SSLShader Accelerating SSL with GPUs

Keon Jang[1], Sangjin Han[1], SeungYeop Han[2], Sue Moon[1] , and KyoungSoo Park[3]
[1]Department of Computer Science, KAIST, {sangjin, keonjang}@an.kaist.ac.kr, sbmoon@kaist.edu
[2]NHN Corporation, haneul0318@gmail.com
[3]Department of Electrical Engineering, KAIST, kyoungsoo@ee.kaist.ac.kr

SSLShader

- SSL-proxy exploiting GPU
  - » Four times faster than CPU-only in terms of TPS
- Offloads cryptographic functions to GPUs
  - » RSA, AES, SHA1
- Opportunistic offloading
  - » Balance loads between CPU and GPU depending on the load
- Implementation
  - » Support TLS1.0 protocol
  - » Support RSA, AES, HMAC-SHA1 cipher suite

Motivation

SSL in today's Internet

- ☺ Secure end-to-end communication
- ☺ Easy to integrate existing applications
- ☺ Popular in security critical web services
- ☹ Consumes huge amount of CPU cycles

General-Purpose Computation on GPUs

- GPUs are widely used for data-intensive workloads
  - » E.g. Medical imaging, bioinformatics, finance, etc.
- High performance with massively-parallel processing

	Price	# of cores	# of HW threads	Peak performance
CPU (Intel Core i7 920)	$260	4	8	43 GFLOPS
GPU (NVIDIA GTX480)	$499	480	23,040	1345 GFLOPS

