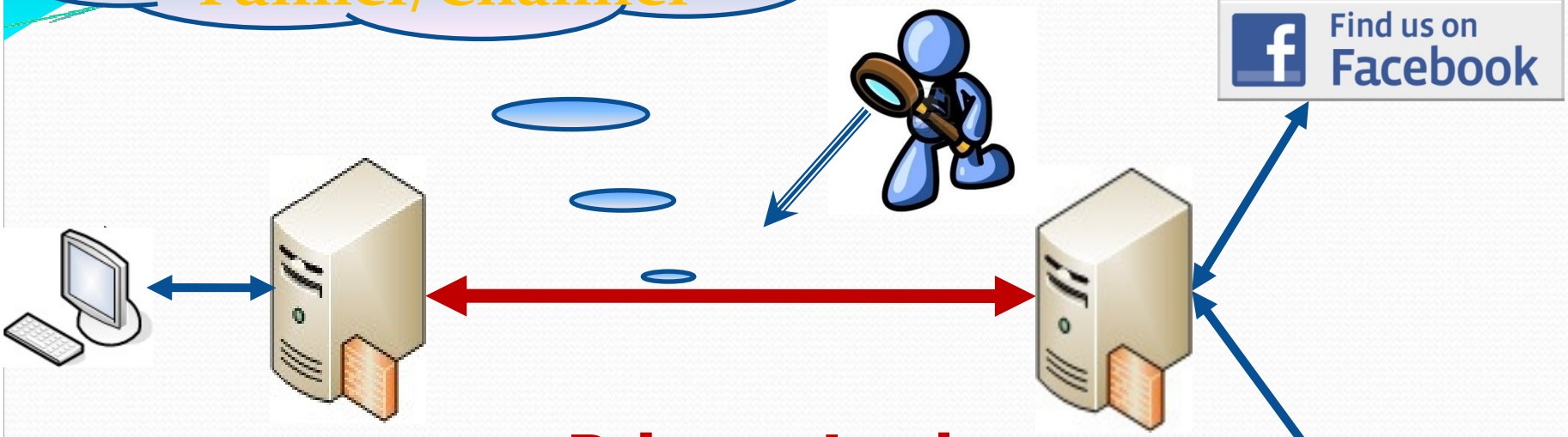
HTTPOS:
Sealing Information Leaks with Browser-side Obfuscation of Encrypted Flows

Daniel Xiapu Luo § , *Brent Peng Zhou* § , Edmond W. W. Chan §
Wenke Lee†, Rocky K. C. Chang § , Roberto Perdisci‡

The Hong Kong Polytechnic University §
Georgia Institute of Technology†
University of Georgia‡

2011 02

Encrypted Tunnel/Channel



Privacy Leakage

Major Security Threat to Web Applications

Content

Motivation

Encryption is not enough to prevent Privacy Leakage

To avoid privacy leakage (e.g., padding at the sever side)

HTTPOS

Methods with better scalability and flexibility Browser-side solution

Challenges and Contributions

Challenges in a browser-side solution:

- Can't modify the server's behavior directly
- Encrypted tunnels at different layers have different features
- Performance degradation

HTTPOS Contributions:

- ✓Provide a comprehensive and configurable suite of traffic transformation techniques
- ✓Protect privacy for four popular scenarios
- ✓Reduce performance degradation

