# Post-Quantum Signatures in DNSSEC via Request-Based Fragmentation

**Jason Goertzen**, Douglas Stebila

**Email: jason.goertzen@uwaterloo.ca**

UNIVERSITY OF WATERLOO

PQCrypto 2023

# WHAT ARE WE TALKING ABOUT TODAY?

## Background

- What is DNS?
- What (security) problems does DNS have?
- How are these problems addressed?
- Are these solutions future proofed?

## PQC, DNSSEC, & Request-based Fragmentation

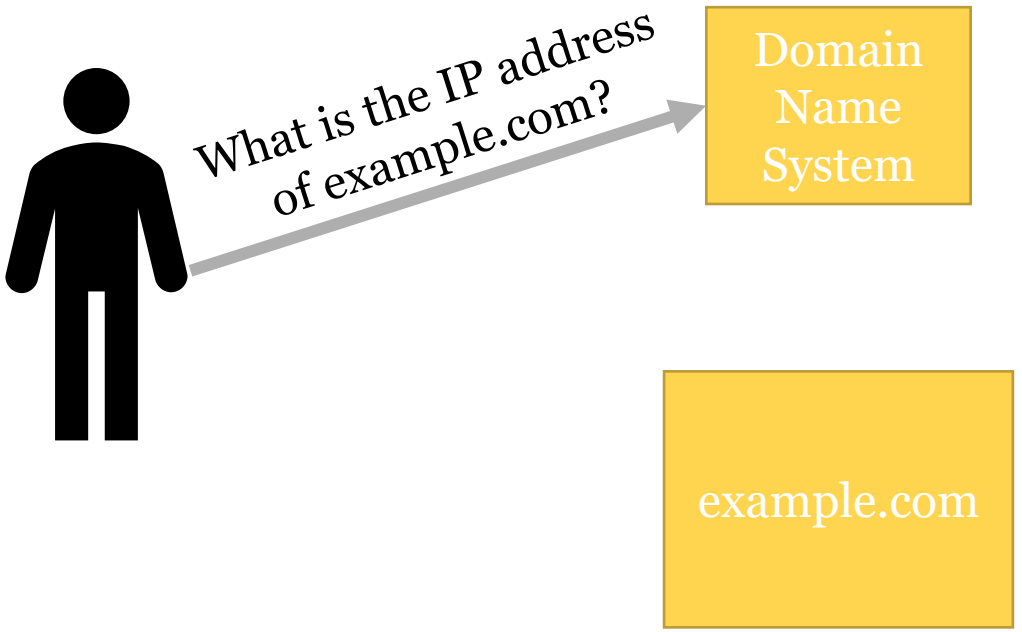
- Design
- Results & Evaluation
- Next Steps

# WHAT IS DNS?

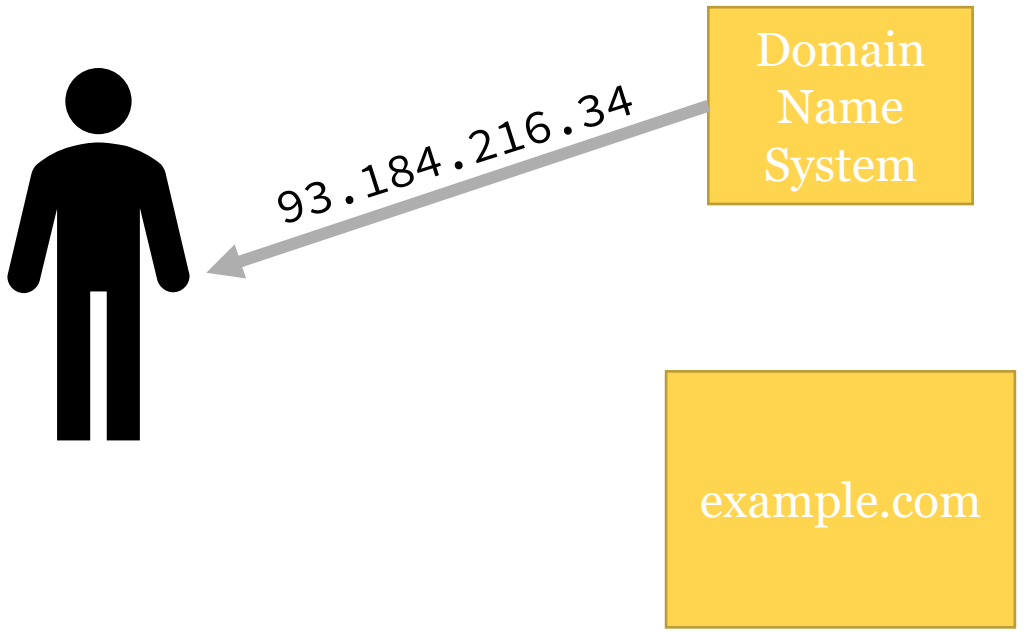
- The internet uses IP addresses to determine where to send messages
- IP addresses are difficult for people to remember!
- The Domain Name System is responsible to translating something easy for a human to remember into IP addresses

```
example.com -> 93.184.216.34
```

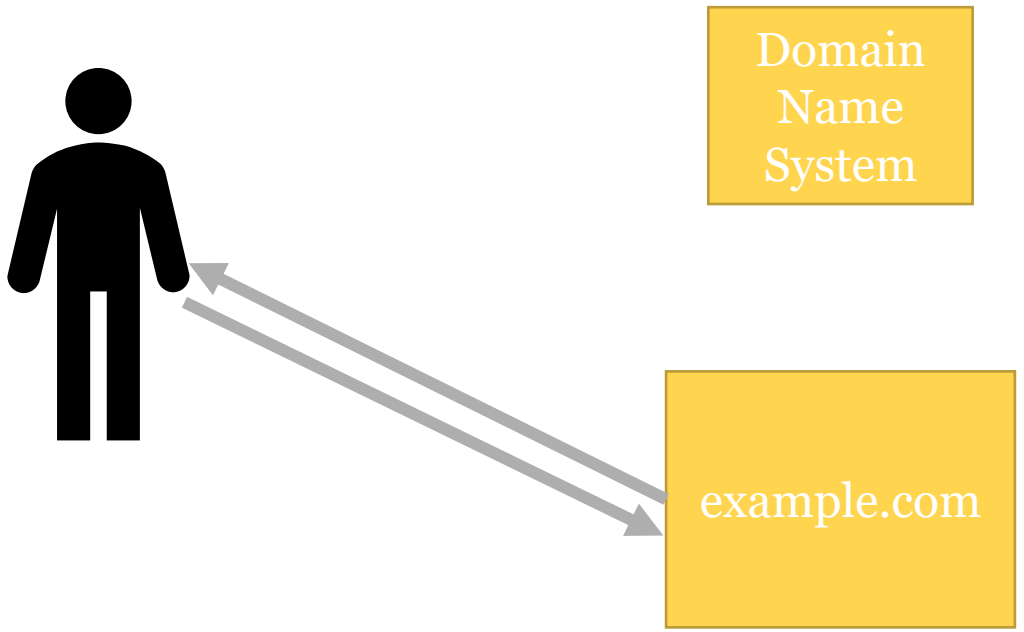
# WHAT IS DNS?


What is the IP address of example.com?
Domain Name System
example.com

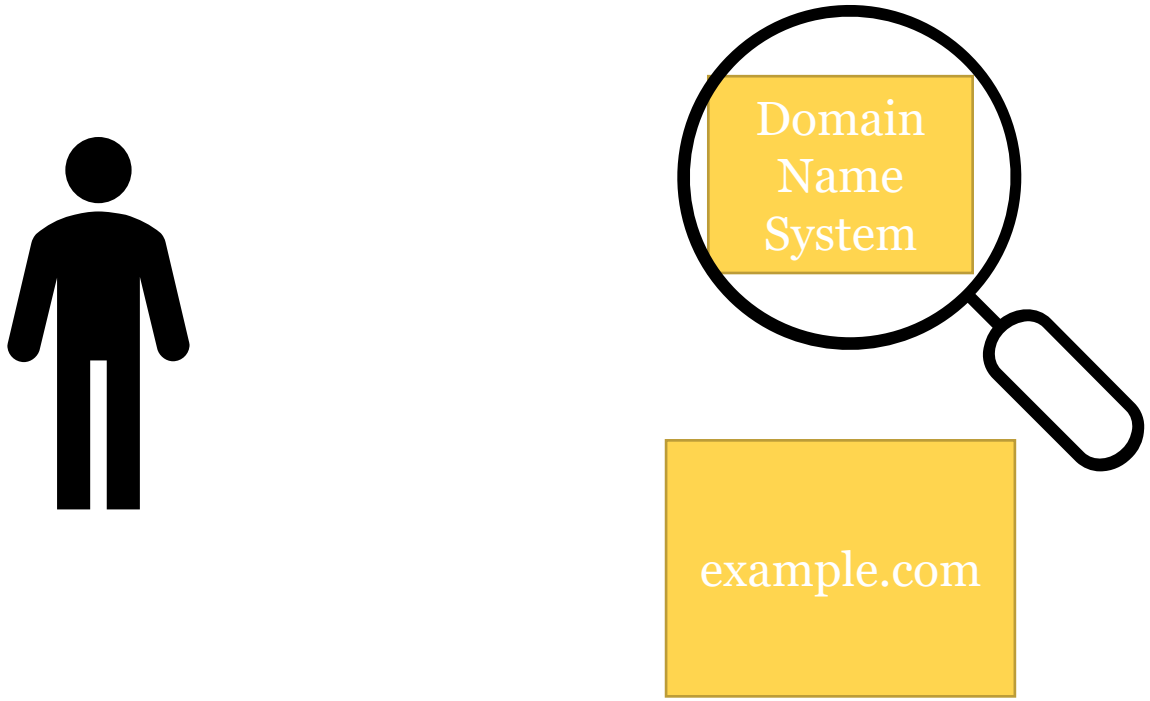
# WHAT IS DNS?