Design and Implementation

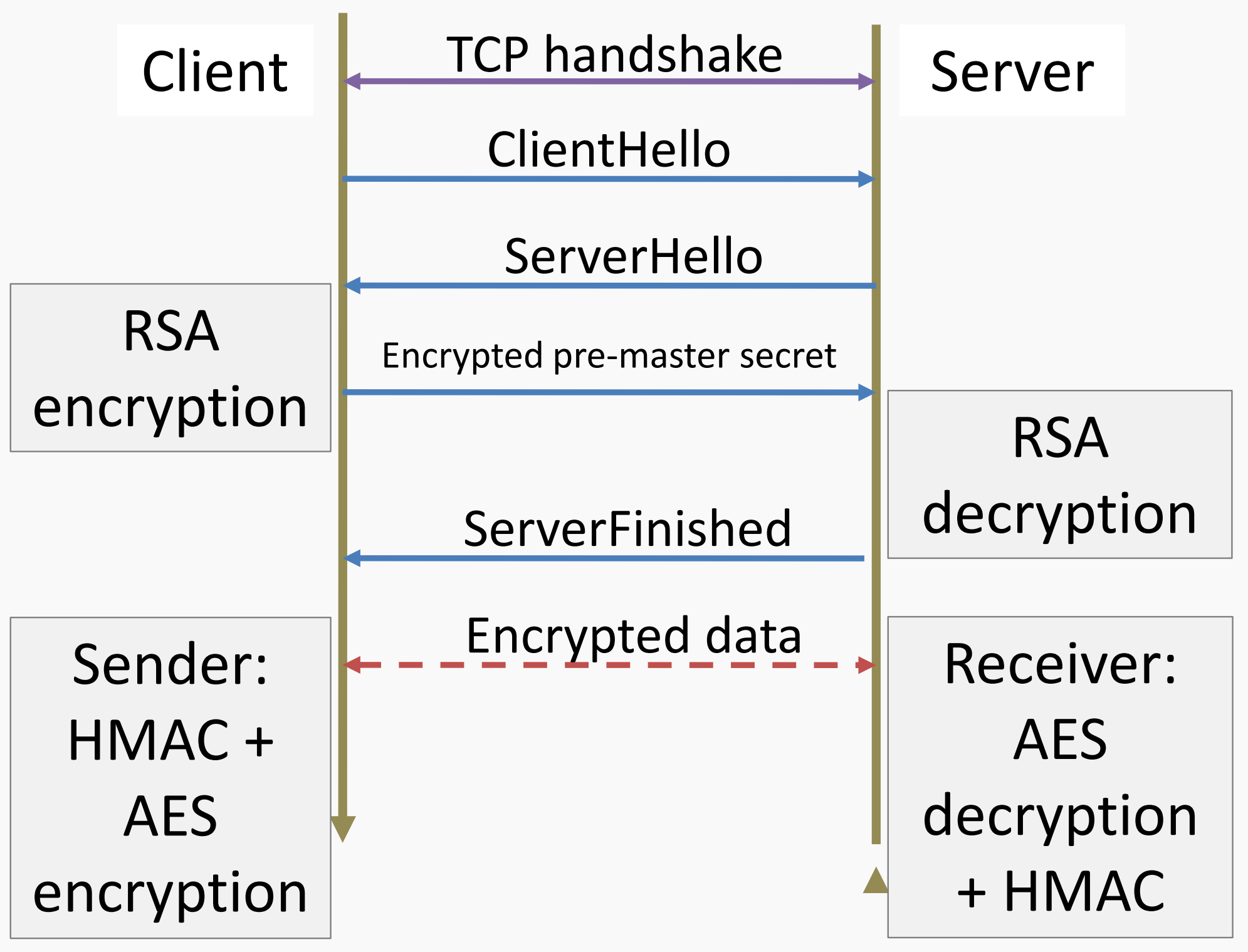
Basic Design

- SSL proxy
  - » No modification on the server
    - » Server uses TCP and proxy tunnels TCP through SSL protocol to client
  - » Many servers behind single proxy
    - » More parallelism with more concurrent connections
    - » Cost-effective in server farms
- Opportunistic Offloading
  - » GPU is not always faster than CPU
    - » GPU requires tens to thousands of same task for max utilization
    - » Single threaded job is slower on GPU
  - » Use GPU only when there's benefit
  - » Minimize latency in light load
  - » More throughput in high load
- NUMA-aware GPU sharing
  - » Scalable with # of CPUs and GPUs
  - » Each core spawns worker thread
  - » GPU is shared by workers in the same Numa-node

Cryptographic Algorithms

- RSA
  - » Secure exchange of secrets under eavesdropping
  - » GPU executes single multiplication of large integer (> 512 bits) in parallel
- AES
  - » Encrypt exchange of data
  - » In CBC-mode, AES-DEC is parallelized in 16-byte block level
- HMAC-SHA1
  - » Prevent tampering of message

Workflow of SSL


Client

Server

TCP handshake

ClientHello

ServerHello

RSA encryption

Encrypted pre-master secret

RSA decryption

ServerFinished

Encrypted data

Sender: HMAC + AES encryption

Receiver: AES decryption + HMAC

Microbenchmarks

- RSA

	1024-bit	2048-bit	4096-bit
GPU	66,970	9,995	1,348
CPU	7,268	1,160	164

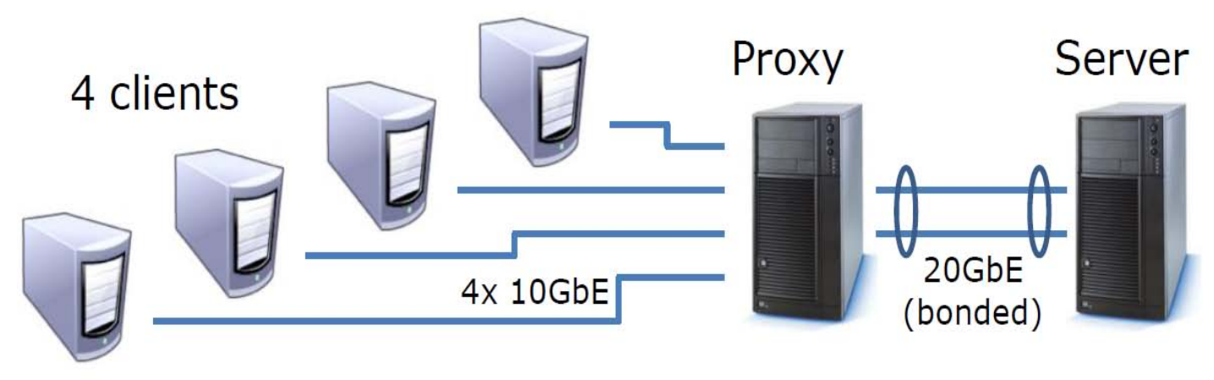
- AES and HMAC-SHA1

	AES-ENC	AES-DEC	HMAC-SHA1
GPU	9,254	9,342	27,863
CPU	4,620	4,620	10,429

RSA unit is msg/s, and AES/HMAC-SHA1 unit is Mbps. GPU is GTX480, and CPU is Intel X5550 (all four cores are used). CPU performance is measured with OpenSSL 1.0.0.