Content

Motivation

Threat Model

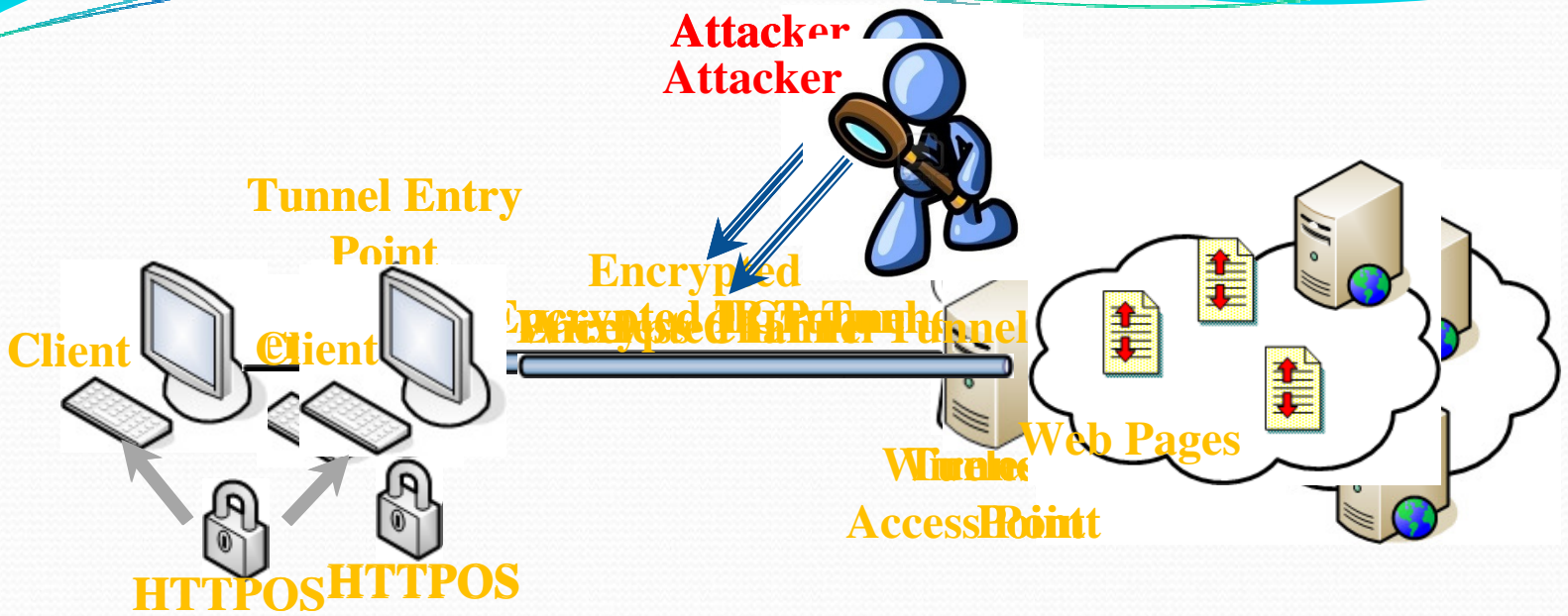
HTTPOS Design

Implementation

Evaluation

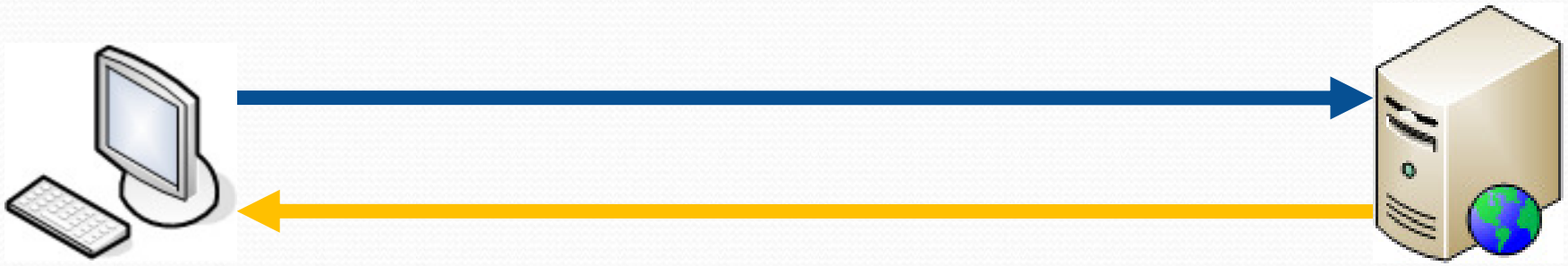
Conclusion

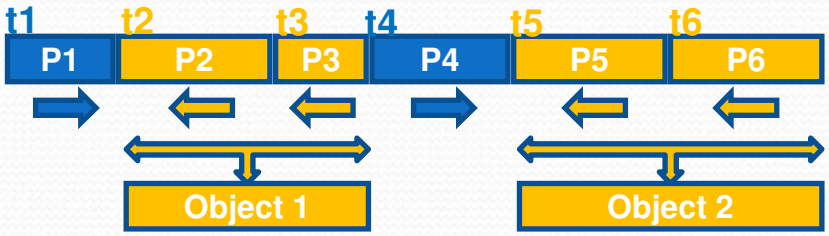
Attack Scenarios



	Attacker's goal	Visibility of HTTP header	Visibility of TCP header	Visibility of Destination IP	HTTPOS's location
Wireless	Web Site	✖	✖	✖	Client
IP Tunnel	Web Site	✖	✖	✖	Client/Tunnel Entry
TCP Tunnel	Web Site	✖	✓	✖	Client/Tunnel Entry
HTTPS	Web Page	✖	✓	✓	Client

Targeted Traffic Analysis Attacks





Attack Name	Features	Methods
SSWRPQ (SP'02)	The number and size of web objects	Jaccard Coefficient
BLJL (PET'05)	Inter-arrival time between packets and packet size	Cross Correlation
LL-JC (CCS'06)	Tuples of (flow direction, packet size)	Jaccard Coefficient
LL-NBC (CCS'06)	Tuples of (flow direction, packet size)	Naïve Bayesian
CWWZ (SP'10)	Sequence of tuples (flow direction, packet size)	Sequence Comparison

