RotoRouter: Router Support for Endpoint-Authorized Decentralized Traffic Filtering to Prevent DoS Attacks

Albert Kwon [1] [2] Kaiyu Zhang [2] Perk Lun Lim [2] Yu Pan [2]
Jonathan Smith [2] André DeHon [2]

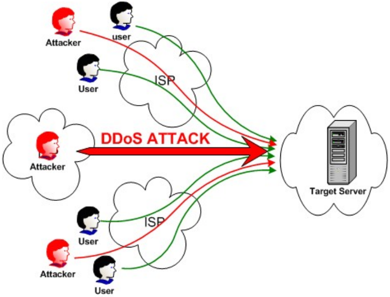
[1] MIT

[2] University of Pennsylvania

December 12, 2014

Denial-of-Service (DoS) Attacks

- Denial-of-service is an attack that makes network or server unavailable
- Overload the network with junk messages so that valid traffic can't make through



DoS in Real Life

- Bank of America, JP Morgan, and Citi (2012)

DoS in Real Life

- Bank of America, JP Morgan, and Citi (2012)
- Bitcoin (Dwolla, Mt. Gox) (2013)

DoS in Real Life

- Bank of America, JP Morgan, and Citi (2012)
- Bitcoin (Dwolla, Mt. Gox) (2013)
- Reddit (2013)

DoS in Real Life

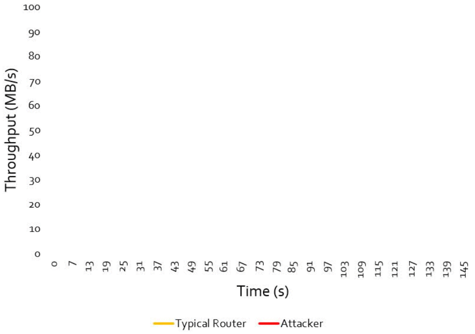
- Bank of America, JP Morgan, and Citi (2012)
- Bitcoin (Dwolla, Mt. Gox) (2013)
- Reddit (2013)
- Sony's Playstation Network (2014)

DoS in Real Life

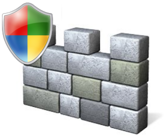
- Bank of America, JP Morgan, and Citi (2012)
- Bitcoin (Dwolla, Mt. Gox) (2013)
- Reddit (2013)
- Sony's Playstation Network (2014)
- DoS costs $240k-$1.2 million in lost revenue/day

DoS in Real Life

- Bank of America, JP Morgan, and Citi (2012)
- Bitcoin (Dwolla, Mt. Gox) (2013)
- Reddit (2013)
- Sony's Playstation Network (2014)
- DoS costs $240k-$1.2 million in lost revenue/day

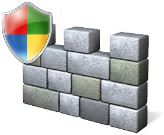


Existing Solutions for DoS



- Software firewalls
  - Non-solution

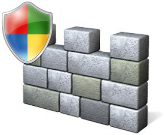
Existing Solutions for DoS



- Software firewalls
  - Non-solution

- Hardware firewalls
  - Inflexible

Existing Solutions for DoS



- Software firewalls
  - Non-solution

- Hardware firewalls
  - Inflexible

- Replication
  - Expensive

RotoRouter Idea

Routers cooperate to only route desired traffic

RotoRouter Idea

Routers cooperate to only route desired traffic

- End points add metadata to packets

RotoRouter Idea

Routers cooperate to only route desired traffic

- End points add metadata to packets
- Routers validate all traffic going through

RotoRouter Idea

Routers cooperate to only route desired traffic

- End points add metadata to packets
- Routers validate all traffic going through
- Enable the end points to “program” the routers
    - Similar to OpenFlow, but decentralized

RotoRouter Idea

Routers cooperate to only route desired traffic

- End points add metadata to packets
- Routers validate all traffic going through
- Enable the end points to “program” the routers
  - Similar to OpenFlow, but decentralized
- Both protocol change and hardware support

