SMC) allows computation over encrypted inputs
- Highly constrained system resources (memory, power, processing, communication)

Why don’t we have mobile SMC?

- The dominant construction, garbled circuits, require too much memory and processing power
- Special purpose protocols can be optimized, but no efficient general purpose techniques
- Wish: an efficient mobile two-party SMC scheme that generalizes to any function

Leveraging the cloud?

- Kreuter et al. provide an efficient way to perform maliciously secure SMC in large servers
- Assuming a device has a connection to a cloud service, can the expensive computation associated with garbled circuits be outsourced?
- We cannot simply trust the cloud.

Outsourcing Garbled Circuit Evaluation

- Setting: A limited mobile device (Alice) communicating with a web server (Bob). Alice also has access to a cloud service (Cloud).
- Goal: Alice and Bob securely compute a two-party function using garbled circuits. We consider the case where Bob generates the circuit and Alice evaluates.
- Security:
  - Preserve input and output privacy from both the other party and the cloud
  - Security in the malicious setting.

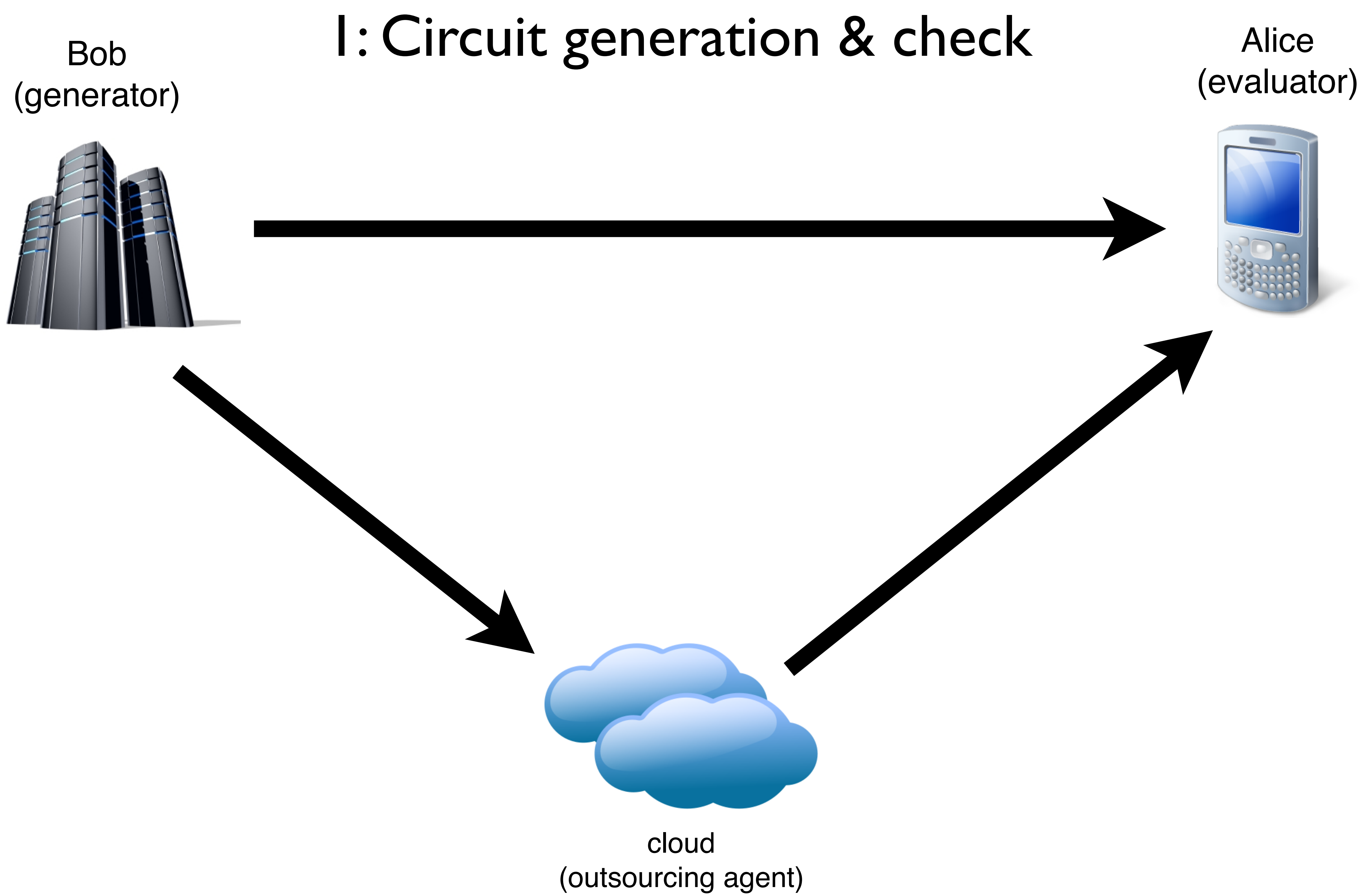
Our construction

- Begin with malicious secure technique developed by Kreuter et al.
- Adapt consistency checks such that Alice and Bob are assured that *all* parties are behaving
- Add "Outsourced Oblivious Transfer" construction to preserve mobile bandwidth

