Achieving Keyless CDNs with Conclaves

Stephen Herwig

Christina Garman     Dave Levin
Content Delivery Networks host their customers’ websites
Content Delivery Networks host their customers’ websites.

CDNs

CDN’s edge server

customer’s origin server

CDNs

Akamai

Cloudflare

fastly
CDNs reduce page load times

CDN's edge server

customer's origin server
CDNs reduce page load times

CDN’s edge server

Customer’s origin server
CDNs mitigate and block attacks

CDN’s edge server

customer’s origin server
CDNs mitigate and block attacks
Customers share their keys with CDNs

CDN's
edge server
Customers share their keys with CDNs

- CDN’s edge server
- Bank’s private key
Key sharing is widespread.

Cangiosi et al., CCS 2016
Key sharing is widespread

43% of the top 10k most popular websites

At least one key shared

Fraction of Domains Hosted on Third-party Providers

Alexa Site Rank (bins of 10,000)

Cangialosi et al., CCS 2016
Key sharing is widespread

43% of the top 10k most popular websites

The web has consolidated keys in the hands of a few CDNs
Keyless SSL

Introduced by Cloudflare to mitigate key sharing
Keyless SSL

Introduced by Cloudflare to mitigate key sharing

Private keys stay at the key server (origin)
Keyless SSL

Introduced by Cloudflare to mitigate key sharing

Private keys stay at the key server (origin)

Key server performs actions requiring private key
Keyless SSL

Introduced by Cloudflare to mitigate key sharing

Private keys stay at the key server (origin)

Key server performs actions requiring private key
Keyless SSL

Introduced by Cloudflare to mitigate key sharing

Private keys stay at the key server (origin)
Key server performs actions requiring private key
Keyless SSL

Introduced by Cloudflare to mitigate key sharing

Private keys stay at the key server (origin)

Key server performs actions requiring private key
Keyless SSL

Introduced by Cloudflare to mitigate key sharing

Private keys stay at the key server (origin)

Key server performs actions requiring private key
Keyless SSL

Introduced by Cloudflare to mitigate key sharing

Private keys stay at the key server (origin)
Key server performs actions requiring private key
Keyless SSL

Introduced by Cloudflare to mitigate key sharing

Private keys stay at the key server (origin)
Key server performs actions requiring private key

The CDN learns all session keys
Keyless SSL

Introduced by Cloudflare to mitigate key sharing

Private keys stay at the key server (origin)
Key server performs actions requiring private key

The CDN learns all session keys
Can we **Maintain privacy** using **Legacy applications** on **Third-party resources**?
Maintain privacy

Legacy applications

Third-party resources

The CDN is no more trusted than a standard on-path attacker
Maintain privacy

Legacy applications

Third-party resources

The CDN is no more trusted than a standard on-path attacker

No changes to existing code-bases; facilitates deployment and adoption
Maintain privacy

The CDN is no more trusted than a standard on-path attacker

Legacy applications

No changes to existing code-bases; facilitates deployment and adoption

Third-party resources

Leverage the existing infrastructure. One additional assumption: TEEs
Maintain privacy

The CDN is no more trusted than a standard on-path attacker

Legacy applications

No changes to existing code-bases; facilitates deployment and adoption

Third-party resources

Leverage the existing infrastructure. One additional assumption: TEEs
Maintain privacy

Legacy applications

Third-party resources

No changes to existing code-bases; facilitates deployment and adoption

Leverage the existing infrastructure.

One additional assumption: TEEs

The CDN is no more trusted than a standard on-path attacker
Trusted execution environments

By default, assume all system components are untrusted
Trusted execution environments

By default, assume all system components are untrusted.

- Application: Code
- Operating System: Service
- Hardware:
  - Small trusted CPU
  - Resistant to physical attacks
Trusted execution environments

By default, assume all system components are **untrusted**

**Enclave**: Isolated application memory

- **Application**
  - Code
  - Enclave

- **Operating System**
  - Service

- **Hardware**
  - Small *trusted* CPU
  - Resistant to physical attacks
Trusted execution environments

By default, assume all system components are untrusted

**Enclave**: Isolated application memory

- **Application**: Code
- **Operating System**: Service
- **Hardware**: Small trusted CPU Resistant to physical attacks

Model: Code and data can safely reside inside an enclave
Practical limitations of TEEs

Applications inside enclaves cannot make syscalls

Diagram:
- Application
- Code
- Enclave
- Operating System
- Service
- Untrusted
- Hardware
libOSes

Idea: Implement a small “OS” inside the enclave
libOSes

Idea: Implement a small “OS” inside the enclave
libOSes

Idea: Implement a small “OS” inside the enclave
libOSes

Idea: Implement a small “OS” inside the enclave
libOSes

Idea: Implement a small “OS” inside the enclave

Enclave

Application

Code

"Syscalls"

libOS

Service

Service locally when possible

Syscalls

Operating System

Service

Hardware

Hardware
Graphene-SGX

A libOS for Intel SGX that supports some services
Graphene-SGX
A libOS for Intel SGX that supports some services

Graphene’s supported services:
✓ fork
✓ exec
✓ pipes, signals, semaphores
Graphene-SGX

A libOS for Intel SGX that supports some services

Graphene’s supported services:
- fork
- exec
- pipes, signals, semaphores

What constitutes a CDN?
- Web server
- Cache
- Web Application Firewall
- Key Server

- Multiple tenants
- Needs disk
- Needs plaintext
- Needs safe storage
Graphene-SGX
A libOS for Intel SGX that supports some services

What constitutes a CDN?