Outline

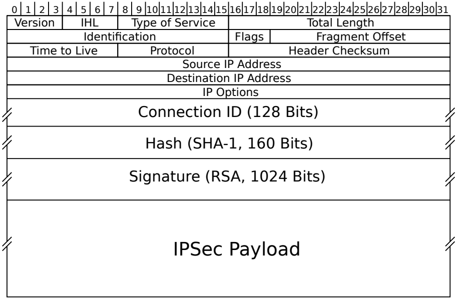
1. Motivation
2. RotoRouter Network Protocol
3. RotoRouter Architecture
4. RotoRouter Evaluation
5. Conclusion

Outline

1. Motivation
2. RotoRouter Network Protocol
3. RotoRouter Architecture
4. RotoRouter Evaluation
5. Conclusion

Network Protocol

- Extend TCP/IP



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment Offset

Time to Live | Protocol | Header Checksum

Source IP Address

Destination IP Address

IP Options

Connection ID (128 Bits)

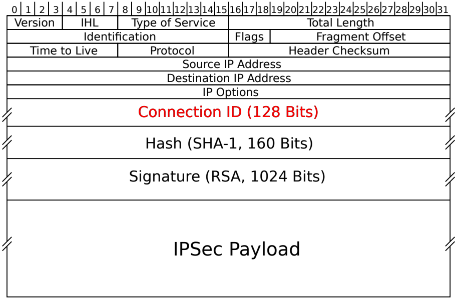
Hash (SHA-1, 160 Bits)

Signature (RSA, 1024 Bits)

IPSec Payload

Network Protocol

- Extend TCP/IP
- Connection ID: *flow*
    - IPv4 source + destination, and random number



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment Offset

Time to Live | Protocol | Header Checksum

Source IP Address

Destination IP Address

IP Options

Connection ID (128 Bits)

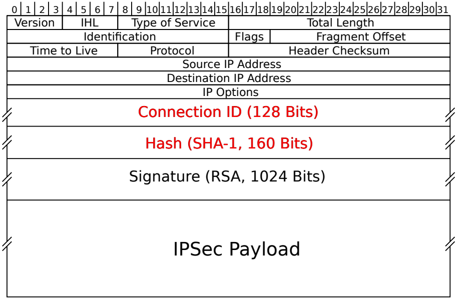
Hash (SHA-1, 160 Bits)

Signature (RSA, 1024 Bits)

IPSec Payload

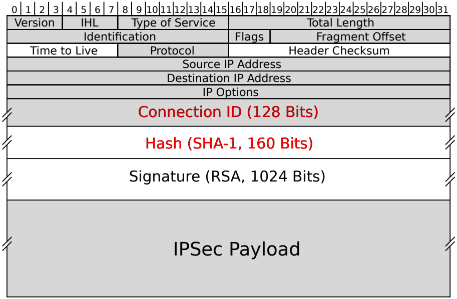
Network Protocol

- Extend TCP/IP
- Connection ID: *flow*
  - IPv4 source + destination, and random number
- Hash
  - Prevents tampering



Network Protocol

- Extend TCP/IP
- Connection ID: *flow*
    - IPv4 source + destination, and random number
- Hash
    - Prevents tampering



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment Offset

Time to Live | Protocol | Header Checksum

Source IP Address

Destination IP Address

IP Options

Connection ID (128 Bits)

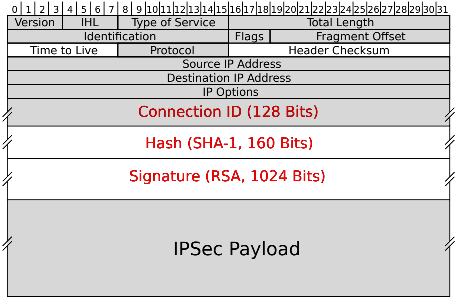
Hash (SHA-1, 160 Bits)

Signature (RSA, 1024 Bits)

IPSec Payload

Network Protocol

- Extend TCP/IP
- Connection ID: *flow*
  - IPv4 source + destination, and random number
- Hash
  - Prevents tampering
- Public key signature
  - Prevents spoofing
  - Assume that public keys of end points are distributed