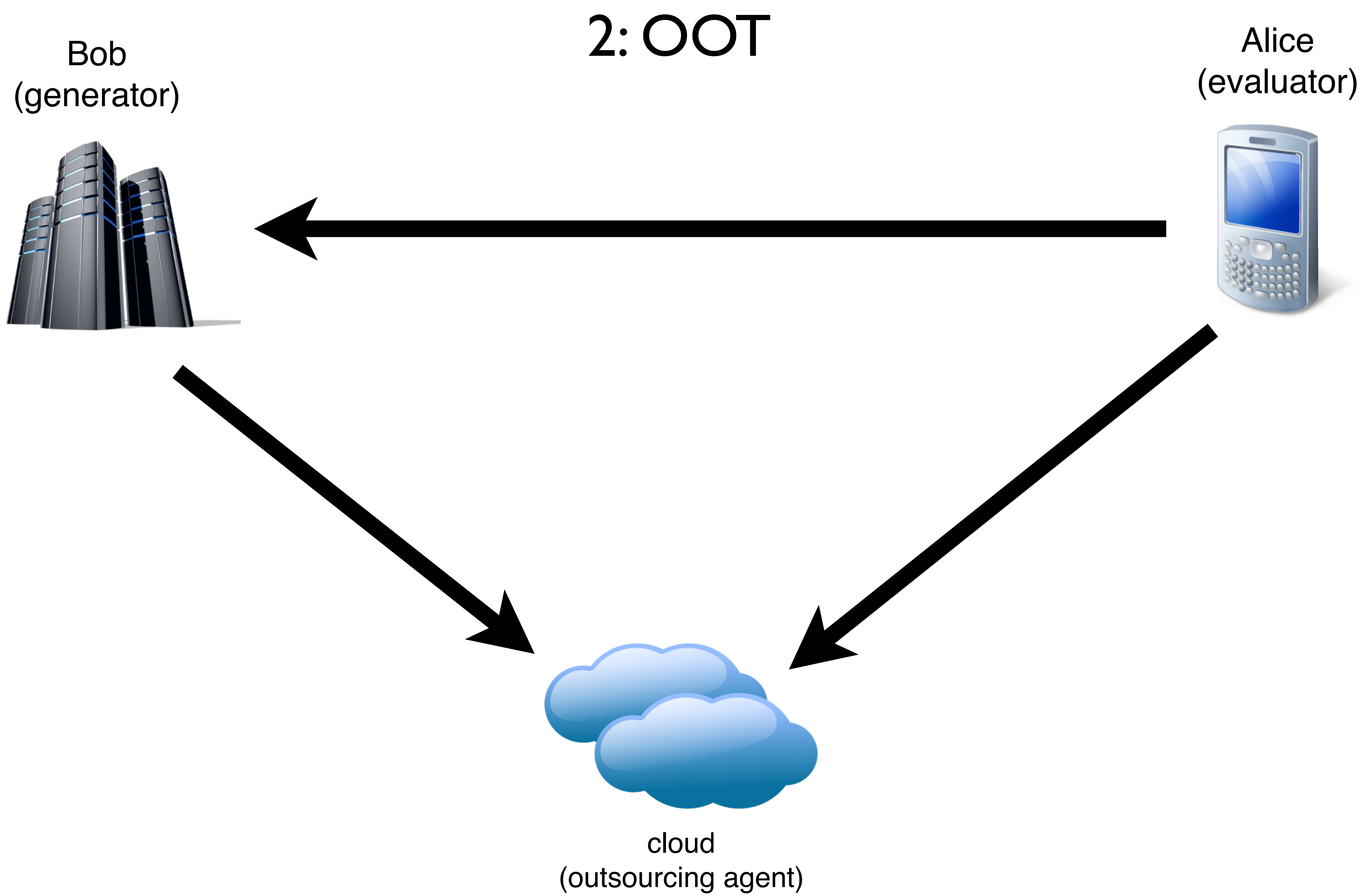
The Protocol

- 5 stages:
  - Circuit construction and validity check
  - OOT
  - Generator input consistency check
  - Circuit evaluation in the cloud
  - Output check and delivery