Preliminary Results

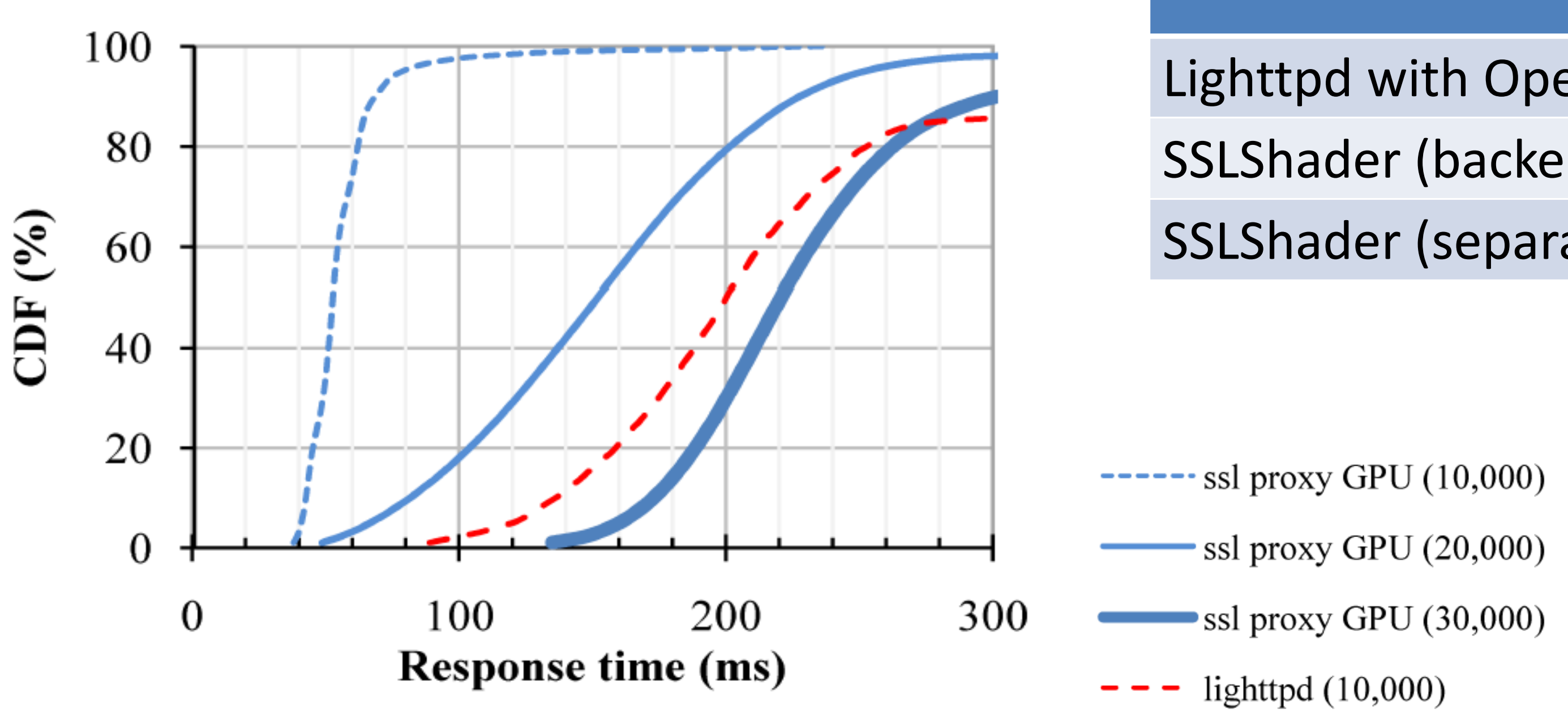
- Experiment Configuration



- Machine Specification

Item	Specification	Qty
CPU	Xeon X5550 (quad-core 2.66GHz)	2
RAM	DDR3 ECC FBDIMM 2GB 1,333Mhz	6
Motherboard	Super Micro X8DAH+	1
Graphics card	NVIDIA GTX480 (480 cores)	2
NIC	Intel X520-DA2 (dual-port 10GbE)	4

- Latency
  - » Measured with 1-byte content size over HTTPS
  - » Number in parenthesis indicates offered load in TPS



- Transactions per seconds
  - » Measured with 1-byte content size over HTTPS

Target	TPS
Lighttpd with OpenSSL (without proxy)	9,246
SSLShader (backend in the same machine)	16,497
SSLShader (separate backend)	25,823

This work was supported by NAP of Korea Research Council of Fundamental Science & Technology (KRCF) and MKE (Ministry of Knowledge Economy of Repbulic of Korea, project no. N02100053).