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment Offset

Time to Live | Protocol | Header Checksum

Source IP Address

Destination IP Address

IP Options

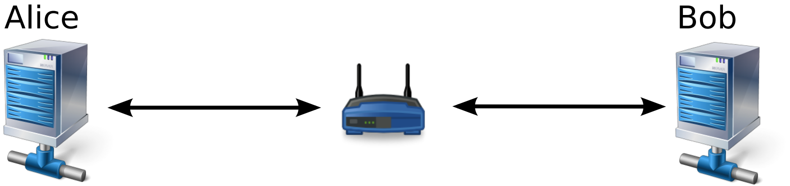
Connection ID (128 Bits)

Hash (SHA-1, 160 Bits)

Signature (RSA, 1024 Bits)

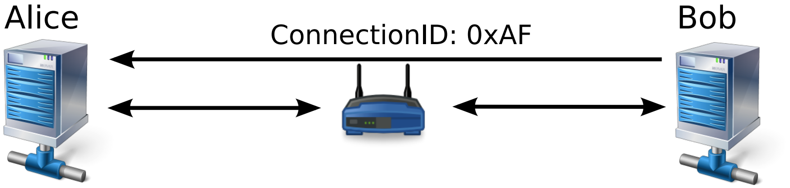
IPSec Payload

Router Enforceable Protocol: Setup



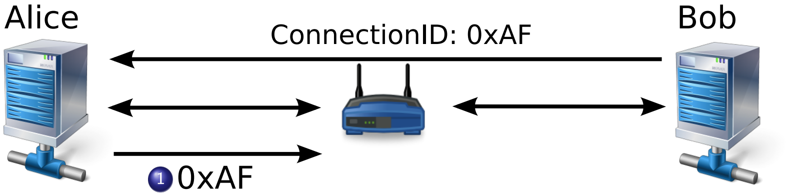
Router Enforceable Protocol: Setup

- Receiving end point sends:



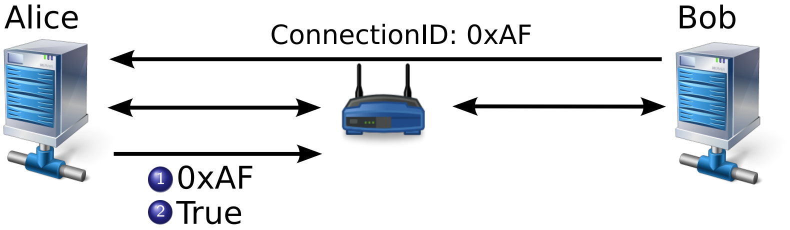
Router Enforceable Protocol: Setup

- Receiving end point sends:
  1. ConnectionID corresponding to the flow



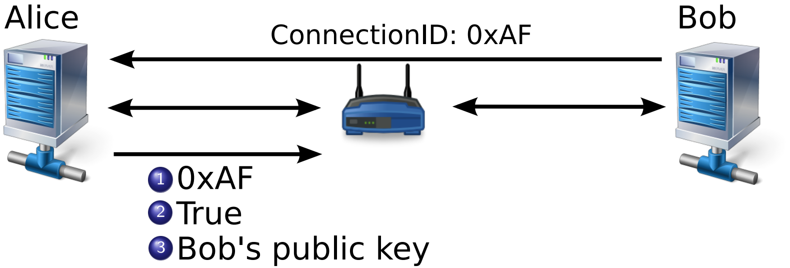
Router Enforceable Protocol: Setup

- Receiving end point sends:
  1. ConnectionID corresponding to the flow
  2. Boolean indicating if flow is desired or not



Router Enforceable Protocol: Setup

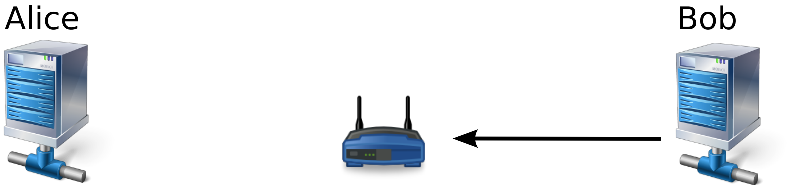
- Receiving end point sends:
  1. ConnectionID corresponding to the flow
  2. Boolean indicating if flow is desired or not
  3. Source node's public key