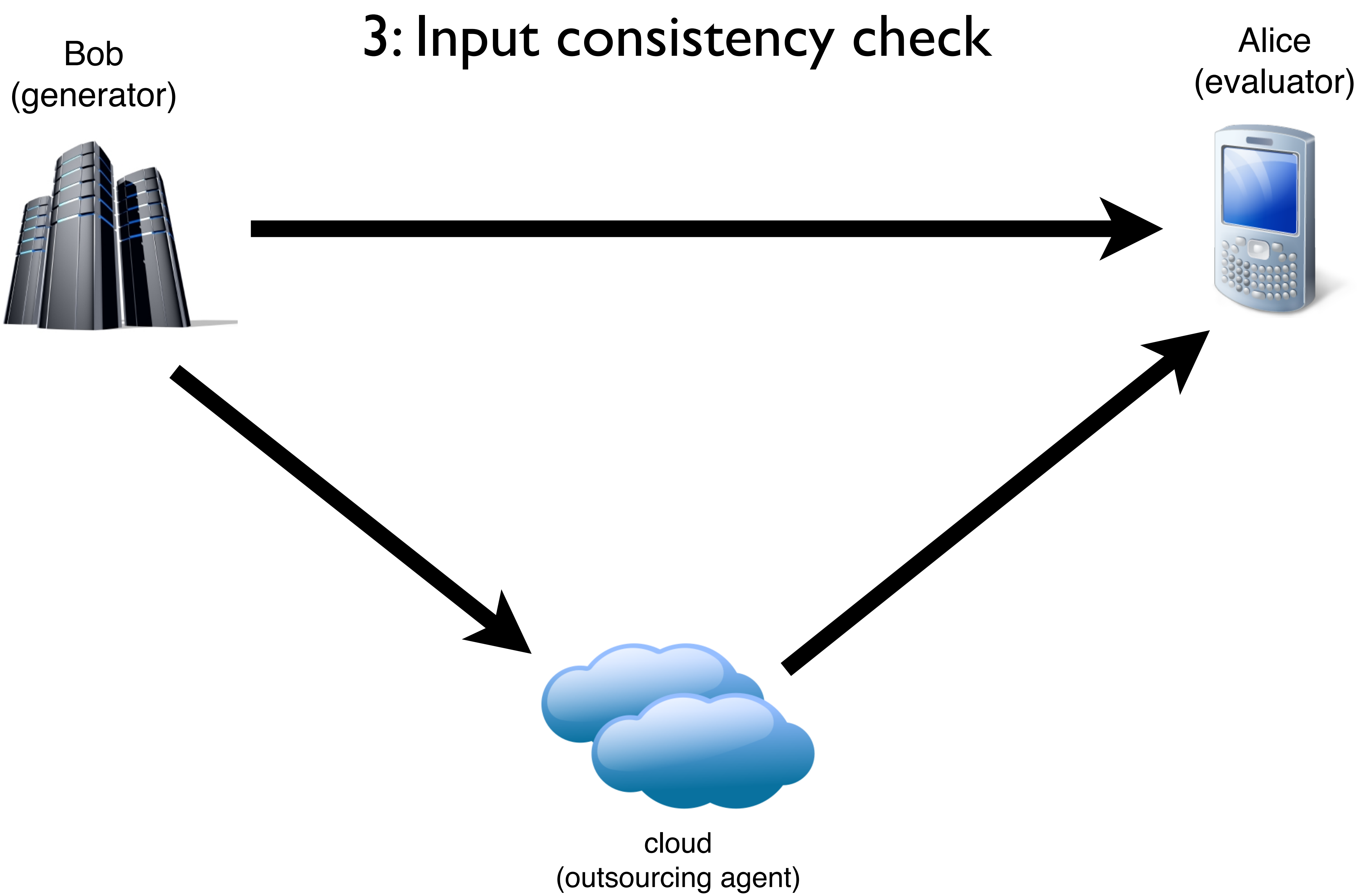
The Protocol



The Protocol

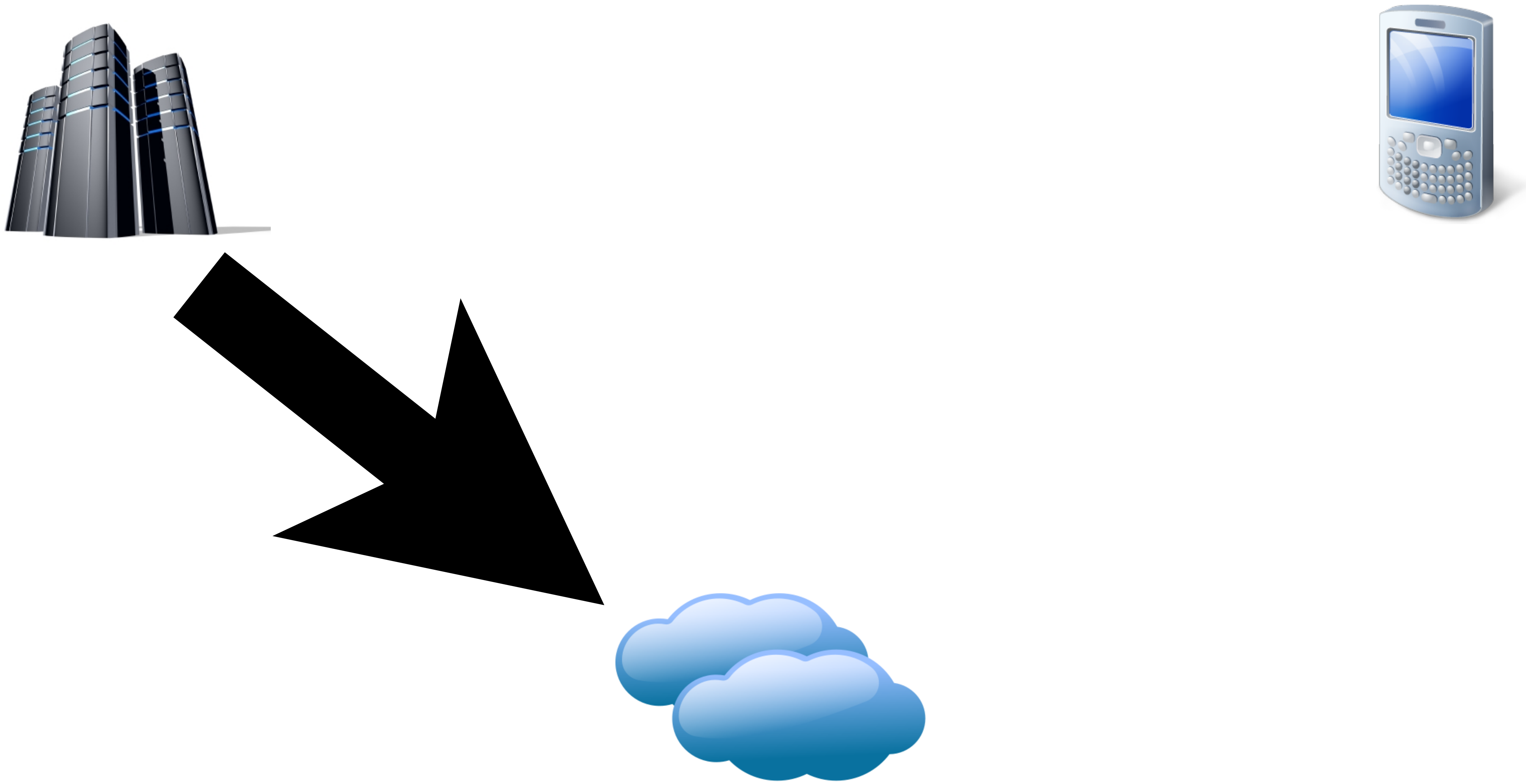


The Protocol



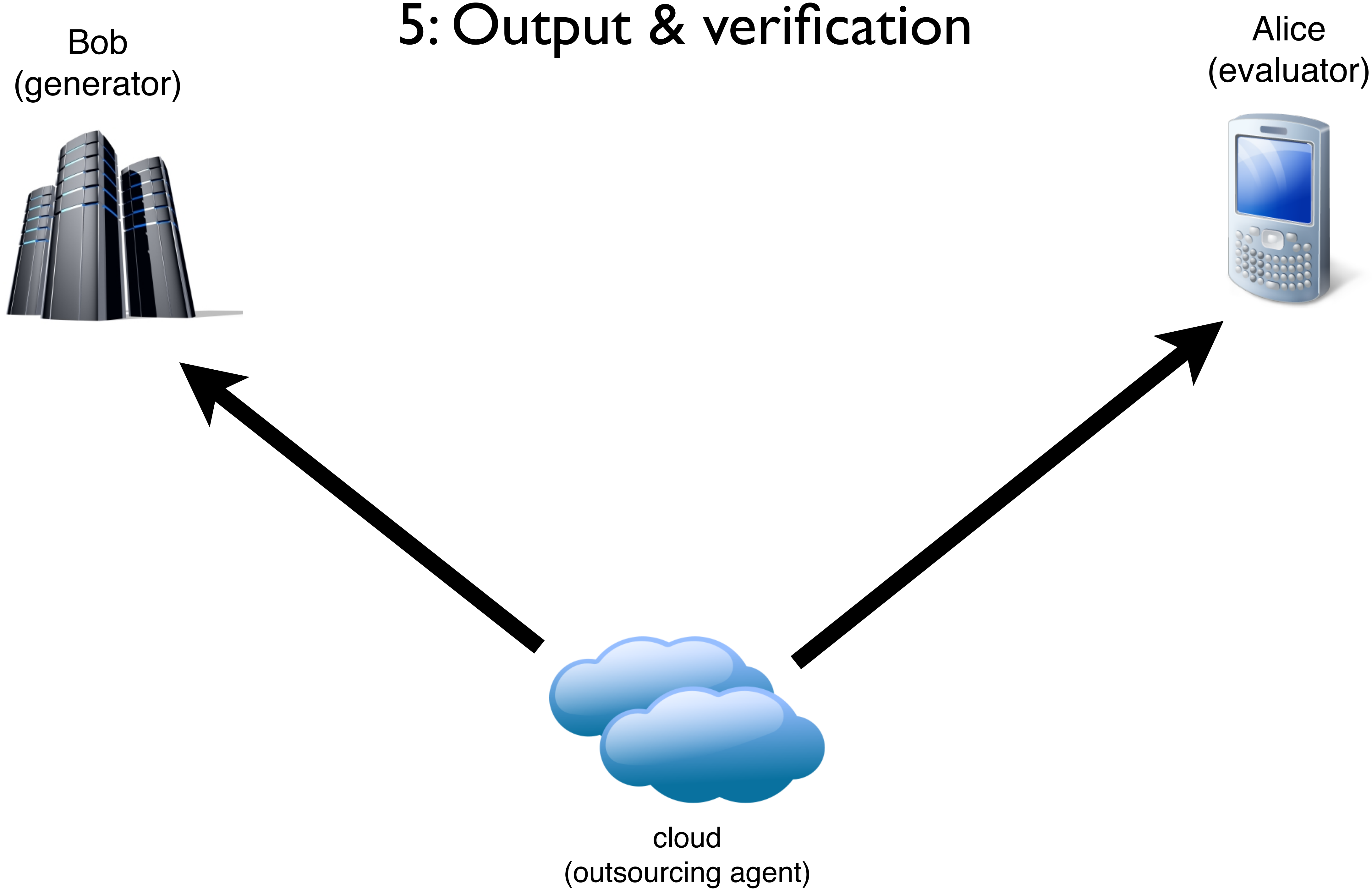
The Protocol

4: Evaluation



The Protocol

5: Output & verification



Security

- We retain all the security checks from Kreuter et al. to preserve security in:
  - Garbled circuits
  - Input consistency between evaluation circuits
  - Output integrity and majority check
  - OOT
- Formal proofs of malicious security in our tech report

Definition 1 *A protocol securely computes a function f if there exists a set of probabilistic polynomial-time (PPT) simulators $\{Sim_i\}_{i \in [3]}$ such that for all PPT adversaries $(A_1, ..., A_3)$, x, z, and for all $i \in [3]$:*

$$\{REAL^{(i)}(k, x; r)\}_{k \in N} \stackrel{c}{\approx} \{IDEAL^{(i)}(k, x; r)\}_{k \in N}$$

Where $S = (S_1, ..., S_3)$, $S_i = Sim_i(A_i)$, and r is random and uniform.