Content

Motivation

Threat Model

HTTPOS Design

Implementation

Evaluation

Conclusion

Two Defence Strategies in HTTPOS

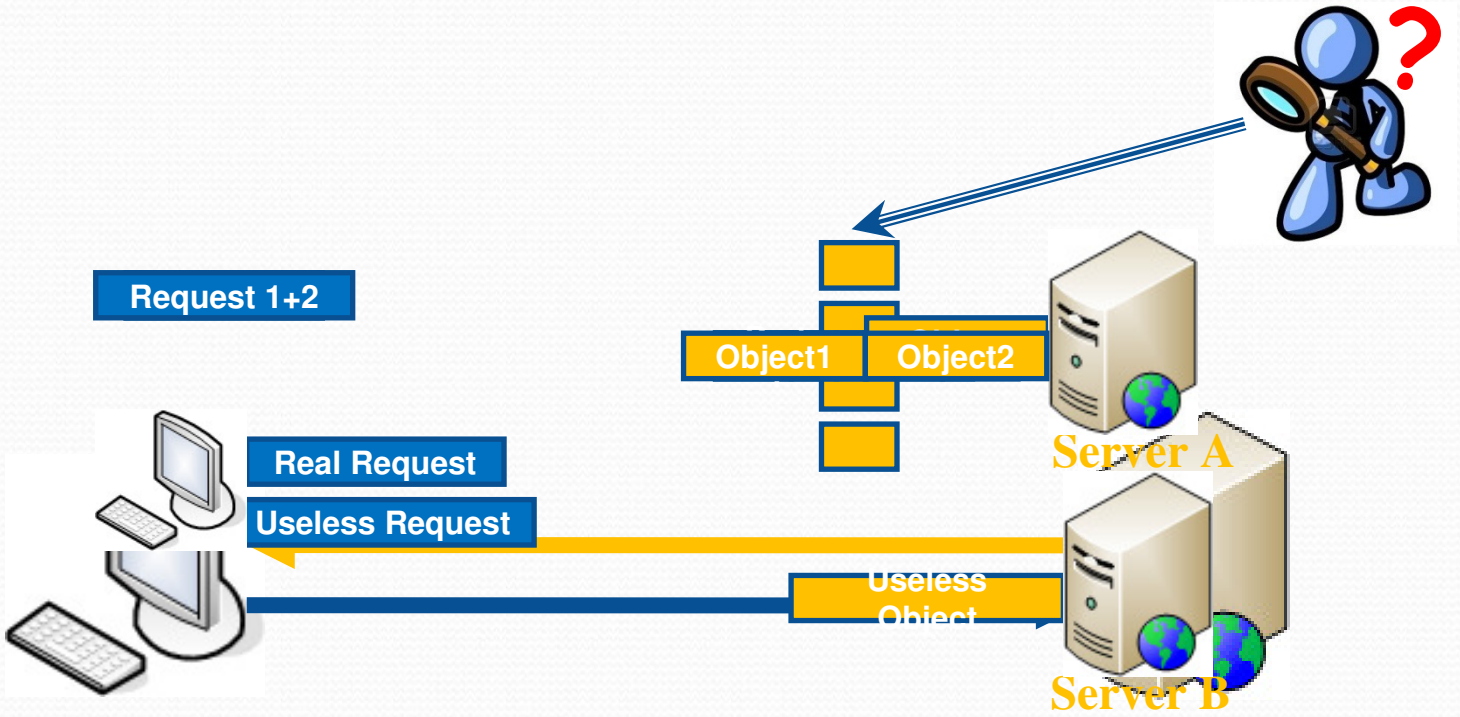
Diffusion Strategy:

- Generate features that never appear in the training data set

Confusion Strategy:

- Make features in flow A similar to those in flow B

Basic Methods in HTTPOS

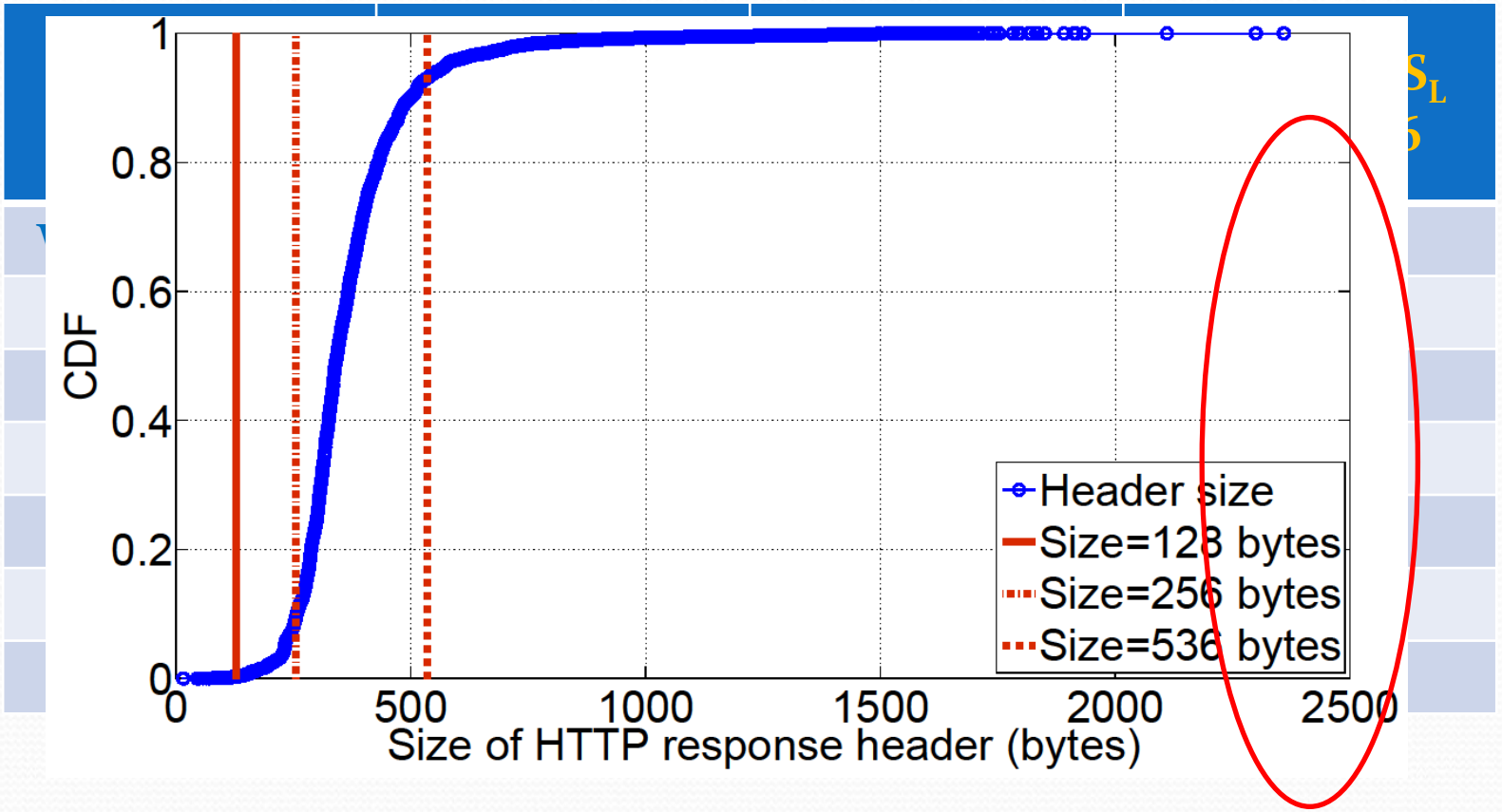


Protocol	Method	Effectiveness
TCP	MSS	Packet Size etc.
TCP	ADWIN	Packet Size etc.
HTTP	Range	Packet Size, Object Size etc.
HTTP	Pipelining	Packet Size, Object Size etc.
HTTP	Useless Request	Packet Size, Object Size etc.