Router Enforceable Protocol: Filter

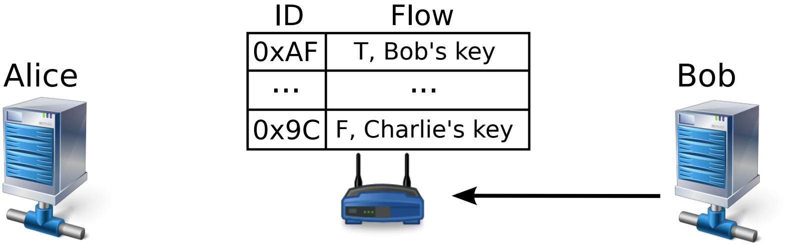
Router Enforceable Protocol: Filter

- Router performs:



Router Enforceable Protocol: Filter

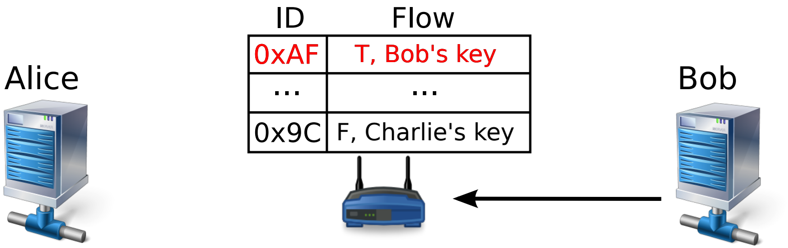
- Router performs:



ID	Flow
0xAF	T, Bob's key
...	...
0x9C	F, Charlie's key

Router Enforceable Protocol: Filter

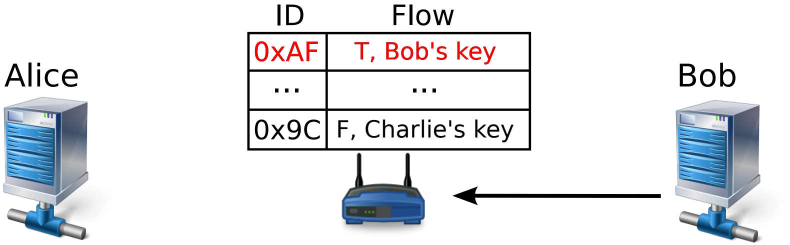
- Router performs:
  1. Look up connection ID



ID	Flow
0xAF	T, Bob's key
...	...
0x9C	F, Charlie's key

Router Enforceable Protocol: Filter

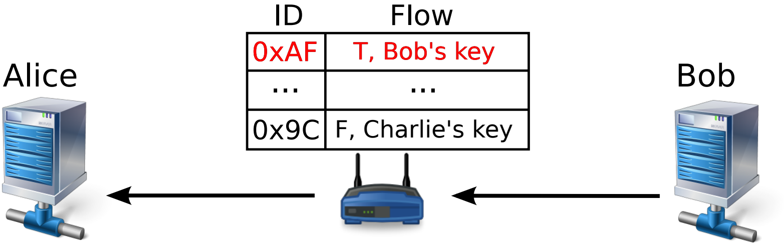
- Router performs:
  1. Look up connection ID
  2. Verify the hash of the packet
  3. Verify the signature with the public key



ID	Flow
0xAF	T, Bob's key
...	...
0x9C	F, Charlie's key

Router Enforceable Protocol: Filter

- Router performs:
  1. Look up connection ID
  2. Verify the hash of the packet
  3. Verify the signature with the public key
  4. Drop or relay the packet