Side note: collusion

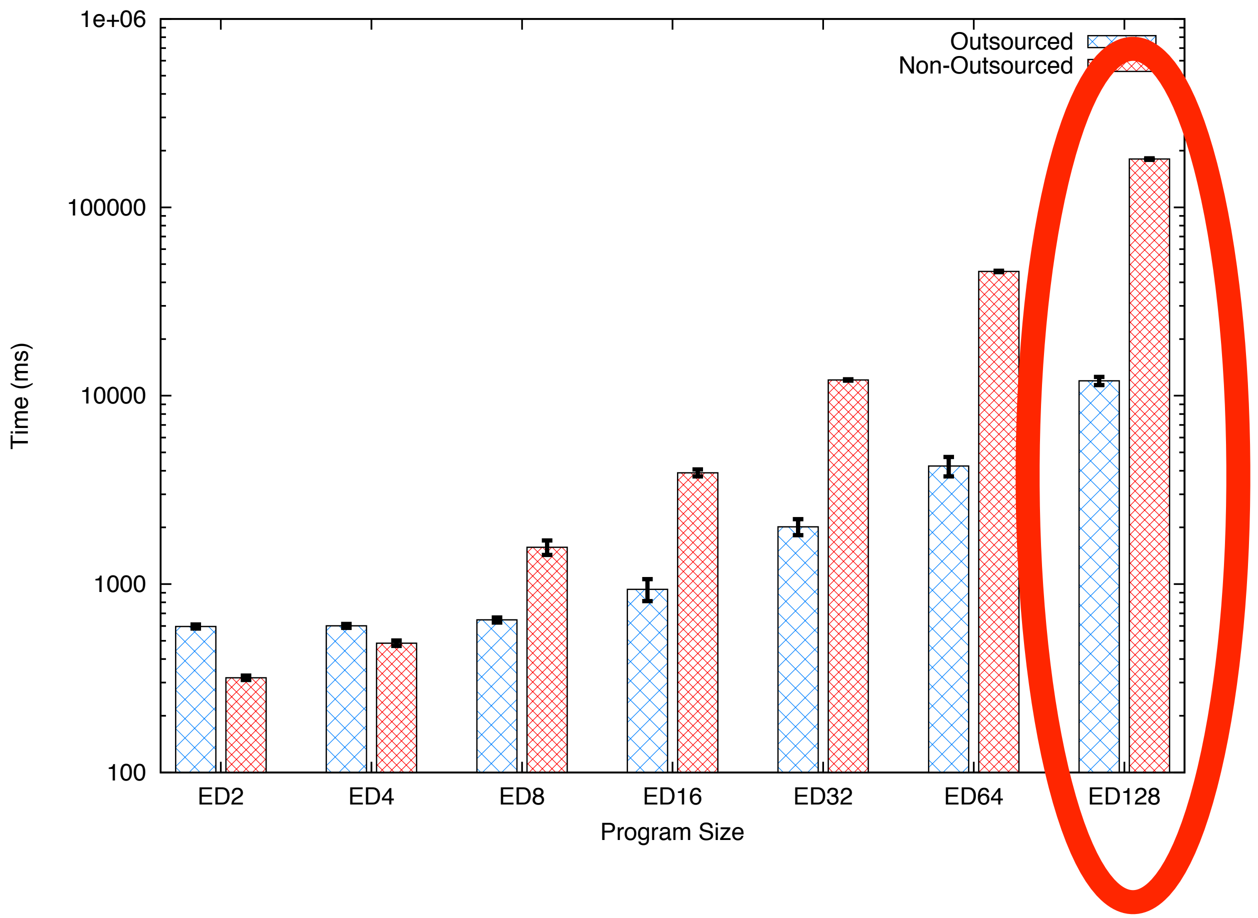
- We prohibit collusion between the cloud and either party
  - Our OT construction breaks if Alice + cloud collude
  - The garbled circuit security breaks if Bob + cloud collude
- Kamara et al. notes that an outsourcing scheme with collusion implies an SFE scheme where one party performs sub-linear work w.r.t. circuit size.
- Realistic scenario: cloud service must preserve security and business reputation