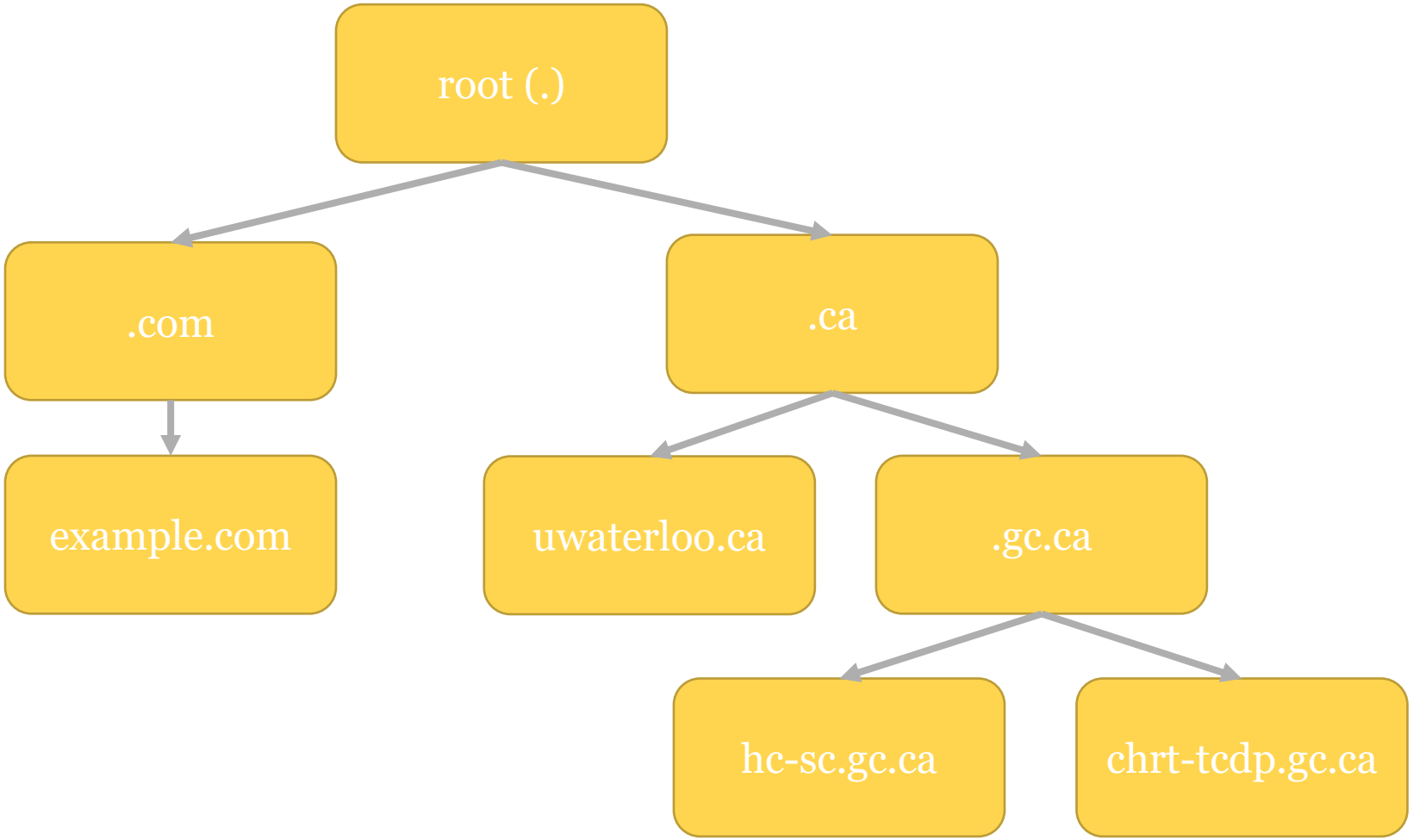
# WHAT IS DNS?



# WHAT IS DNS?



# DNS IS BROKEN UP INTO ZONES


Root level

Top level domains

Second level domains

Third level domains

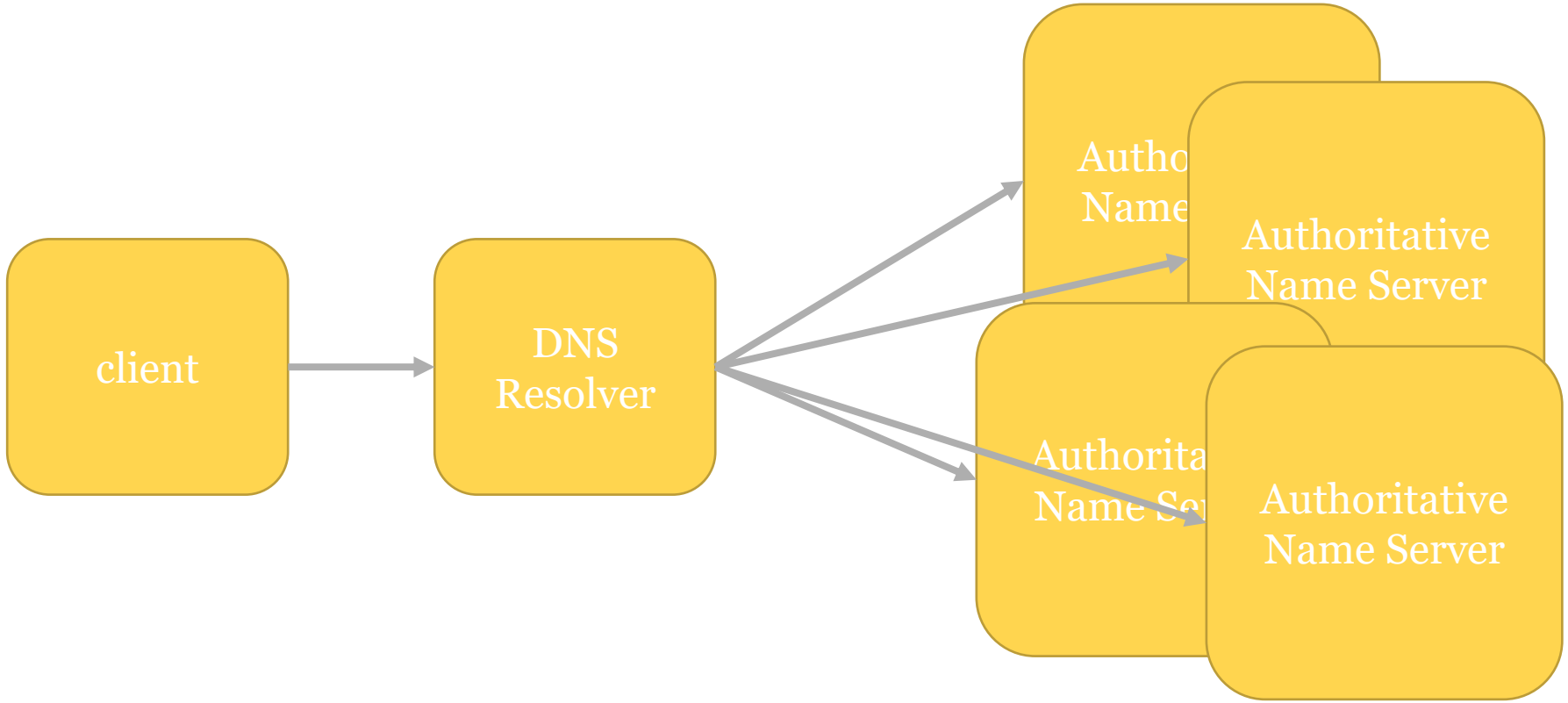


UNIVERSITY OF WATERLOO

# ZONES CONTAIN RESOURCE RECORDS


| | |
|---|---|
| example.com. 57094 IN AAAA | 2606:2800:220:1:248:1893:25c8:1946 |
| example.com. 57047 IN A | 93.184.216.34 |
| example.com. 57094 IN NS | b.iana-servers.net. |
| example.com. 57094 IN NS | a.iana-servers.net. |