TCP Features Measurement Result

Top 2,000 Web Sites from www.Alexa.com

143,333 URLs from 8,845 Web Servers



HTTP Features Measurement Result

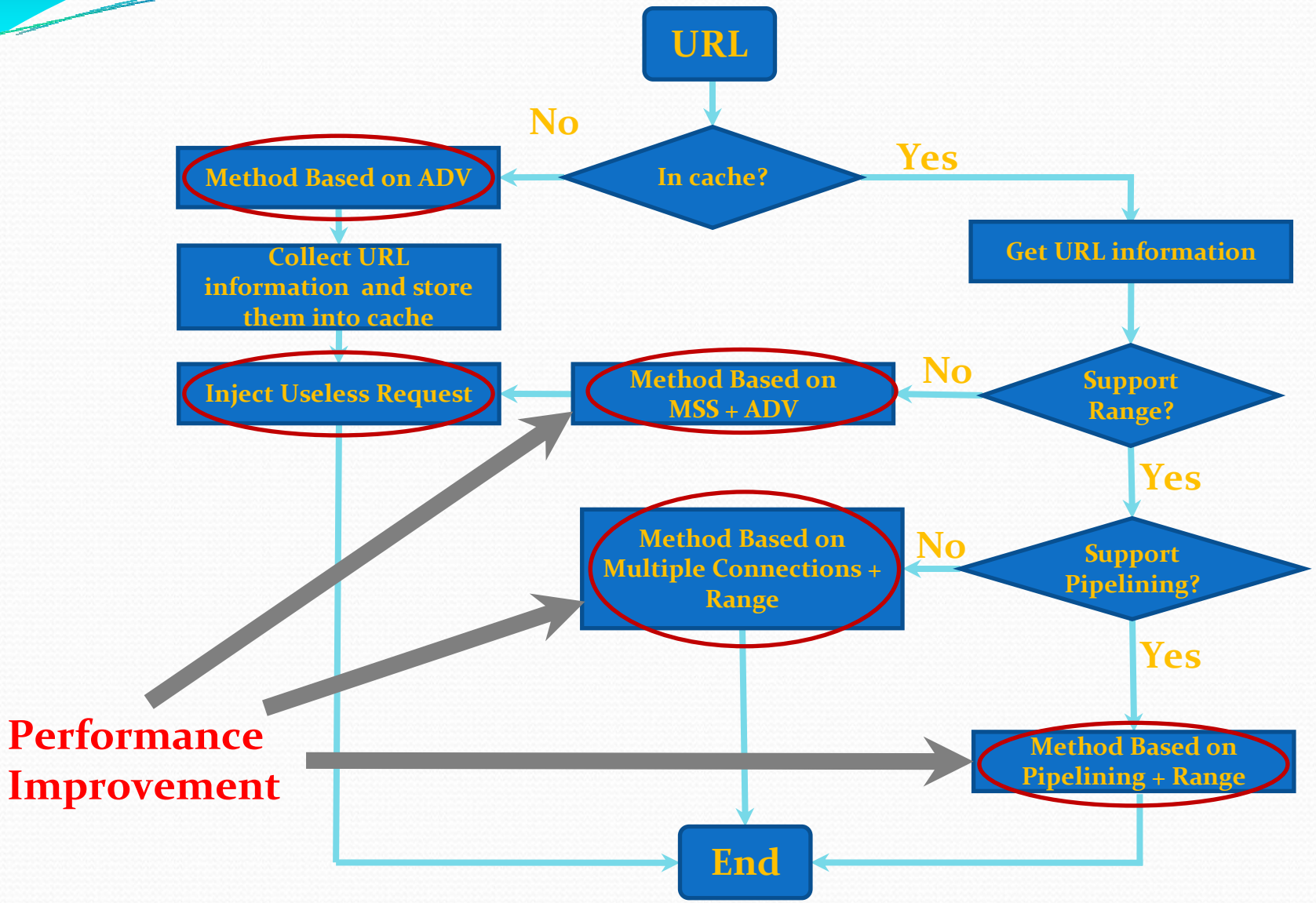
Web servers (# of URLs)	Range	Range + Pipelining
Apache(59698)	89.02%	68.80%
IIS(22485)	85.03%	73.38%
nginx(18714)	83.16%	70.74%
lighttpd(5506)	82.64%	67.51%
Others(36930)	66.74%	53.98%

Google web servers (# of URLs)	Range	Range + Pipelining
sffe(2580)	99.88%	99.88%
DFE/largefile(461)	100.0%	100.0%
GSE(906)	48.59%	48.59%
codesite(335)	0%	0%
Others(340)	0%	0%

Web servers (# of servers)	Pipelining
Apache(4249)	63.90%
IIS(1738)	77.00%
nginx(1103)	75.16%
lighttpd(367)	74.70%
Others(1388)	65.13%