Implementation

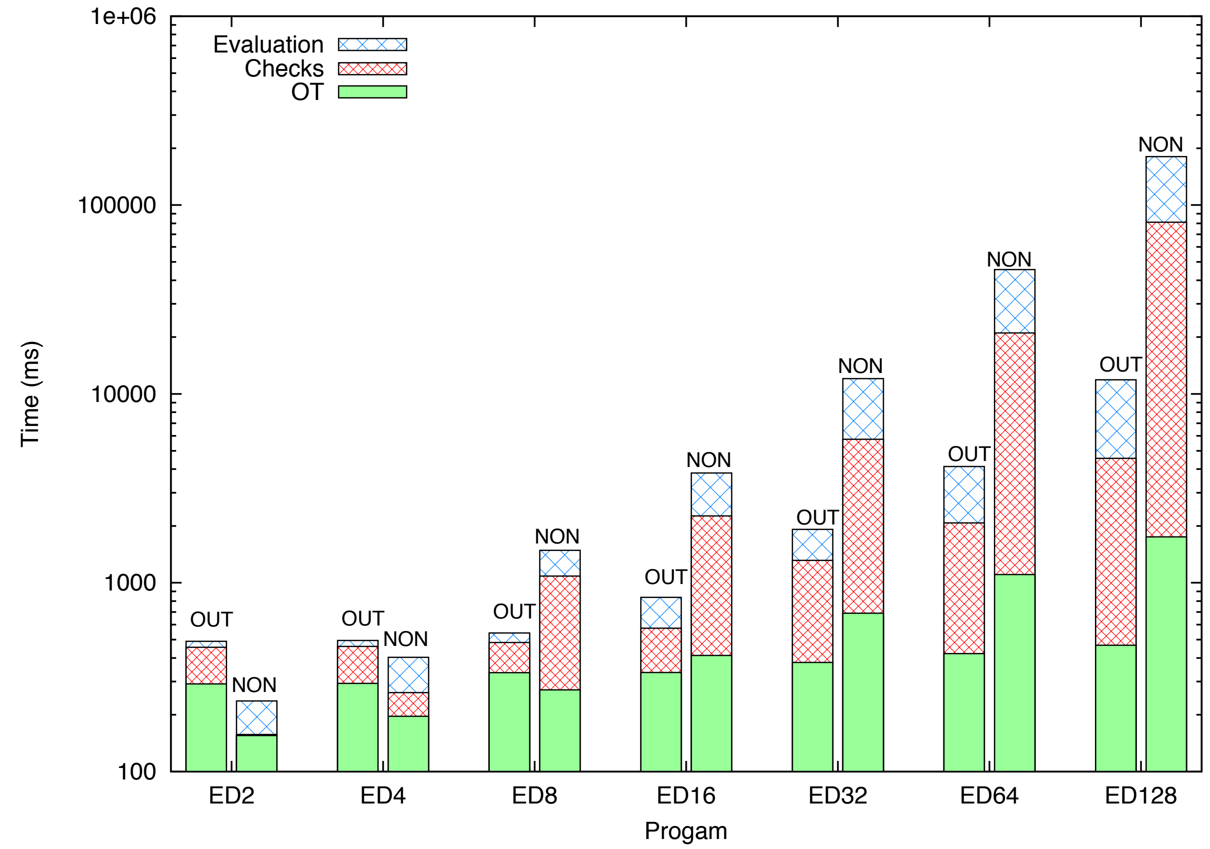
- Testbed
  - Dell R610 servers, dual 6-core Intel Xeon, 32 GB RAM
  - Galaxy Nexus, dual core 1.2 GHz, 1 GB RAM
  - 802.11n (54 Mbps), Internal VLAN (1 Gbps)
- Test apps
  - Millionaire's Problem
  - Edit Distance
  - Set Intersection
  - AES-128