# CLIENTS RARELY QUERY DIRECTLY

# PROBLEM WITH DNS


- Designed with no integrity projection

client

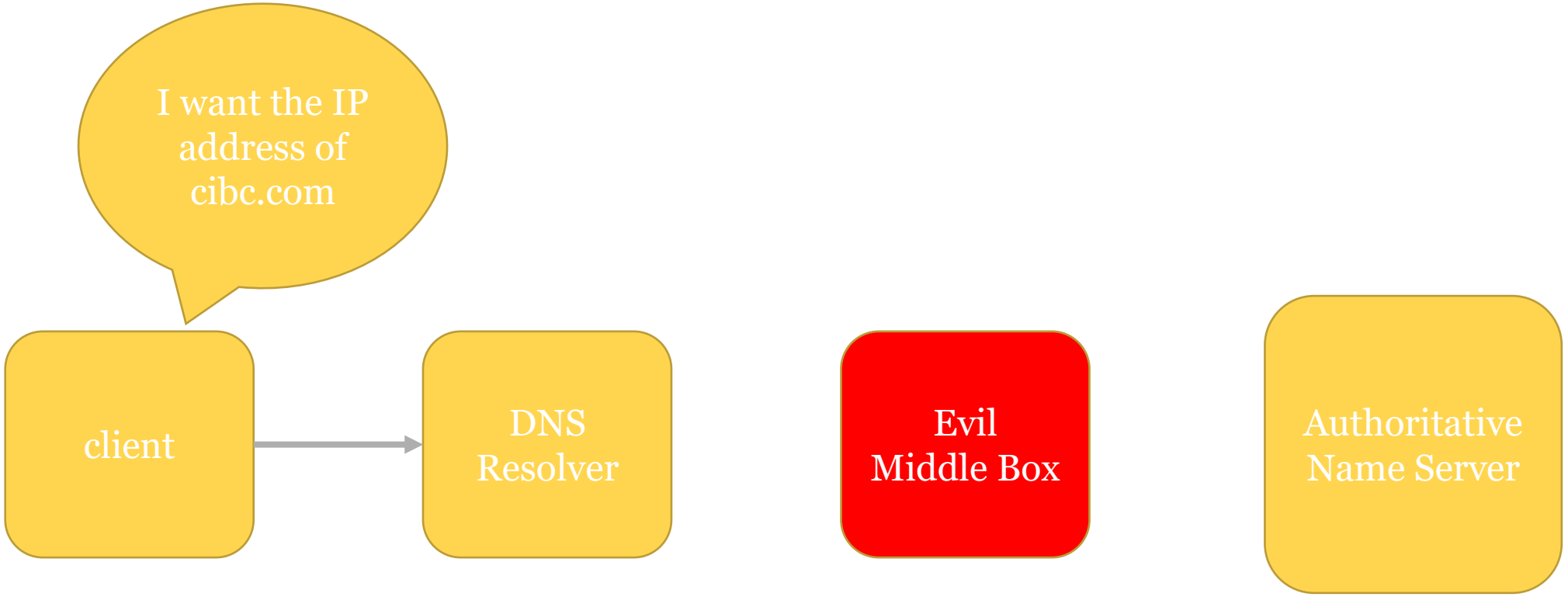
DNS Resolver

Evil Middle Box

Authoritative Name Server

# PROBLEM WITH DNS


- Designed with no integrity projection

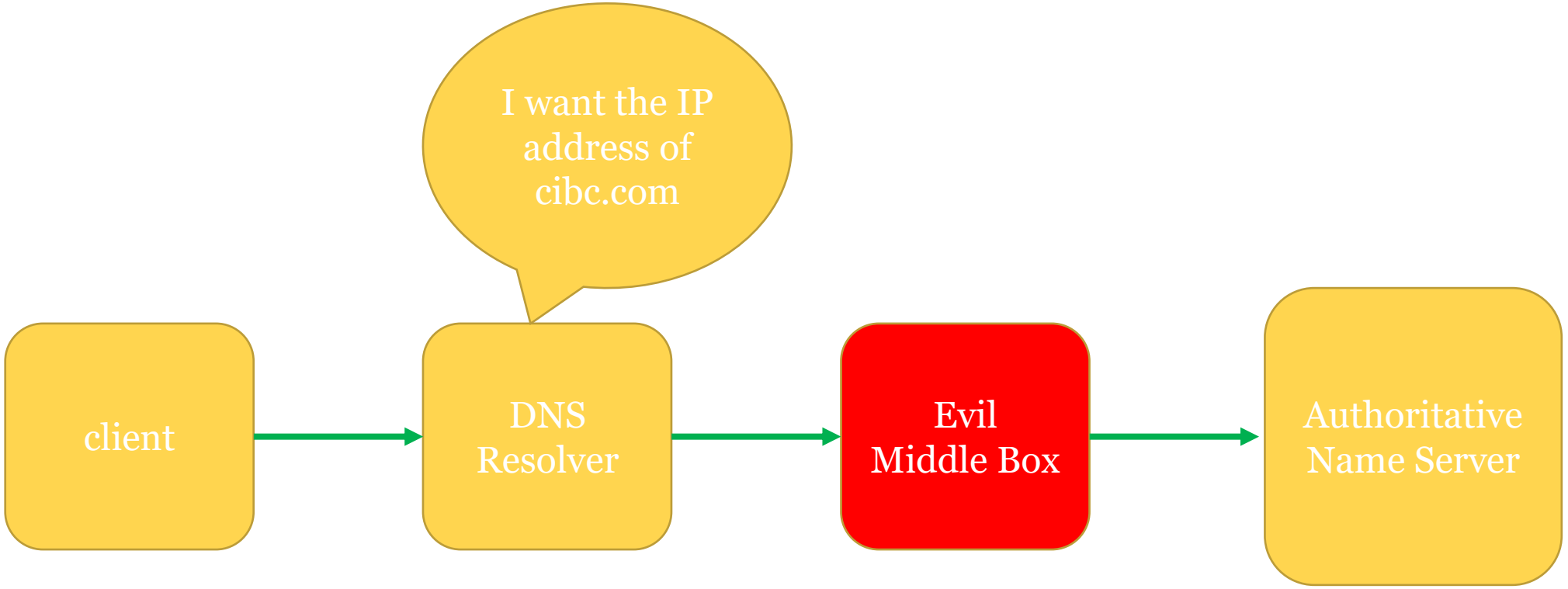





UNIVERSITY OF WATERLOO

# PROBLEM WITH DNS


- Designed with no integrity projection

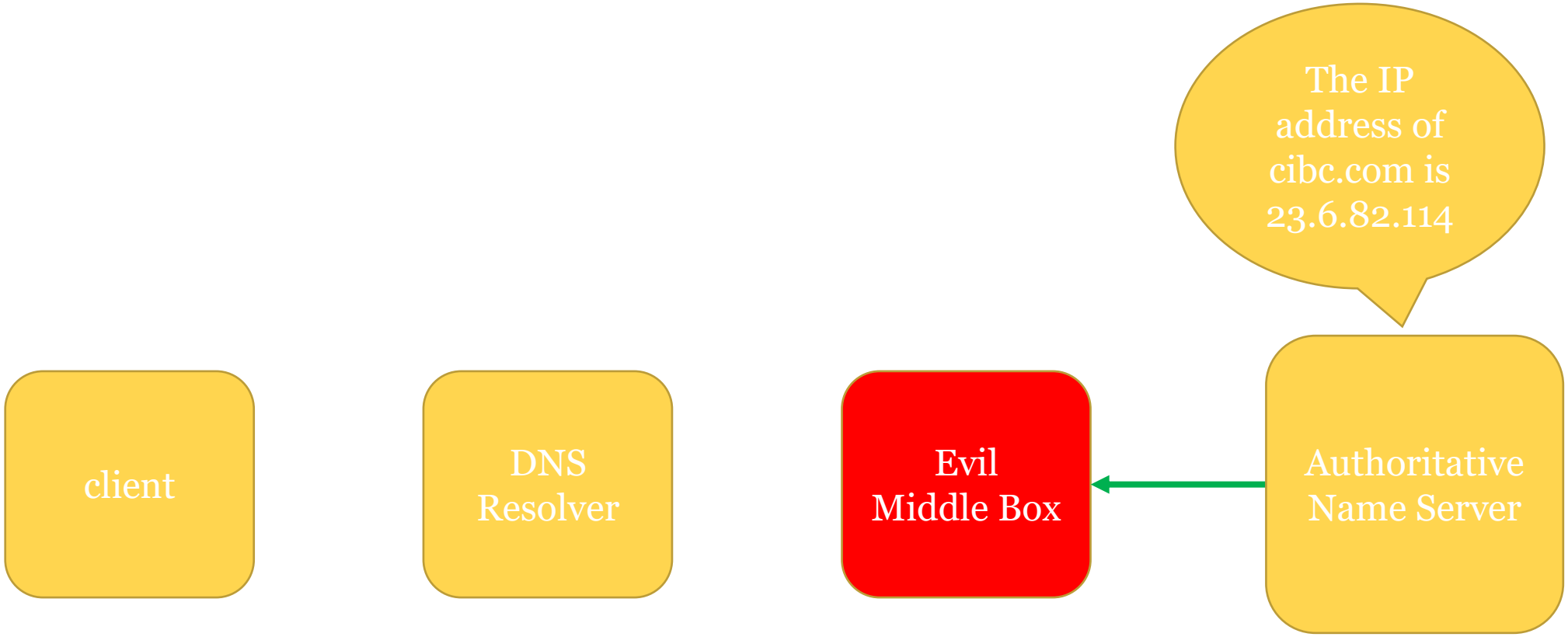





UNIVERSITY OF WATERLOO

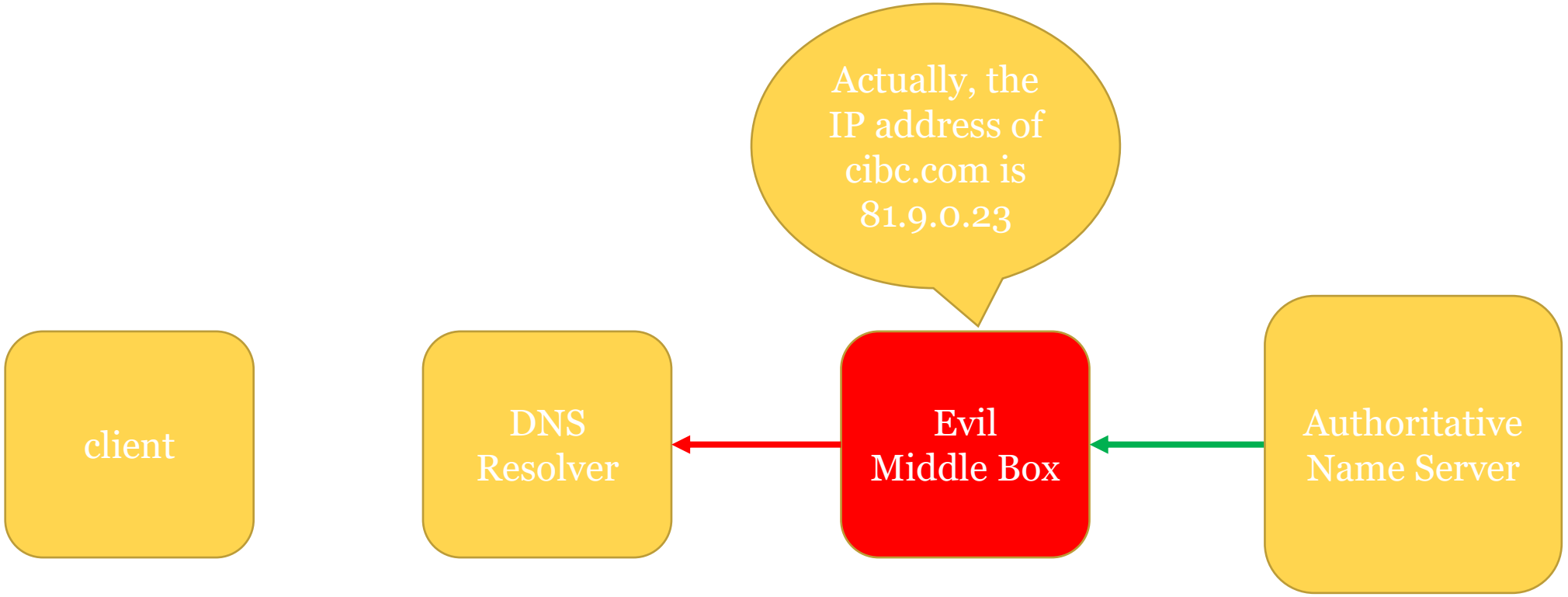