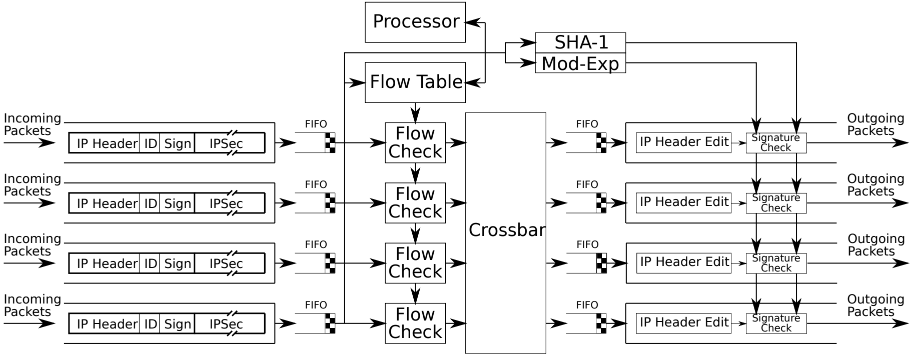


ID	Flow
0xAF	T, Bob's key
...	...
0x9C	F, Charlie's key

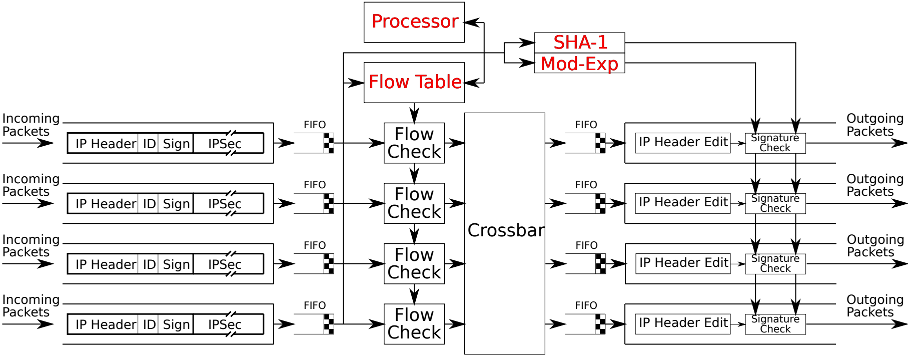
Outline

1. Motivation
2. RotoRouter Network Protocol
3. RotoRouter Architecture
4. RotoRouter Evaluation
5. Conclusion

RotoRouter Architecture

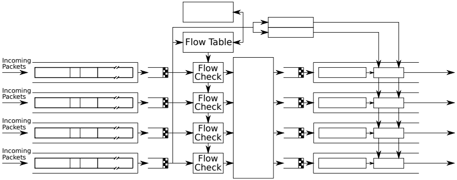


RotoRouter Architecture



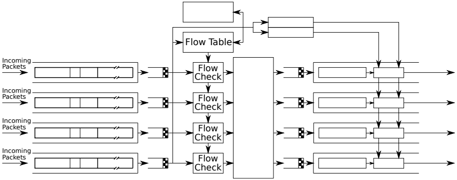
Flow Table

- Dictionary mapping connection ID to source public key, and a valid flow boolean



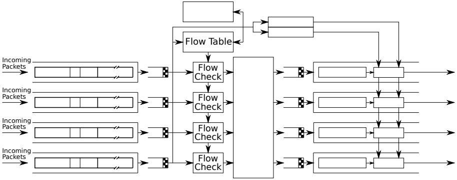
Flow Table

- Dictionary mapping connection ID to source public key, and a valid flow boolean
- Small cache (on BRAM) backed by larger memory
    - *Negative flows* are cached as well



Flow Table

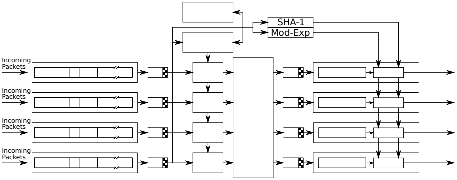
- Dictionary mapping connection ID to source public key, and a valid flow boolean
- Small cache (on BRAM) backed by larger memory
  - *Negative flows* are cached as well
- Crucial for router performance
  - (Near) Associative memory [1]



[1] Udit Dhawan and André DeHon. Area-Efficient Near-Associative Memories on FPGAs. FPGA 2013

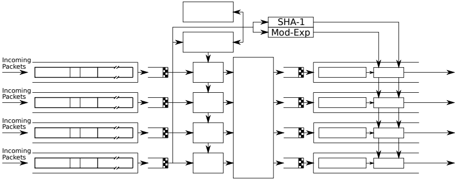
Crypto Modules

- Cryptographic hash and signature verification



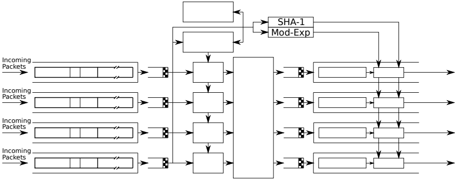
Crypto Modules

- Cryptographic hash and signature verification
  - Currently: SHA-1 for hash, and RSA for signature