Google web servers (# of servers)	Pipelining
sffe(38)	100.0%
DFE/largefile(109)	100.0%
GSE(24)	100.0%
codesite(2)	100.0%
Others(58)	100.0%

HTTPOS's Operation



Content

Motivation

Threat Model

HTTPOS

Implementation

Evaluation

Conclusion

Implementation

TCP Layer:

iptables + **libnetfilter_queue**

HTTP Layer:

Scenario	HTTPOS Implementation
Wireless Channel IPSec Tunnel	HTTP proxy
SSH Tunnel	SOCKS v4 proxy
HTTPS Channel	HTTPs proxy

Content

Motivation

Threat Model

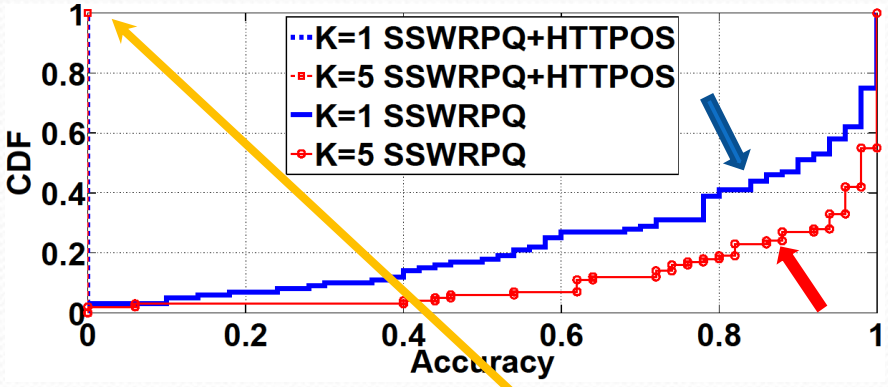
HTTPOS

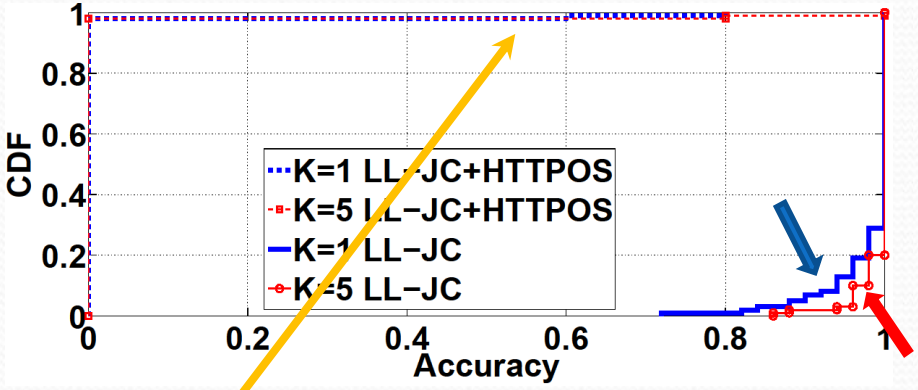