# PROBLEM WITH DNS


- Designed with no integrity projection



# PROBLEM WITH DNS


- Designed with no integrity projection

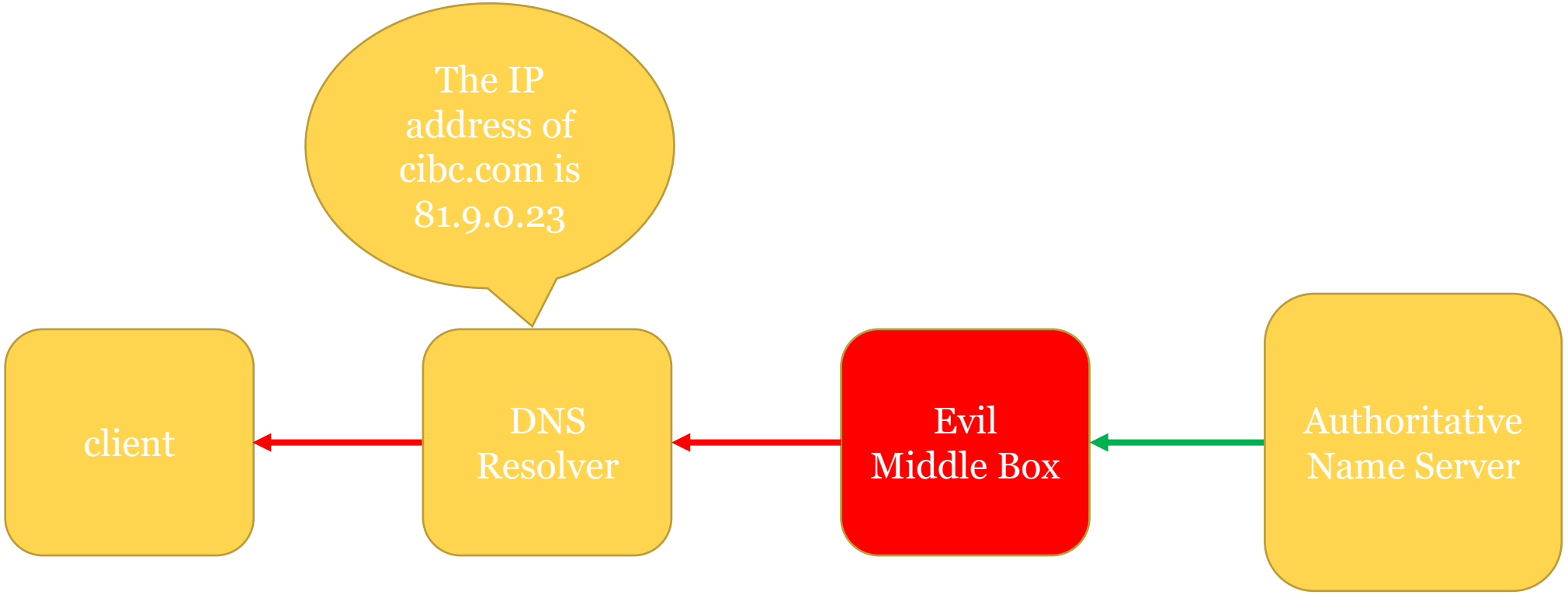





UNIVERSITY OF WATERLOO

# PROBLEM WITH DNS


- Designed with no integrity projection

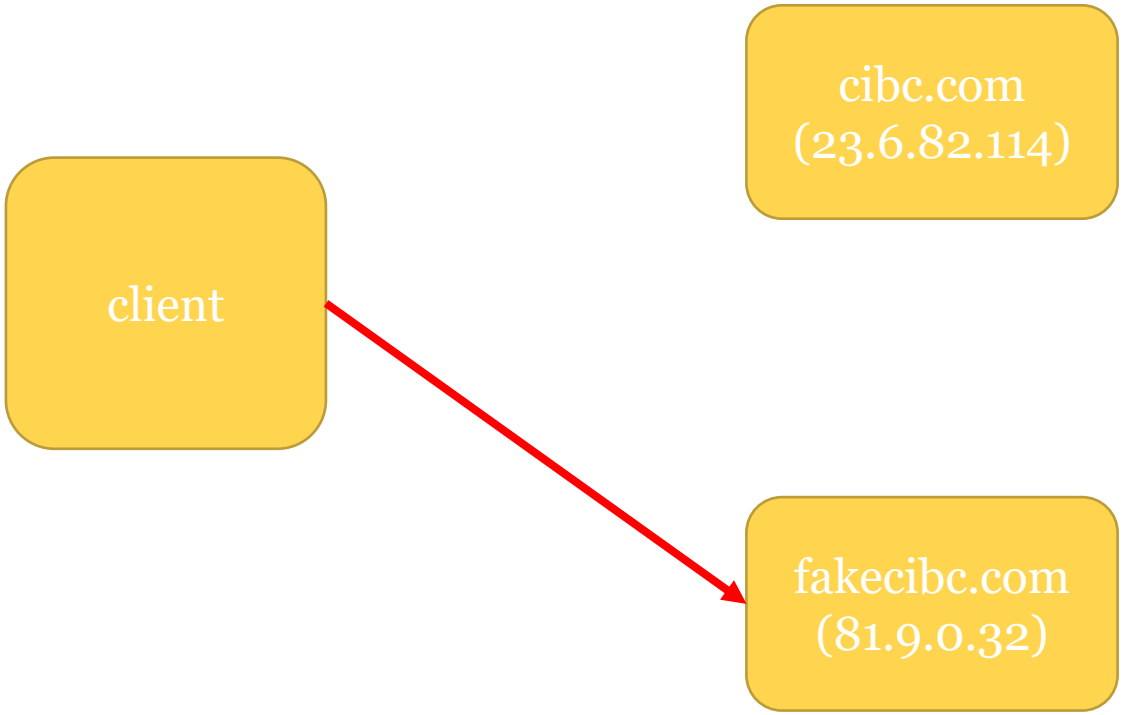





UNIVERSITY OF WATERLOO

# PROBLEM WITH DNS


- Designed with no integrity projection






UNIVERSITY OF WATERLOO

# SOLVING DNS INTEGRITY

Use digital signatures to make sure the correct unmodified message is received and is from the correct entity!