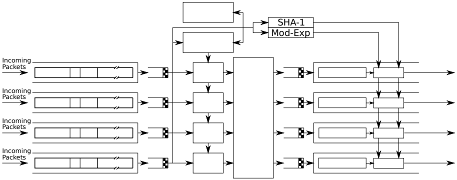
Crypto Modules

- Cryptographic hash and signature verification
  - Currently: SHA-1 for hash, and RSA for signature
- Crucial for router performance



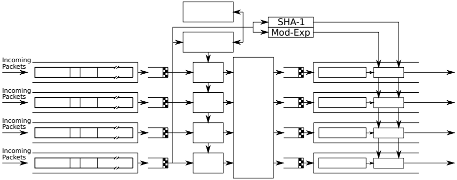
Crypto Modules

- Cryptographic hash and signature verification
  - Currently: SHA-1 for hash, and RSA for signature
- Crucial for router performance
  - Large exponentiation ⇒ No line-rate public key signature



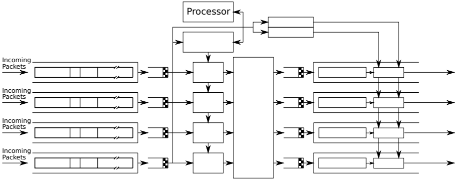
Crypto Modules

- Cryptographic hash and signature verification
  - Currently: SHA-1 for hash, and RSA for signature
- Crucial for router performance
  - Large exponentiation ⇒ No line-rate public key signature
  - Okay to use small exponent for verification



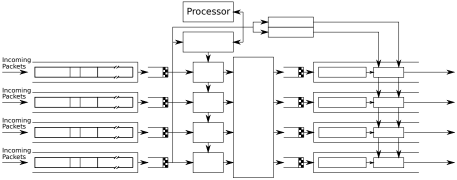
On-chip Processor

- Communicates with the end points to setup new flows



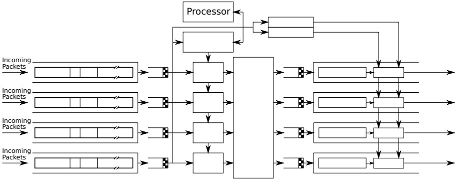
On-chip Processor

- Communicates with the end points to setup new flows
  - Only impacts initial latency



On-chip Processor

- Communicates with the end points to setup new flows
    - Only impacts initial latency
- Manages the flow table entries



Outline

1 Motivation

2 RotoRouter Network Protocol

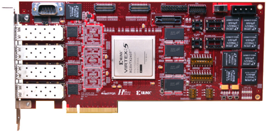
3 RotoRouter Architecture

4 RotoRouter Evaluation

5 Conclusion

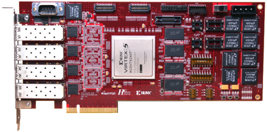
Prototype Implementation

- Hardware prototype on NetFPGA-10G platform



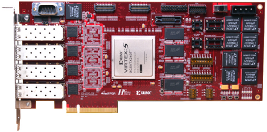
Prototype Implementation

- Hardware prototype on NetFPGA-10G platform
  - Xilinx Virtex 5 (xc5vtx240tffg1759-2) using 65nm technology



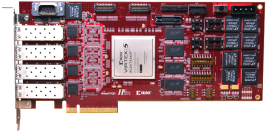
Prototype Implementation

- Hardware prototype on NetFPGA-10G platform
    - Xilinx Virtex 5 (xc5vtx240tffg1759-2) using 65nm technology
- Supports four 1 Gbps ports



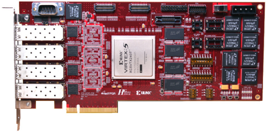
Prototype Implementation

- Hardware prototype on NetFPGA-10G platform
    - Xilinx Virtex 5 (xc5vtx240tffg1759-2) using 65nm technology
- Supports four 1 Gbps ports
- Implemented using Bluespec System Verilog, and open source libraries