Results: Edit Distance Execution Time

Total runtime



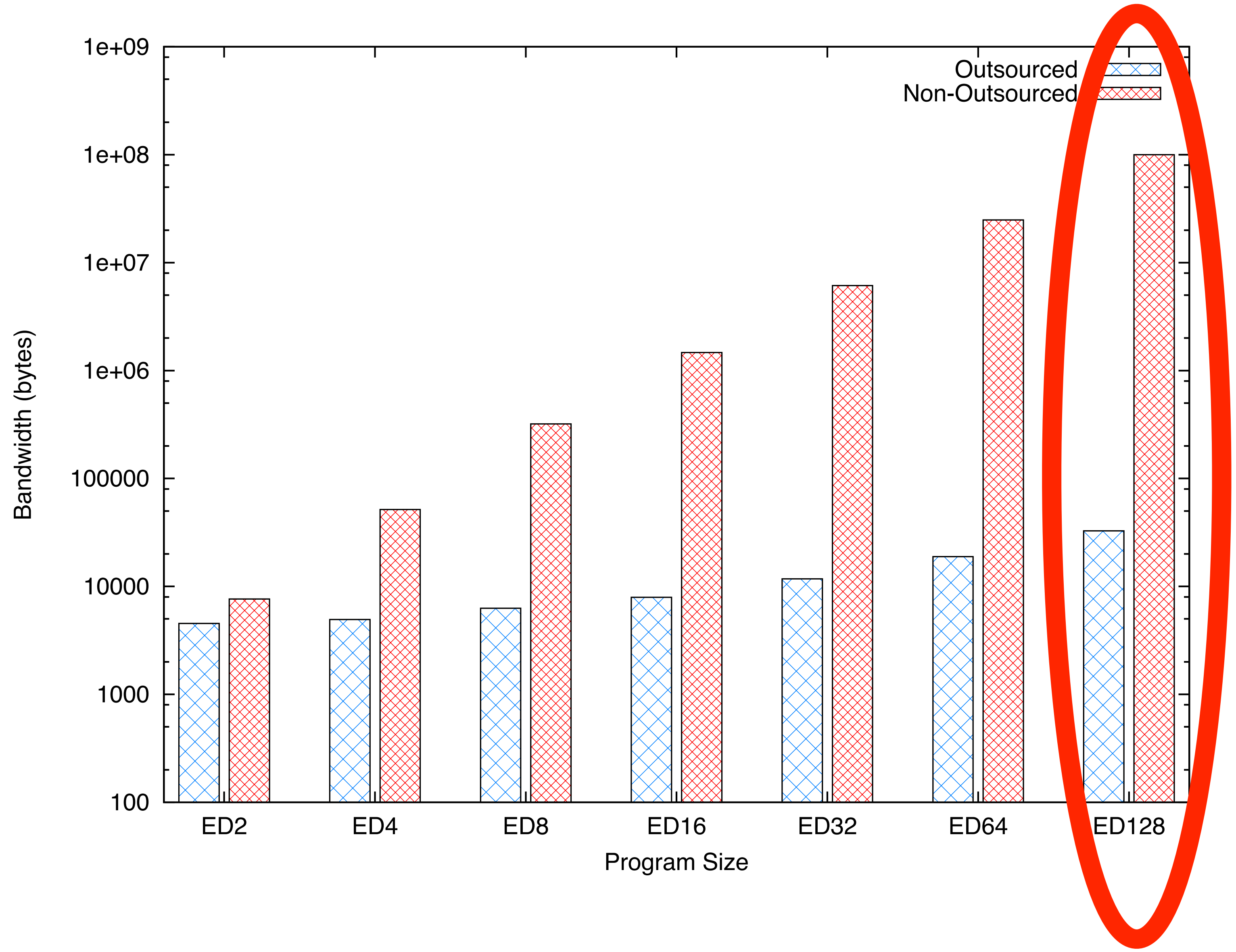
Phase runtimes



98.9% speedup

Results: Edit Distance Bandwidth

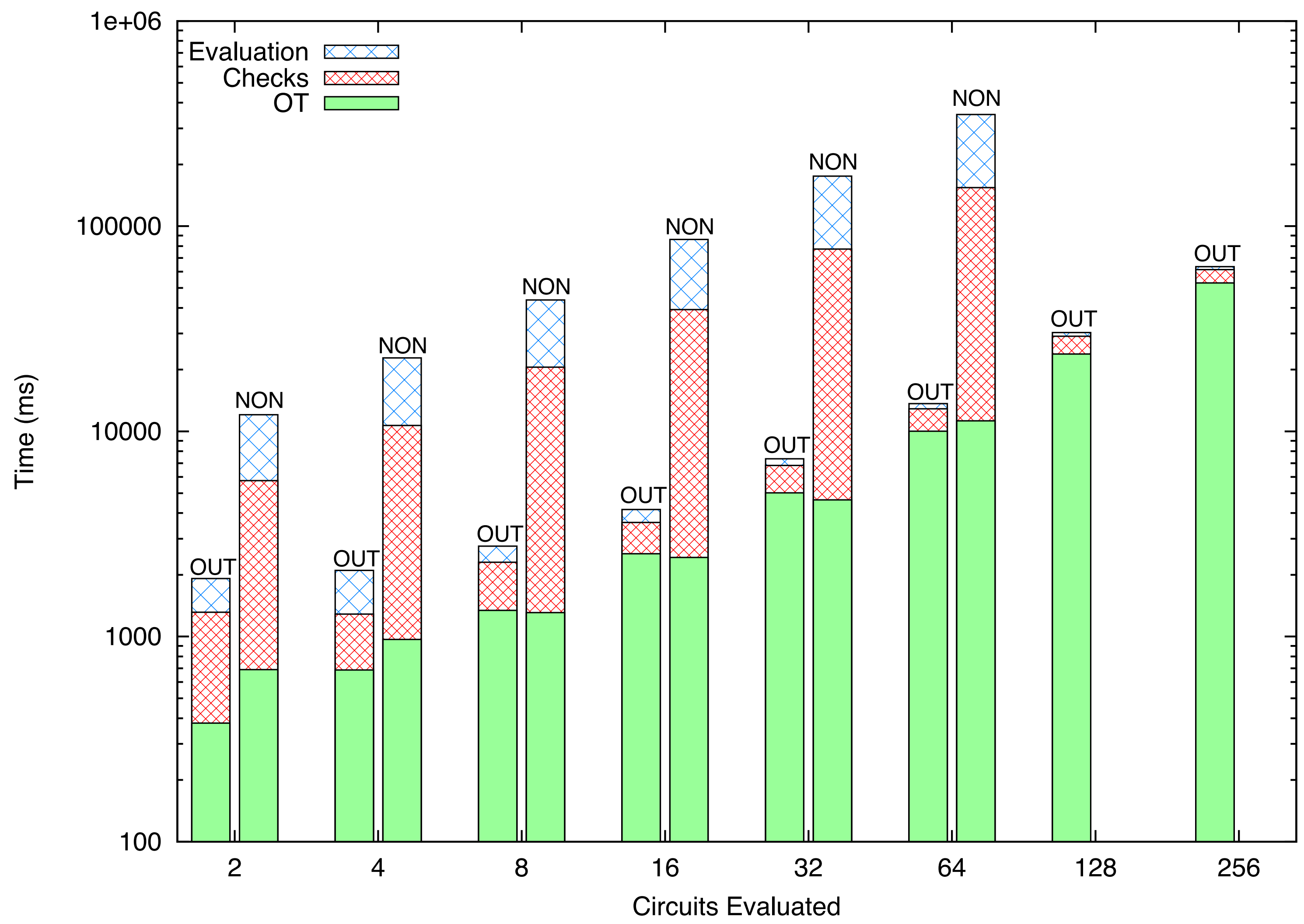
Total bandwidth



3400x reduction

Results: Edit Distance Over Multiple Circuits

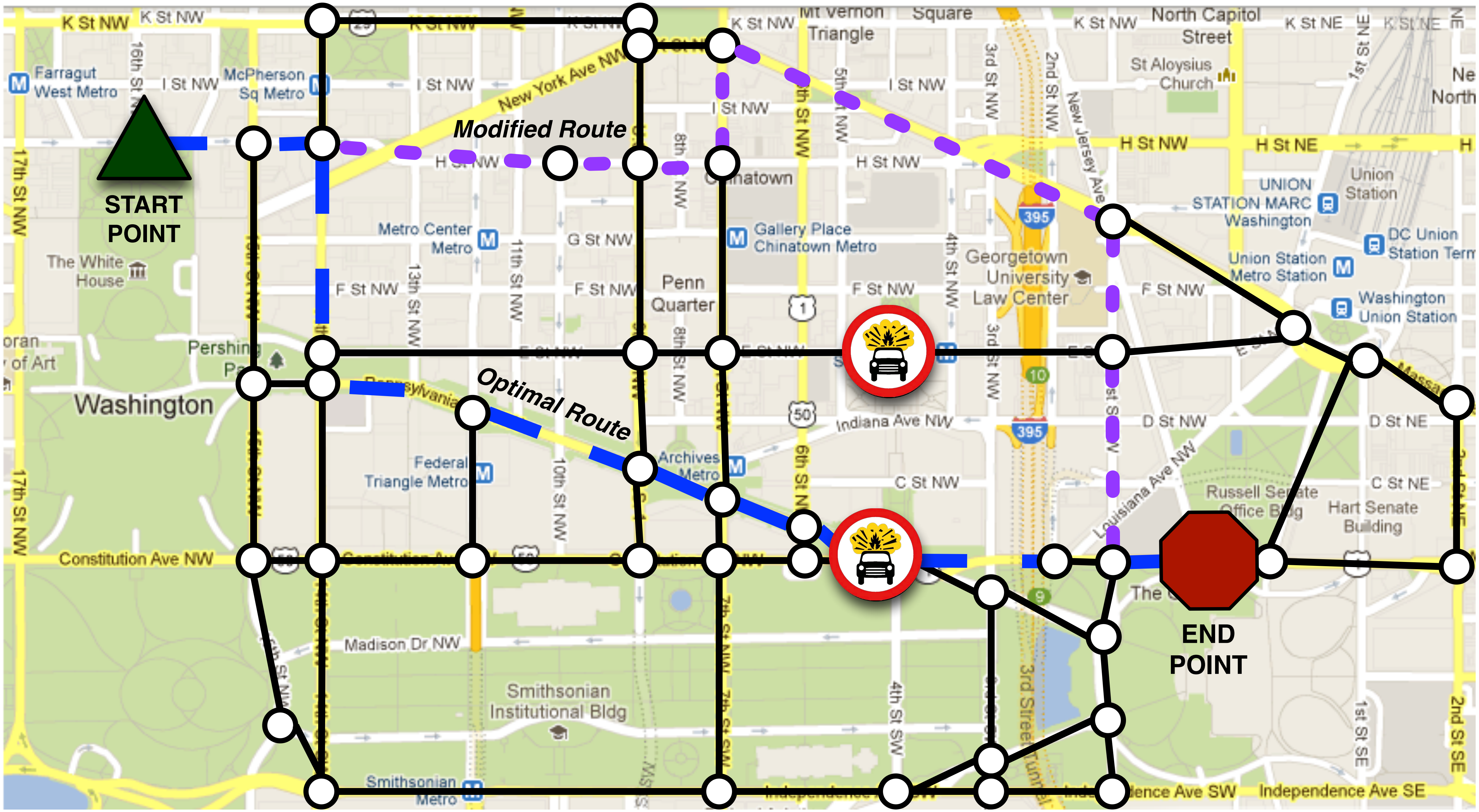
Phase runtimes



Case study: large circuits

- Examined RSA-128 circuit used by Kreuter et al.
- Developed privacy-preserving navigation application
  - Alice inputs a start and end point, Bob inputs outages in a road system. The area map is publicly available
  - The circuit performs Dijkstra's shortest path algorithm to determine the shortest path from start to finish avoiding outages
  - The circuit returns the route to Alice.
  - Considered graphs of 20, 50, 100 nodes
- Testbed: 64 cores, 1 TB memory

Privacy-Preserving Navigation



Case study: results

- Run some of the largest circuits ever publicly evaluated from a mobile device
- Dijkstra's over 100 nodes > 2 billion gates un-optimized
- Evaluation times (128 circuits):
  - 100 nodes ~ 42 minutes
  - 20 nodes ~ 100 seconds

Conclusion

- Costly SMC operations can be outsourced to the cloud securely
- We develop a new OOT protocol to allow outsourced garbled circuit evaluation
- Experimental results show significant performance gains over evaluating directly from the device