- New records added to DNSSEC signed zone
- Sets of records (RRSets) are signed, rather than individual records
- Have two keys:
  - Key Signing Key
  - Zone Signing Key

# SOLVING DNS INTEGRITY

| | | | | |
|---|---|---|---|---|
| example.com. | 86379 | IN | A | 93.184.216.34 |
| example.com. | 86328 | IN | RRSIG | A 8 2 86400 20221120061546 20221030132217 59208 example.com. rZvjehQxdT5pJ4cw+o1y/BYmLkBLuqzjFaEOn9773Bhywt4qhKmME8DK oKD4yLjYJYFaqhUNCYb+iimCTdK+9+3UjJ35gRIDC3kuZ9hogtCoLBnt ltfgFwLQomdye8iH/FDDVKTm+CAz3UMfcwNzNahvg4BOnZ04HqnpZcWW pu4= |
| example.com. | 73820 | IN | NS | a.iana-servers.net. |
| example.com. | 73820 | IN | NS | b.iana-servers.net. |
| example.com. | 86237 | IN | RRSIG | NS 8 2 86400 20221122065049 20221101032317 59208 example.com. Uit3UXCeClM+iwVkc2lX8n5A1O0CD9mH8rsTSfcsjQaZD9y54q7bT2mM cGMaiyjj/sODGKLNvbFKLEgHgP0LNF4i+YzHvpct5MZD1c8JqnzYisOf xq+JQ4tLcsDmrnhJEinBVbiq/epEXs04I4GES+zyEgnz5TPErjTNRDzP 7CE= |
| example.com. | 3600 | IN | DNSKEY | 257 3 8 AwEAAZ0aqu1rJ6orJynrRfNpPmayJZ0Ax9Ic2/Rl9VQWLMHyjxxem3VU S0NUIFXERQbj0A9Ogp0zDM9YIccKLRd6LmWiDCt7UJQxVdD+heb5Ec4q lqGmyX9MDabkvX2NvMwsUecbYBq80XeTT9LRmCUt9KUt/WOi6DKECx0G /bWTykrXyBR8elD+SQY43OAVjlWrVltHxgp4/rhBCvRbmdflunaPIgu2 7eE2U4myDSLT8a4A0rB5uHG4PkOa9dIRs9y00M2mWf4lyPee7vi5few2 dbayHXmieGcaAHrx76NGAABeY393xjlmDNcUkF1gpNWUla4fWZbbaYQz A93mLdrng+M= |
| example.com. | 3600 | IN | DNSKEY | 256 3 8 AwEAAb10JO+fCqdkxHtQYVB/tFPgJphc+VxjUYz+eVGf077zMxHKgce9 EwGBifFuKhjl2EA0VQPsWVX1vzuUmWri3OgsTBlITkdMz6VU4g94uO6T 9MIktokouOidIzvOqLR+O2LSXNhiYOIWA9s3Lxk5R2lrwd6vrRvT2CR1 GdZuUlKB |
| example.com. | 2694 | IN | RRSIG | DNSKEY 8 2 3600 20221129010414 20221107233521 31406 example.com. nMEQXWFatPZd/fkGgi9TI4ZO2vokX+6zNNmZPSOnweki1Vb25f+oISgH b1WEg84IzyUw+zzwmS2G4J08PvS8+rFfu9vprvPwKVMsgOzBSyt3CCLS qa1DtY20BMWXCzqHD1n16220AUMNGuvrta6ikmuGfXT/gXyK5isenUPn kSbGsbrgEQKPZZQU6H/9nLK2qttyBscCQmJ4ziIbsMyannBWgXtJgXhu 4AhiVAZIxCqII/ISNei3vOcl+h6C+RgjYsnoPD59HkpnC2H7TsaiLNf7 uYtbCjzRKLhRzIwIS3ASbWccGJ3LXruZwUNd0E/XqrxaCZXuwFrq+vtP RYAaPA== |



UNIVERSITY OF WATERLOO

# SOLVING DNS INTEGRITY

| | | | | |
|---|---|---|---|---|
| example.com. | 86379 | IN | A | 93.184.216.34 } RRSet #1 |
| example.com. | 86328 | IN | RRSIG | A 8 2 86400 20221120061546 20221030132217 59208 example.com. rZvjehQxdT5pJ4cw+o1y/BYmLkBLuqzjFaEOn9773Bhywt4qhKmME8DK oKD4yLjYJYFaqhUNCYb+iimCTdK+9+3UjJ35gRIDC3kuZ9hogtCoLBnt ltfgFwLQomdye8iH/FDDVKTm+CAz3UMfcwNzNahvg4BOnZ04HqnpZcWW pu4= |
| example.com. | 73820 | IN | NS | a.iana-servers.net. |
| example.com. | 73820 | IN | NS | b.iana-servers.net. |
| example.com. | 86237 | IN | RRSIG | NS 8 2 86400 20221122065049 20221101032317 59208 example.com. Uit3UXCeClM+iwVkc2lX8n5A1O0CD9mH8rsTSfcsjQaZD9y54q7bT2mM cGMaiyjj/sODGKLNvbFKLEgHgP0LNF4i+YzHvpct5MZD1c8JqnzYisOf xq+JQ4tLcsDmrnhJEinBVbiq/epEXs04I4GES+zyEgnz5TPErjTNRDzP 7CE= |
| example.com. | 3600 | IN | DNSKEY | 257 3 8 AwEAAZ0aqu1rJ6orJynrRfNpPmayJZ0Ax9Ic2/Rl9VQWLMHyjxxem3VU S0NUIFXERQbj0A9Ogp0zDM9YIccKLRd6LmWiDCt7UJQxVdD+heb5Ec4q lqGmyX9MDabkvX2NvMwsUecbYBq80XeTT9LRmCUt9KUt/WOi6DKECx0G /bWTykrXyBR8elD+SQY43OAVjlWrVltHxgp4/rhBCvRbmdflunaPIgu2 7eE2U4myDSLT8a4A0rB5uHG4PkOa9dIRs9y00M2mWf4lyPee7vi5few2 dbayHXmieGcaAHrx76NGAABeY393xjlmDNcUkF1gpNWUla4fWZbbaYQz A93mLdrng+M= |
| example.com. | 3600 | IN | DNSKEY | 256 3 8 AwEAAb10JO+fCqdkxHtQYVB/tFPgJphc+VxjUYz+eVGf077zMxHKgce9 EwGBifFuKhjl2EA0VQPsWVX1vzuUmWri3OgsTBlITkdMz6VU4g94uO6T 9MIktokouOidIzvOqLR+O2LSXNhiYOIWA9s3Lxk5R2lrwd6vrRvT2CR1 GdZuUlKB |
| example.com. | 2694 | IN | RRSIG | DNSKEY 8 2 3600 20221129010414 20221107233521 31406 example.com. nMEQXWFatPZd/fkGgi9TI4ZO2vokX+6zNNmZPSOnweki1Vb25f+oISgH b1WEg84IzyUw+zzwmS2G4J08PvS8+rFfu9vprvPwKVMsgOzBSyt3CCLS qa1DtY20BMWXCzqHD1n16220AUMNGuvrta6ikmuGfXT/gXyK5isenUPn kSbGsbrgEQKPZZQU6H/9nLK2qttyBscCQmJ4ziIbsMyannBWgXtJgXhu 4AhiVAZIxCqII/ISNei3vOcl+h6C+RgjYsnoPD59HkpnC2H7TsaiLNf7 uYtbCjzRKLhRzIwIS3ASbWccGJ3LXruZwUNd0E/XqrxaCZXuwFrq+vtP RYAaPA== |



UNIVERSITY OF WATERLOO

# SOLVING DNS INTEGRITY

| | | | | |
|---|---|---|---|---|
| example.com. | 86379 | IN | A | 93.184.216.34 |
| example.com. | 86328 | IN | RRSIG | A 8 2 86400 20221120061546 20221030132217 59208 example.com. rZvjehQxdT5pJ4cw+o1y/BYmLkBLuqzjFaEOn9773Bhywt4qhKmME8DK oKD4yLjYJYFaqhUNCYb+iimCTdK+9+3UjJ35gRIDC3kuZ9hogtCoLBnt ltfgFwLQomdye8iH/FDDVKTm+CAz3UMfcwNzNahvg4BOnZo4HqnpZcWW pu4= |
| example.com. | 73820 | IN | NS | a.iana-servers.net. |
| example.com. | 73820 | IN | NS | b.iana-servers.net. |
| example.com. | 86237 | IN | RRSIG | NS 8 2 86400 20221122065049 20221101032317 59208 example.com. Uit3UXCeClM+iwVkc2lX8n5A1O0CD9mH8rsTSfcsjQaZD9y54q7bT2mM cGMaiyjj/sODGKLNvbFKLEgHgP0LNF4i+YzHvpct5MZD1c8JqnzYisOf xq+JQ4tLcsDmrnhJEinBVbiq/epEXs04I4GES+zyEgnz5TPErjTNRDzP 7CE= |
| example.com. | 3600 | IN | DNSKEY | 257 3 8 AwEAAZoaqu1rJ6orJynrRfNpPmayJZoAx9Ic2/Rl9VQWLMHyjxxem3VU SoNUIFXERQbj0A9Ogp0zDM9YIccKLRd6LmWiDCt7UJQxVdD+heb5Ec4q lqGmyX9MDabkvX2NvMwsUecbYBq8oXeTT9LRmCUt9KUt/WOi6DKECxoG /bWTykrXyBR8elD+SQY43OAVjlWrVltHxgp4/rhBCvRbmdflunaPIgu2 7eE2U4myDSLT8a4A0rB5uHG4PkOa9dIRs9y00M2mWf4lyPee7vi5few2 dbayHXmieGcaAHrx76NGAABeY393xjlmDNcUkF1gpNWUla4fWZbbaYQz A93mLdrng+M= |
| example.com. | 3600 | IN | DNSKEY | 256 3 8 AwEAAb10JO+fCqdkxHtQYVB/tFPgJphc+VxjUYz+eVGf077zMxHKgce9 EwGBifFuKhjl2EA0VQPsWVX1vzuUmWri3OgsTBlITkdMz6VU4g94uO6T 9MIktokouOidIzvOqLR+O2LSXNhiYOIWA9s3Lxk5R2lrwd6vrRvT2CR1 GdZuUlKB |
| example.com. | 2694 | IN | RRSIG | DNSKEY 8 2 3600 20221129010414 20221107233521 31406 example.com. nMEQXWFatPZd/fkGgi9TI4ZO2vokX+6zNNmZPSOnweki1Vb25f+oISgH b1WEg84IzyUw+zzwmS2G4Jo8PvS8+rFfu9vprvPwKVMsgOzBSyt3CCLS qa1DtY20BMWXCzqHD1n16220AUMNGuvrta6ikmuGfXT/gXyK5isenUPn kSbGsbrgEQKPZZQU6H/9nLK2qttyBscCQmJ4ziIbsMyannBWgXtJgXhu 4AhiVAZIxCqII/ISNei3vOcl+h6C+RgjYsnoPD59HkpnC2H7TsaiLNf7 uYtbCjzRKLhRzIwIS3ASbWccGJ3LXruZwUNd0E/XqrxaCZXuwFrq+vtP RYAaPA== |