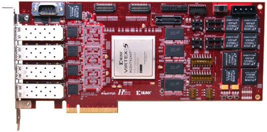
Prototype Implementation

- Hardware prototype on NetFPGA-10G platform
  - Xilinx Virtex 5 (xc5vtx240tffg1759-2) using 65nm technology
- Supports four 1 Gbps ports
- Implemented using Bluespec System Verilog, and open source libraries
  - Bluespec: Processor, flow table, crossbar, mod-exp



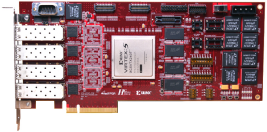
Prototype Implementation

- Hardware prototype on NetFPGA-10G platform
  - Xilinx Virtex 5 (xc5vtx240tffg1759-2) using 65nm technology
- Supports four 1 Gbps ports
- Implemented using Bluespec System Verilog, and open source libraries
  - Bluespec: Processor, flow table, crossbar, mod-exp
  - OpenCore: SHA-1



Prototype Implementation

- Hardware prototype on NetFPGA-10G platform
  - Xilinx Virtex 5 (xc5vtx240tffg1759-2) using 65nm technology
- Supports four 1 Gbps ports
- Implemented using Bluespec System Verilog, and open source libraries
  - Bluespec: Processor, flow table, crossbar, mod-exp
  - OpenCore: SHA-1
  - NetFPGA-10G library: Gigabit ethernet, PCIe, etc



Implementation

	Area		Clock
Module	LUTs	BRAMs	(MHz)
Crossbar w/ Buffers	8249	16	300
Flow Table	38	74	350
Processor	26985	52	200
SHA-1 Module	4×1005	0	125
Mod-Exp	73591	0	200
RotoRouter	112883	142	125
IPv4 Router	22523	35	150
Total available	149760	324	-

Implementation

Module	Area		Clock
	LUTs	BRAMs	(MHz)
Crossbar w/ Buffers	8249	16	300
Flow Table	38	74	350
Processor	26985	52	200
SHA-1 Module	4×1005	0	125
Mod-Exp	73591	0	200
RotoRouter	112883	142	125
IPv4 Router	22523	35	150
Total available	149760	324	-

Implementation

Module	Area		Clock
	LUTs	BRAMs	(MHz)
Crossbar w/ Buffers	8249	16	300
Flow Table	38	74	350
Processor	26985	52	200
SHA-1 Module	4×1005	0	125
Mod-Exp	73591	0	200
RotoRouter	112883	142	125
IPv4 Router	22523	35	150
Total available	149760	324	-

Implementation

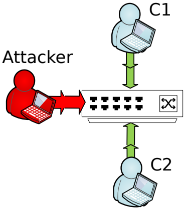
Module	Area		Clock
	LUTs	BRAMs	(MHz)
Crossbar w/ Buffers	8249	16	300
Flow Table	38	74	350
Processor	26985	52	200
SHA-1 Module	4×1005	0	125
Mod-Exp	73591	0	200
RotoRouter	112883	142	125
IPv4 Router	22523	35	150
Total available	149760	324	-

Implementation

	Area		Clock
Module	LUTs	BRAMs	(MHz)
Crossbar w/ Buffers	8249	16	300
Flow Table	38	74	350
Processor	26985	52	200
SHA-1 Module	4×1005	0	125
Mod-Exp	73591	0	200
RotoRouter	112883	142	125
IPv4 Router	22523	35	150
Total available	149760	324	-

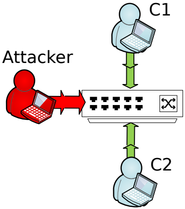
Goodput Measurement

- 3 machines with 1 Gbps Ethernet



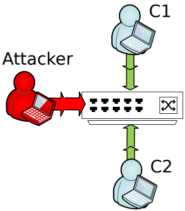
Goodput Measurement

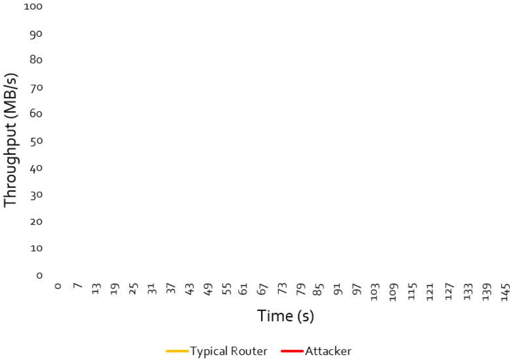
- 3 machines with 1 Gbps Ethernet
- One attacker flooding the network, while the other two saturate network bandwidth