Implementation

Evaluation

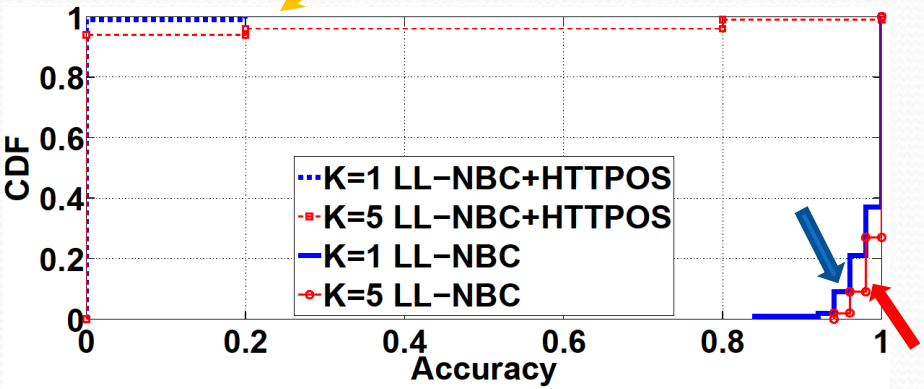
Conclusion

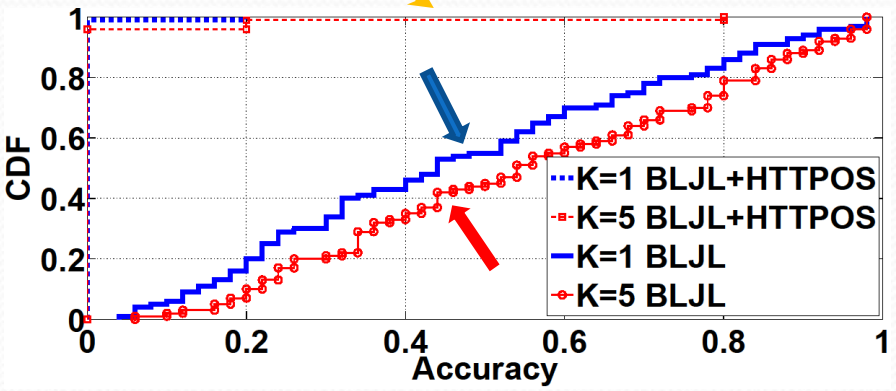
Evasion Evaluation through IPSec Tunnel



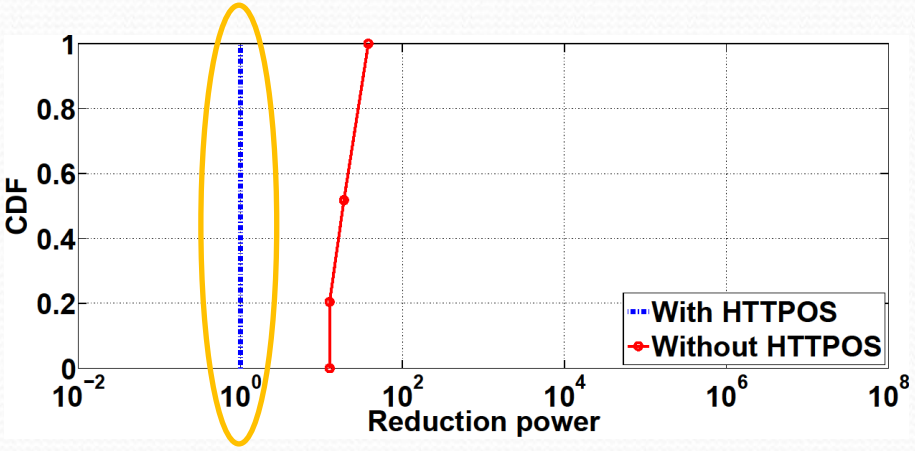


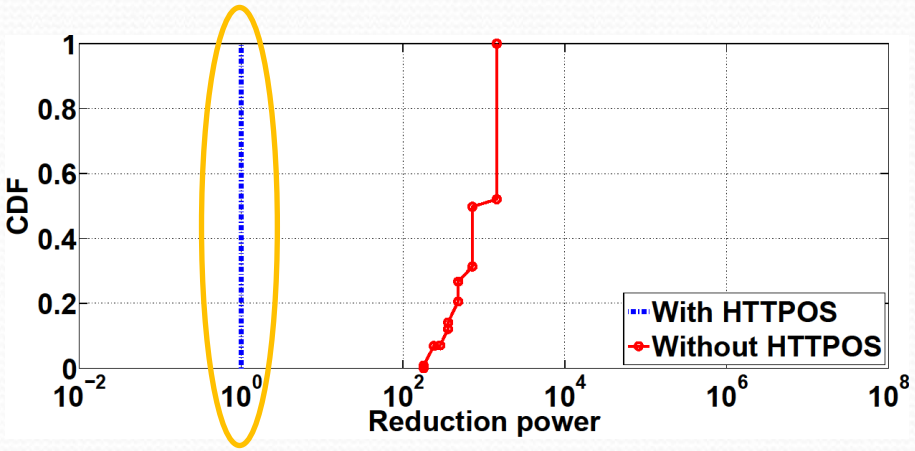
Evasion Capability by HTTPOS

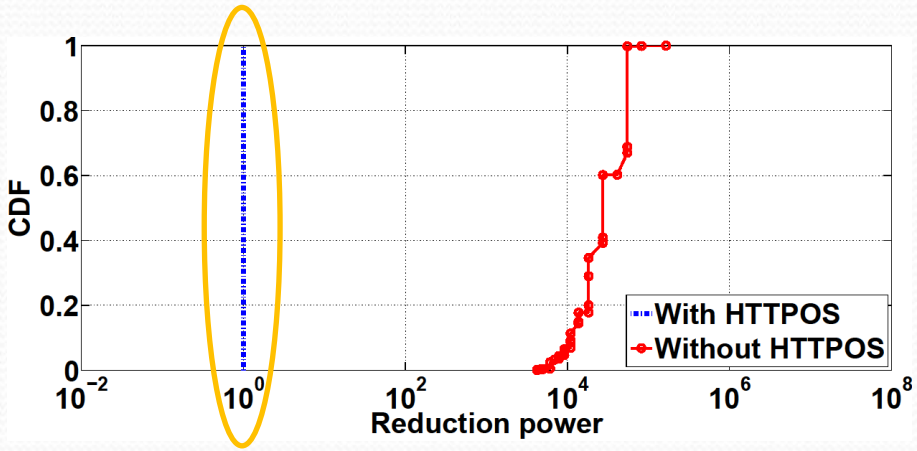


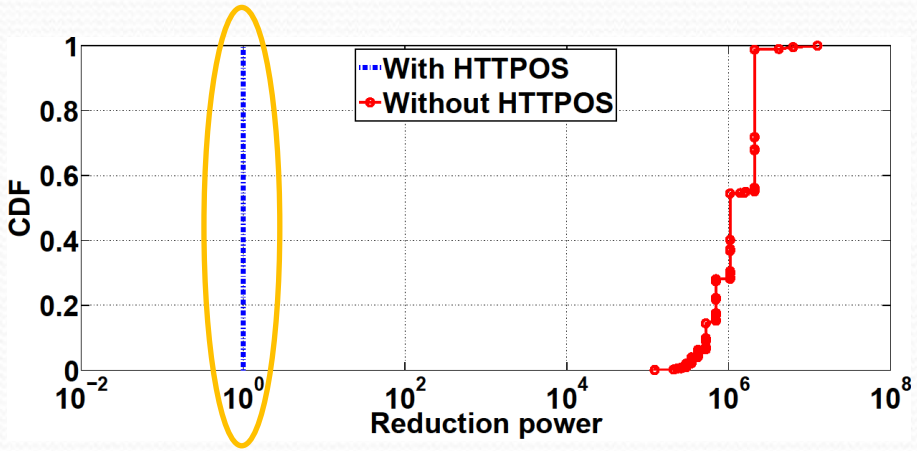


Evading CWWZ Attack through HTTPS Channel

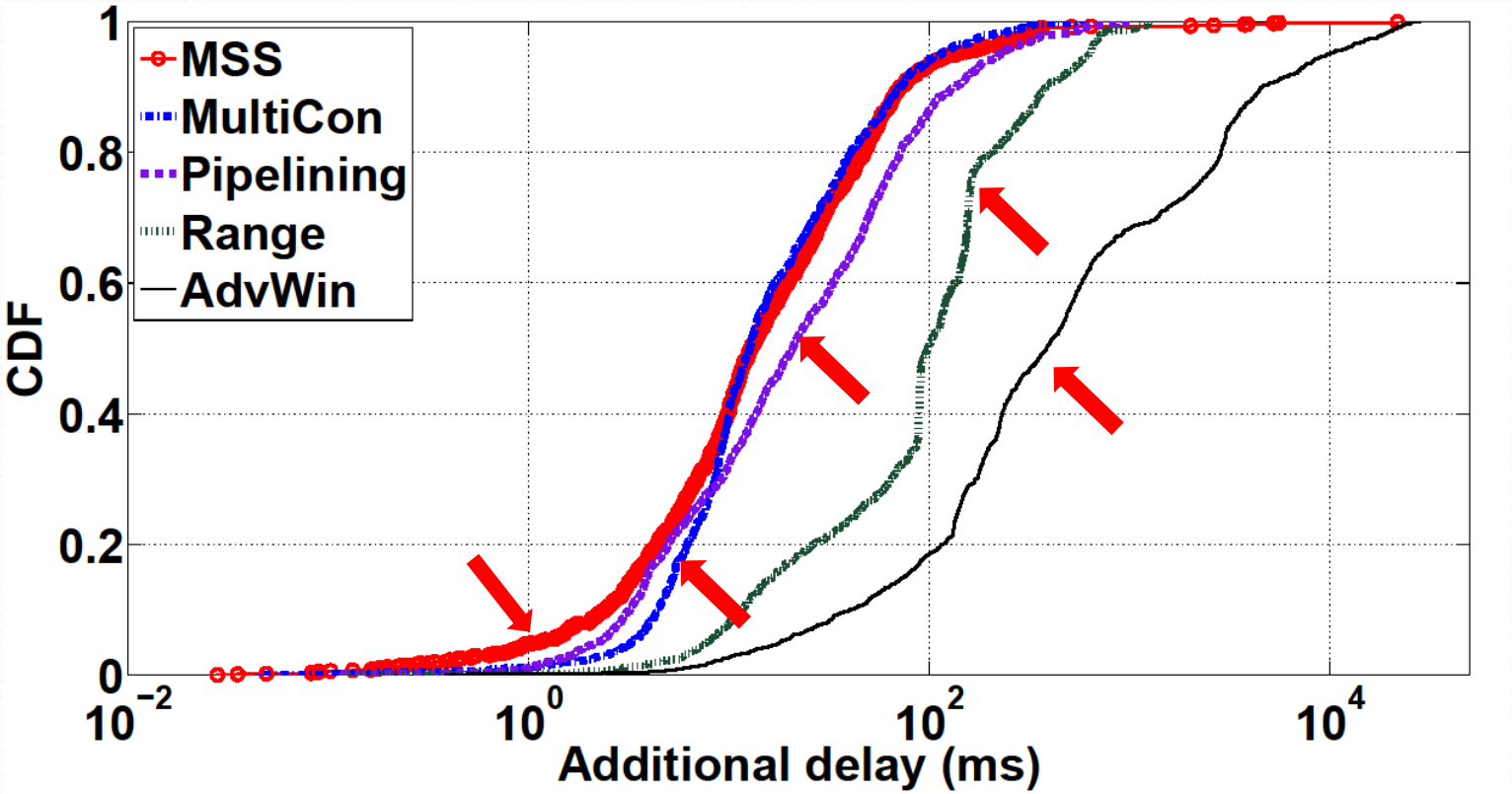






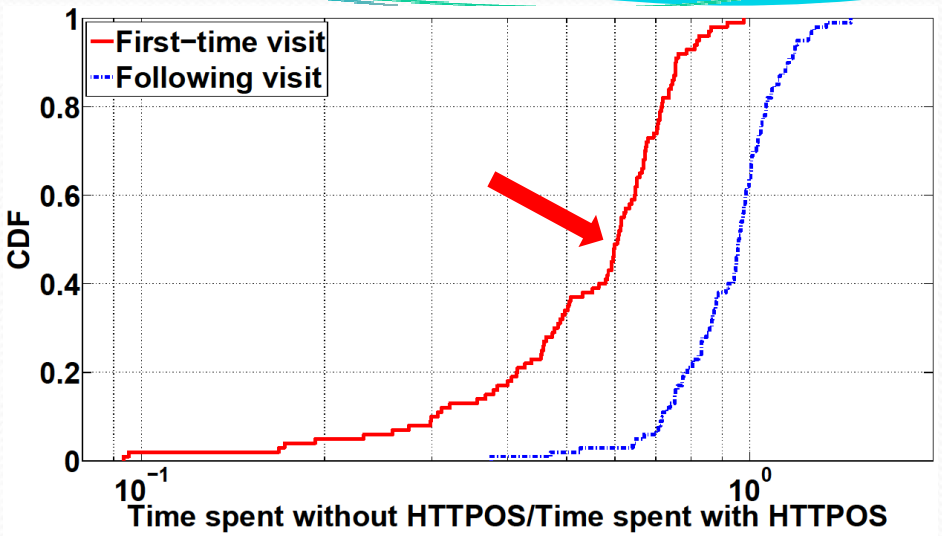


Single Method Performance

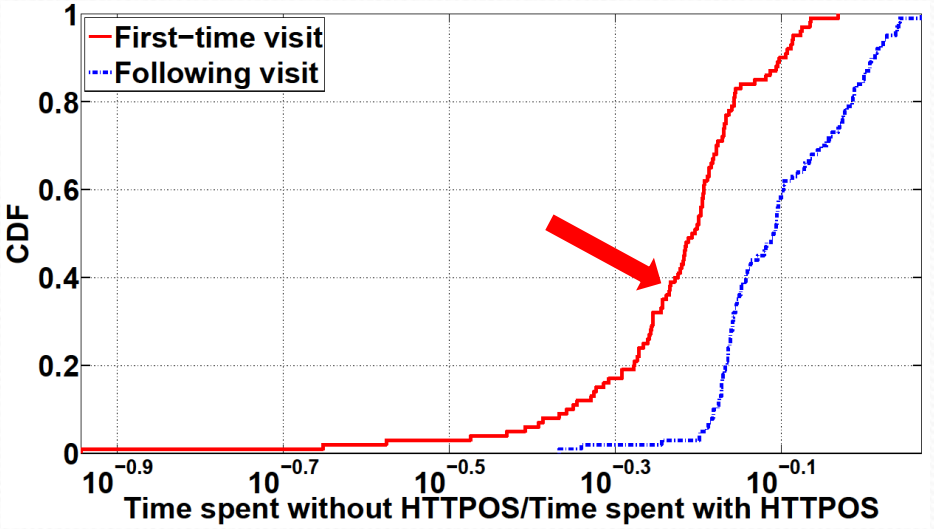