RRSet #1

RRSet #2



UNIVERSITY OF WATERLOO

# SOLVING DNS INTEGRITY

| | | | | |
|---|---|---|---|---|
| example.com. | 86379 | IN | A | 93.184.216.34 |
| example.com. | 86328 | IN | RRSIG | A 8 2 86400 20221120061546 20221030132217 59208 example.com. rZvjehQxdT5pJ4cw+o1y/BYmLkBLuqzjFaEOn9773Bhywt4qhKmME8DK oKD4yLjYJYFaqhUNCYb+iimCTdK+9+3UjJ35gRIDC3kuZ9hogtCoLBnt ltfgFwLQomdye8iH/FDDVKTm+CAz3UMfcwNzNahvg4BOnZ04HqnpZcWW pu4= |
| example.com. | 73820 | IN | NS | a.iana-servers.net. |
| example.com. | 73820 | IN | NS | b.iana-servers.net. |
| example.com. | 86237 | IN | RRSIG | NS 8 2 86400 20221122065049 20221101032317 59208 example.com. Uit3UXCeClM+iwVkc2lX8n5A1O0CD9mH8rsTSfcsjQaZD9y54q7bT2mM cGMaiyjj/sODGKLNvbFKLEgHgP0LNF4i+YzHvpct5MZD1c8JqnzYisOf xq+JQ4tLcsDmrnhJEinBVbiq/epEXs04I4GES+zyEgnz5TPErjTNRDzP 7CE= |
| example.com. | 3600 | IN | DNSKEY | 257 3 8 AwEAAZ0aqu1rJ6orJynrRfNpPmayJZ0Ax9Ic2/Rl9VQWLMHyjxxem3VU S0NUIFXERQbj0A9Ogp0zDM9YIccKLRd6LmWiDCt7UJQxVdD+heb5Ec4q lqGmyX9MDabkvX2NvMwsUecbYBq80XeTT9LRmCUt9KUt/WOi6DKECx0G /bWTykrXyBR8elD+SQY43OAVjlWrVltHxgp4/rhBCvRbmdflunaPIgu2 7eE2U4myDSLT8a4A0rB5uHG4PkOa9dIRs9y00M2mWf4lyPee7vi5few2 dbayHXmieGcaAHrx76NGAABeY393xjlmDNcUkF1gpNWUla4fWZbbaYQz A93mLdrng+M= |
| example.com. | 3600 | IN | DNSKEY | 256 3 8 AwEAAb10JO+fCqdkxHtQYVB/tFPgJphc+VxjUYz+eVGf077zMxHKgce9 EwGBifFuKhjl2EA0VQPsWVX1vzuUmWri3OgsTBlITkdMz6VU4g94uO6T 9MIktokouOidIzvOqLR+O2LSXNhiYOIWA9s3Lxk5R2lrwd6vrRvT2CR1 GdZuUlKB |
| example.com. | 2694 | IN | RRSIG | DNSKEY 8 2 3600 20221129010414 20221107233521 31406 example.com. nMEQXWFatPZd/fkGgi9TI4ZO2vokX+6zNNmZPSOnweki1Vb25f+oISgH b1WEg84IzyUw+zzwmS2G4J08PvS8+rFfu9vprvPwKVMsgOzBSyt3CCLS qa1DtY20BMWXCzqHD1n16220AUMNGuvrta6ikmuGfXT/gXyK5isenUPn kSbGsbrgEQKPZZQU6H/9nLK2qttyBscCQmJ4ziIbsMyannBWgXtJgXhu 4AhiVAZIxCqII/ISNei3vOcl+h6C+RgjYsnoPD59HkpnC2H7TsaiLNf7 uYtbCjzRKLhRzIwIS3ASbWccGJ3LXruZwUNd0E/XqrxaCZXuwFrq+vtP RYAaPA== |

RRSet #1

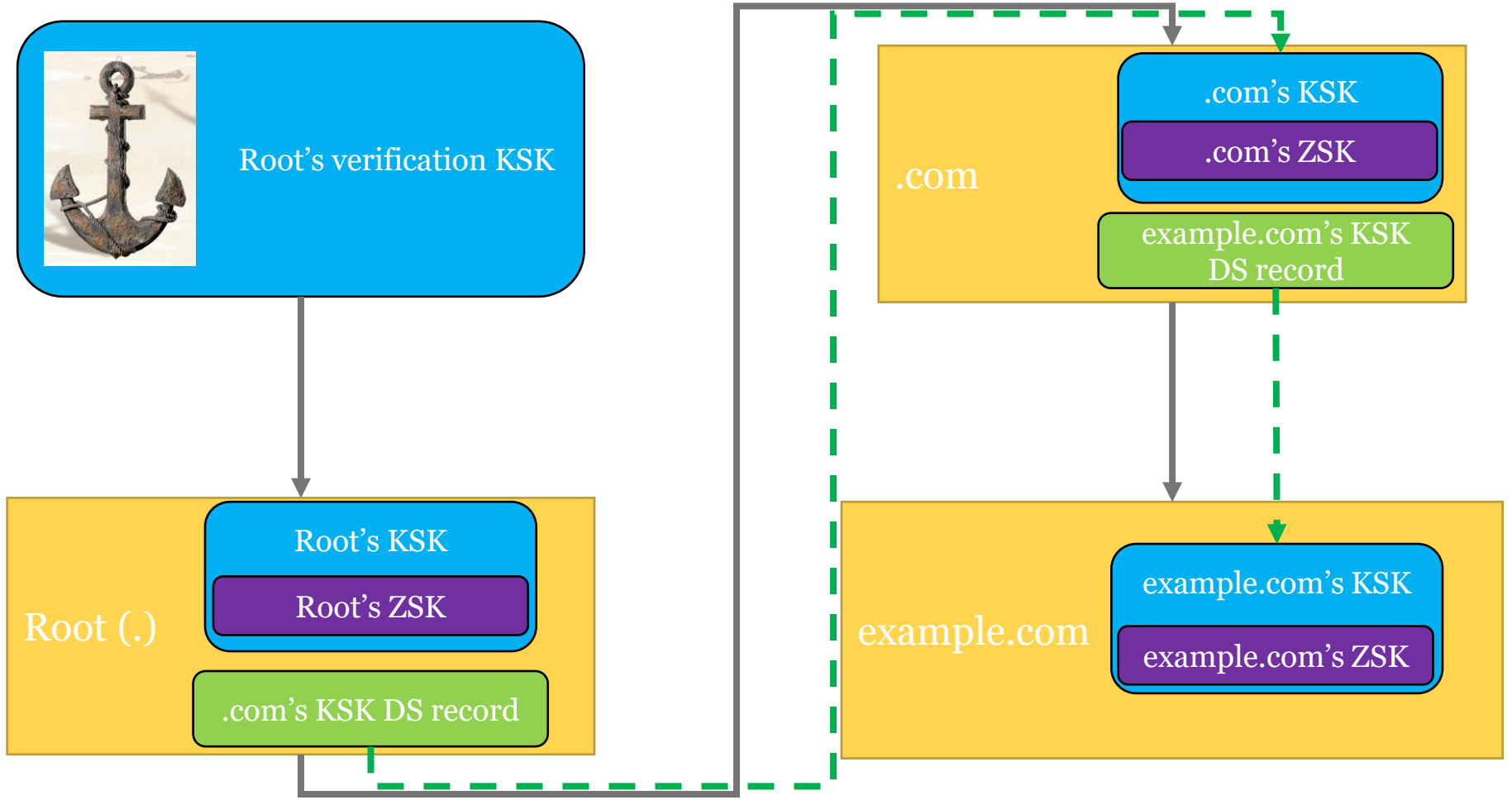
RRSet #2

# HOW DO WE MAINTAIN KEY INTEGRITY?

Construct a chain of trust!

- The root verification KSK must be manually configurated on the machine making the request
- When the root ZSK is queried use the trust anchor to verify key and its signature
- Each zone's parent zone contains a "Delegate signer" (DS) record which is used to verify zone's KSK

# HOW DO WE MAINTAIN KEY INTEGRITY?






UNIVERSITY OF WATERLOO

# POST-QUANTUM CRYPTO IN DNS

- We need to keep messages less than 1232 bytes
  - Prevents (fragile) UDP fragmentation
  - Avoids having to fallback to TCP (expensive)

| Algorithm | Public Key Size | Signature Size |
|---|---|---|
| Falcon-512 | 897 | 666 |
| Dilithium2 | 1,312 | 2,420 |
| SPHINCS+-128s | 32 | 7,856 |

# HOW DO WE SEND LARGE PQC-SIGNED MESSAGES?

Idea: Let’s move fragmentation from UDP to the DNS level

# SENDING LARGE PQC DNS MESSAGES

What if we split large DNS messages into several smaller DNS messages (fragments)?