Goodput Measurement

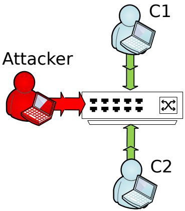
- 3 machines with 1 Gbps Ethernet
- One attacker flooding the network, while the other two saturate network bandwidth

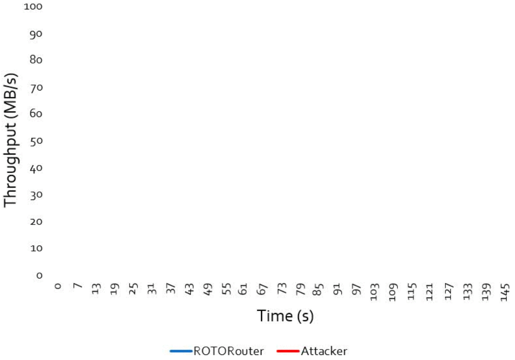




Goodput Measurement

- 3 machines with 1 Gbps Ethernet
- One attacker flooding the network, while the other two saturate network bandwidth





Scalability

- Want to support 10, or even 100, Gbps ports

Scalability

- Want to support 10, or even 100, Gbps ports

	Crossbar	Flow Table	SHA-1	Mod-Exp
Clock Speed (MHz)	300	350	125	200
Individual Throughput (Gbps)	19.2	515	4×0.8	4×1.2
Effective Throughput @ 125 MHz (Gbps)	8	184	3.2	4.8

Scalability

- Want to support 10, or even 100, Gbps ports
- Newer FPGAs support high speeding switching (> 160 Gbps) [2]

	Crossbar	Flow Table	SHA-1	Mod-Exp
Clock Speed (MHz)	300	350	125	200
Individual Throughput (Gbps)	19.2	515	4×0.8	4×1.2
Effective Throughput @ 125 MHz (Gbps)	8	184	3.2	4.8

[2] Z. Dai and J. Zhu. Saturating the transceiver bandwidth: Switch fabric design on FPGAs, FPGA 2012

Scalability

- Want to support 10, or even 100, Gbps ports
- Newer FPGAs support high speeding switching ($>$ 160 Gbps) [2]
- Crypto could be replicated
  - Hash and signature primitives could be switched to faster primitives (e.g., eliptical curve)

	Crossbar	Flow Table	SHA-1	Mod-Exp
Clock Speed (MHz)	300	350	125	200
Individual Throughput (Gbps)	19.2	515	4×0.8	4×1.2
Effective Throughput @ 125 MHz (Gbps)	8	184	3.2	4.8

[2] Z. Dai and J. Zhu. Saturating the transceiver bandwidth: Switch fabric design on FPGAs, FPGA 2012

Outline

1 Motivation

2 RotoRouter Network Protocol

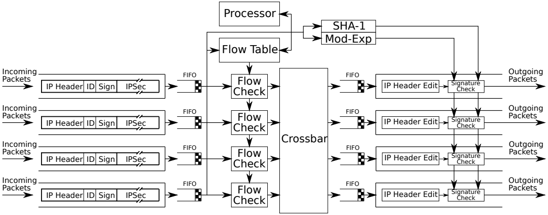
3 RotoRouter Architecture

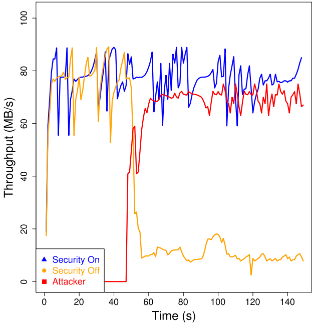
4 RotoRouter Evaluation

5 Conclusion

Conclusion

- Router assisted DoS protection shows great promise
    - Line-rate public key verification is possible!
- Proof-of-concept router demonstrates low-overhead
- Software and hardware co-design leads to better solutions





Thanks!

Future Work

- Characterizing dynamic behaviors
    - Flow setup, router setup, etc
- Throughput impact on larger scale systems
- Incremental deployment