Impacts on the performance of Internet browsing

Evade Attack
SSWRPQ
BLJL
LL-JC & LL-NBC



Overall Performance in IPSec Tunnel

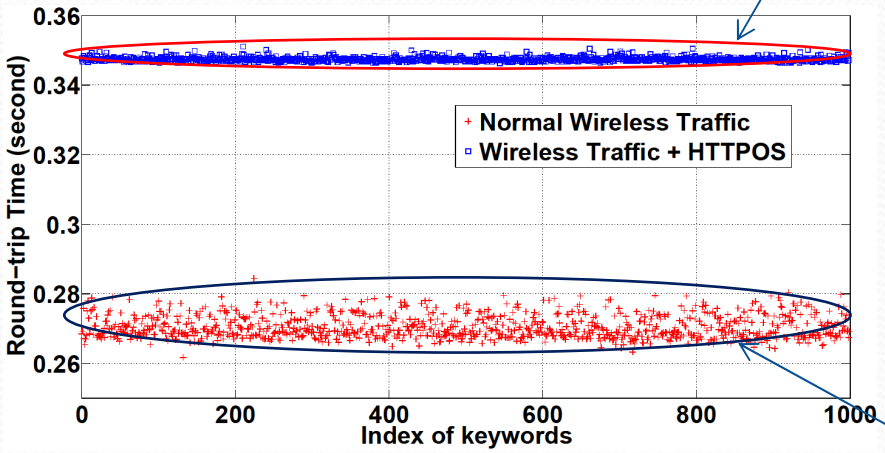


Overall Performance in SSH Tunnel

Impacts on the performance of Google Search

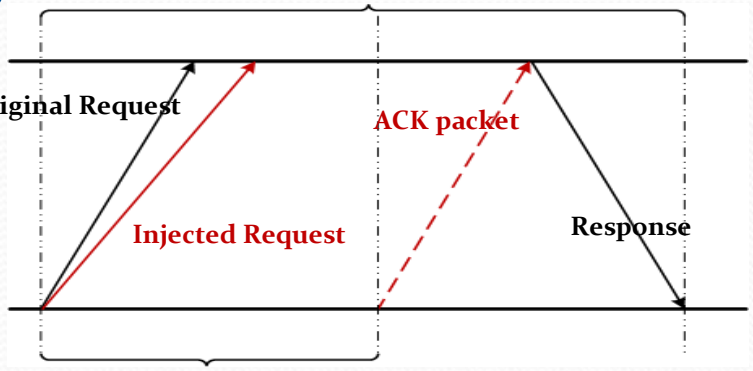
Evade Attack

CWWZ



Overall Performance in Wireless Channel

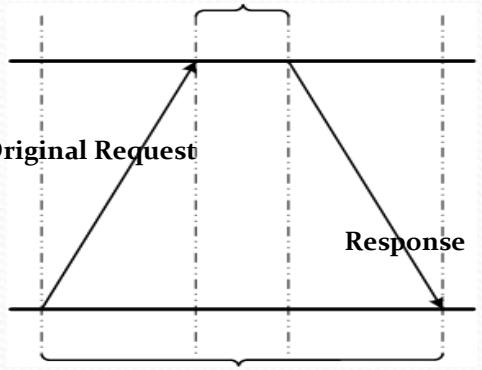
Request-to-Response Time



Injection Delay

With HTTPOS

Processing Time



Request-to-Response Time

Without HTTPOS

Content

Motivation

Threat Model

HTTPOS

Implementation

Evaluation

Conclusion

Limitations

URL does not support any features required by HTTPOS

Privacy leakage from SSL/TLS record length analysis

- **URLs supporting Range can be divided into randomly overlap partials**
- **Useless requests can raise the bar for the CWWZ attack**

Conclusion and Future Work

Browser-side techniques

sufficient and practice to avoid privacy leakage from encrypted HTTP flows

HTTPOS

Protect your own privacy on demand

Future Work

- **Further mitigating impact of HTTPOS on performance**
- **Sealing privacy leakages in other web applications**

Thanks