- Need to pass on extra info such as indicating support for this delivery method
- What if a resource record is **too** large to fit within the 1232 byte requirement?
- What if firewall is configured to only allow a single UDP based DNS response?

# SENDING LARGE PQC DNS MESSAGES

What if we split large DNS messages into several non-fragmented UDP packets and blindly send them to the resolver to reassemble?

- How do we handle out of order delivery?
- What if a firewall is configured to only allow a single UDP packet for DNS messages?
- What if the resolver does not support this fragmentation method?
  - ICMP Flooding (Very bad!)

INTRODUCING

# ARRF: A RESOURCE RECORD FRAGMENTATION MECHANISM

# ARRF

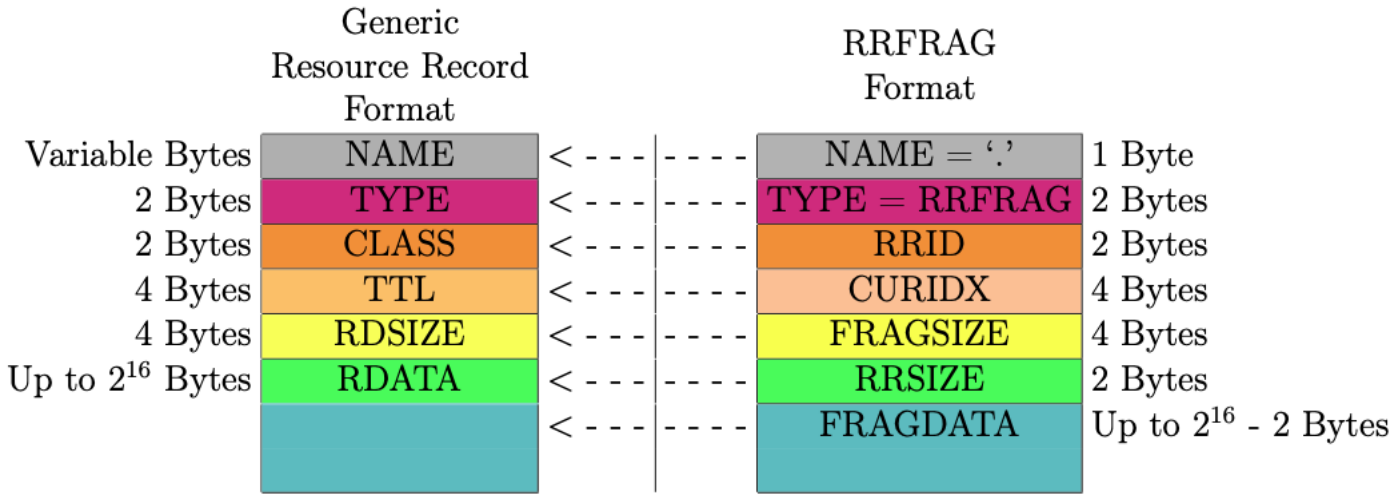
**ARRF is a DNS-level fragmentation mechanism designed with backwards compatibility in mind**

- Rather than breaking up a large DNS message, fragment the problem resource records
- Resolver makes explicit requests for the fragments it wants, so no advertising is required and prevents ICMP flooding
- Does not waste a round trip compared to falling back to TCP
- Can use parallelism to reduce total pain

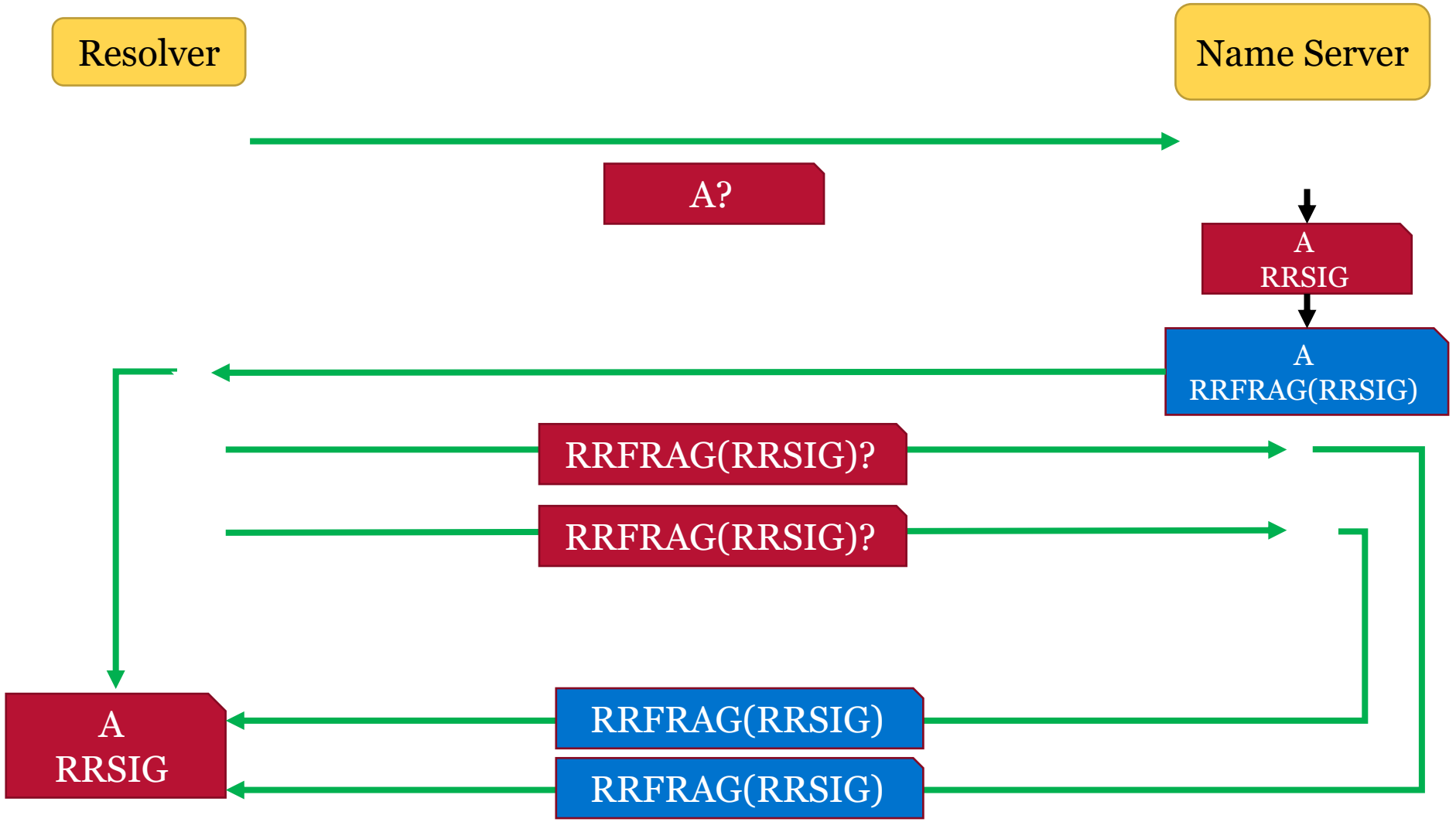
# RRFRAG

When a resource record is too large, it is replaced with a Resource Record Fragment (RRFRAG)

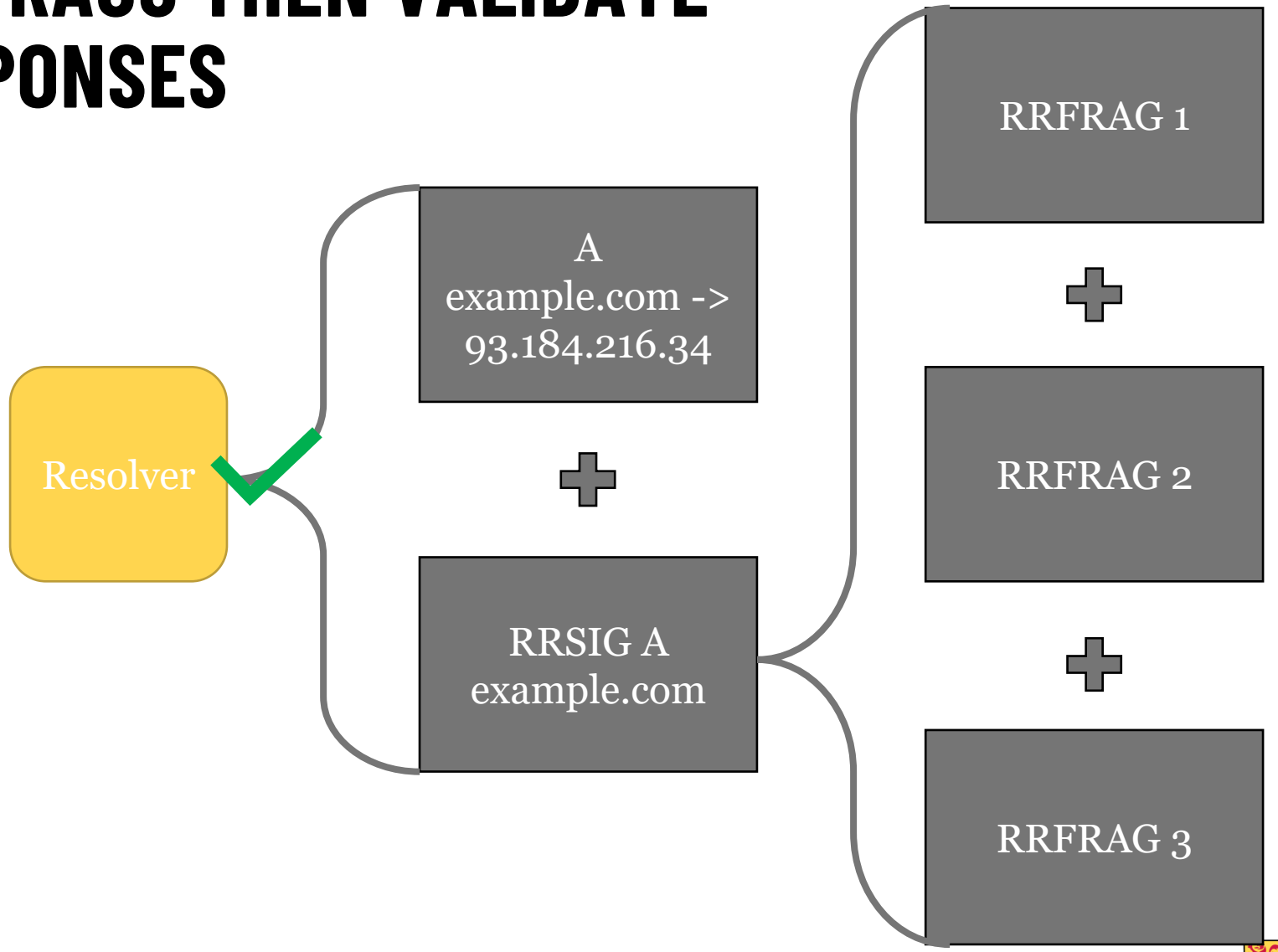
- Constructed to be mapped onto the Generic Resource Record Format to maximize compatibility



# ARRF IN ACTION



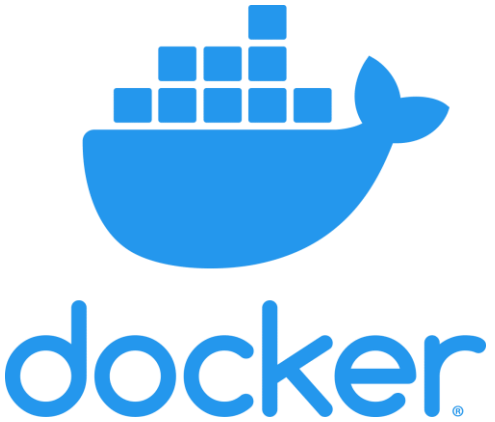
# COMBINE RRFRAGS THEN VALIDATE DNSSEC RESPONSES



# EXPERIMENTAL EVALUATION

- Used Open Quantum Safe's liboqs and OpenSSL fork to add Falcon-512, Dilithium2 and SPHINCS+-SHA256-128s support to Bind9
- Implemented a daemon which runs in front of Bind9 to transparently implement ARRF
- Construct a test DNS network using Docker containers running both Bind9 and ARRF daemon





# EVALUATING PERFORMANCE

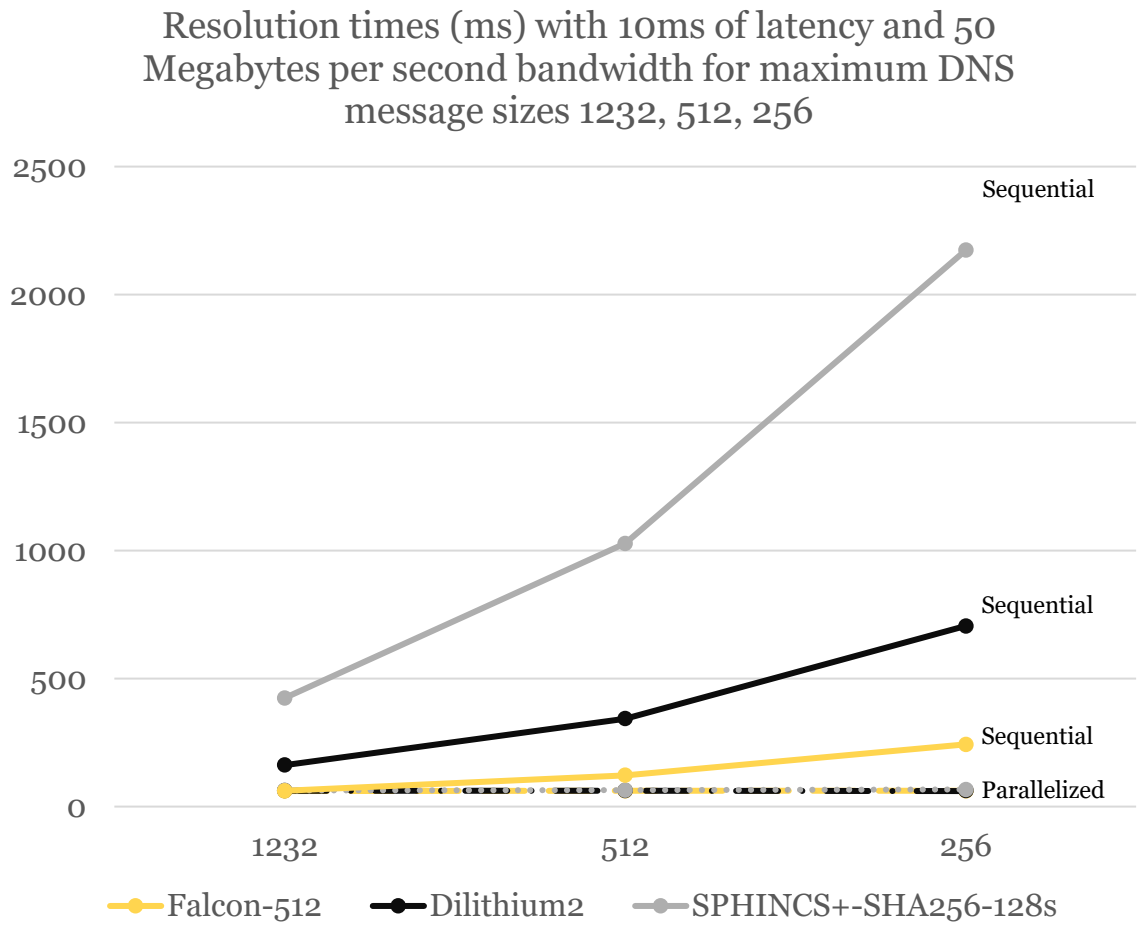
- Perform 1,000 queries for a unique ‘A’ record for each of the three algorithms
- Vary network conditions artificially
  - No delay, no Bandwidth restrictions
  - 10 ms delay 128 KBps bandwidth
  - 10 ms delay 50 MBps bandwidth
  - 100 ms delay 50 MBps bandwidth
- Vary the maximum DNS message size
- Evaluate the algorithms using Standard DNS (with TCP Fallback), sequential ARRF, and a parallelized version of ARRF

# COMPARING LATENCY OF ARRF WITH STANDARD DNS

Resolution times (ms) with 10ms of latency and 50 Megabytes per second bandwidth with a maximum size of 1232 bytes

| Algorithms | Standard DNS with TCP fallback | Parallelized ARRF | Sequential ARRF |
|---|---|---|---|
| Falcon-512 | 82.11 | **61.96** | **62.07** |
| Dilithium2 | 82.24 | **62.52** | 162.9 |
| SPHINCS+-SHA256-128S | 82.59 | **63.45** | 424.7 |
| RSA 2048 with SHA256 | 41.50 | - | - |
| ECDSA P256 | 47.49 | - | - |

# PARALLELIZED ARRF LATENCY SCALES WELL WITH SMALLER PACKET SIZES






UNIVERSITY OF WATERLOO

# DATA OVERHEAD OF ARRF COMPARED TO STANDARD DNS

Total bytes transmitted between resolver and name server during DNS lookup

| Algorithm | Standard DNS with TCP fallback | ARRF | | |
|---|---|---|---|---|
| | | 1232 | 512 | 256 |
| Falcon-512 | 3,112 | 2,557 | 2,947 | 3,637 |
| Dilithium2 | 8,623 | 8,367 | 9,402 | 11,322 |
| SPHINCS+-SHA256-128S | 26,073 | 26,140 | 29,620 | 36,175 |

# ADDRESSING SECURITY CONCERNS

- ARRF does not make DNS based DoS attacks worse
  - All RRFRAGs must be requested, and if a resolver receives one it is not expecting the RRFRAG should be discarded
  - If an RRFRAGs is modified while DNSSEC is being used, a validation failure will occur, which is no worse than a middle box modifying an RRSIG
- DNS Cache Poisoning is not a concern
  - RRFRAG are never to be cached, thus not opening an avenue of attack for cache poisoning
- Memory Exhaustion Attacks
  - ARRF as specified is suspectable to Memory exhaustion attacks due to RRFRAGs not having their integrity ensured

# FUTURE WORK

- Addressing Memory exhaustion attacks
- ARRF was designed for backwards compatibility in mind, but testing on the real internet is needed
- How does ARRF perform on an unreliable network? Would likely need a timeout of some sort
- Does request-based fragmentation work for other protocols?
  - We think so! TurboTLS: https://arxiv.org/abs/2302.05311

# Post-Quantum Signatures in DNSSEC via Request-Based Fragmentation

**Jason Goertzen and Douglas Stebila**

- ARRF is a lightweight, performant, and easy to implement modification to DNS protocol
- ARRF enables transmission of larger post-quantum signatures and keys in DNSSEC
- Removes the need for TCP fallback to send large DNS messages
- Performs 20% better than TCP when using parallelized ARRF
- Uses less data than TCP in cases of light-moderate fragmentation