Graphene’s supported services:
- fork
- exec
- pipes, signals, semaphores

Also critical to a CDN:
- Reading & writing files
- Shared memory
- Access to private keys

Multiple tenants
Web server

Needs disk
Cache

Needs plaintext
Web Application Firewall

Needs safe storage
Key Server
The first truly keyless CDN

Conclaves
Containers of enclaves

What constitutes a CDN?

- Web server
- Cache
- Web Application Firewall
- Key Server

Graphene’s supported services:
- fork
- exec
- pipes, signals, semaphores
The first truly keyless CDN

Conclaves: Containers of enclaves

What constitutes a CDN?

- Multiple tenants
  - Web server
- Needs disk
  - Cache
- Needs plaintext
  - Web Application Firewall
- Needs safe storage
  - Key Server

Graphene’s supported services:
- fork
- exec
- pipes, signals, semaphores

Also critical to a CDN:
- Reading & writing files
- Shared memory
- Access to private keys
Enclave

Web server
Cache
Web Application
Firewall

Phoenix

The first truly keyless CDN

Conclaves
Containers of enclaves

Enclave

Web server
Cache
Web Application
Firewall

Enclave

Key Server

Insight: Treat enclaves like a distributed system
Implement services using kernel servers
Enclave

Web server
Cache
Web Application
Firewall

Enclave

Web server
Cache
Web Application
Firewall

Enclave

Web server
Cache
Web Application
Firewall

Enclave

Web server
Cache
Web Application
Firewall

Enclave

Key Server

Enclaves mutually authenticate via attested TLS

Insight: Treat enclaves like a *distributed system*
Implement services using *kernel servers*

Phoenix: The first truly keyless CDN

Conclaves: Containers of enclaves

Knauth et al., 2018
Insight: Treat enclaves like a *distributed system*. Implement services using *kernel servers*.

*Phoenix*  The first truly *keyless CDN*

*Conclaves*  Containers of enclaves

Enclaves mutually authenticate via attested TLS

Knauth et al., 2018
Enclave

Web server
Cache
Web Application
Firewall

Phoenix
The first truly keyless CDN

Conclaves
Containers of enclaves

Enclaves mutually authenticate via attested TLS

Insight: Treat enclaves like a distributed system
Implement services using kernel servers

Knauth et al., 2018
Phoenix

The first truly keyless CDN

Conclaves

Containers of enclaves

Insight: Treat enclaves like a *distributed system*
Implement services using *kernel servers*
The first truly keyless CDN

Containers of enclaves

Insight: Treat enclaves like a distributed system
Implement services using kernel servers
Phoenix: The first truly keyless CDN

Conclaves: Containers of enclaves

Insight: Treat enclaves like a distributed system
Implement services using kernel servers
Phoenix  The first truly keyless CDN

Conclaves  Containers of enclaves

Insight: Treat enclaves like a **distributed system**
Implement services using **kernel servers**
Shared memory

Enclave

Application  `fcntl()`

"Syscall"

libOS  Service

Operating System  Service

Hardware

Conclaves
Conclaves

Shared memory

Enclave

Application  `fcntl()`

"Syscall"

libOS  Service

Enclave

Memory Server

Operating System  Service

Hardware

Hardware
Application

ließlich

"Syscall"

Enclave

LibOS

Service

fcnt1()
Application

libOS

Enclave

Service

"Syscall"

fcntl()

Enclave Memory Server

Coordinates locks
Maintains memory locations

Operating System

Service

Hardware

Conclaves
Shared memory
Application
libOS

Enclave

Application fcntl()

"Syscall"

libOS Service

Enclave

Operating System Service

Hardware

Conclaves
Shared memory

Enclave Memory Server

Coordinates locks
Maintains memory locations
Application

libOS

Enclave

"Syscall"

fcntl()

Service

Syscall

Operating System

Hardware

Memory Server

Conclaves

Shared memory

Coordinates locks
Maintains memory locations
Application `fcntl()`

"Syscall"

libOS Service

Syscall

Operating System Service

Hardware

Memory file
Encrypted on untrusted disk

Enclave

Memory Server
Coordinates locks
Maintains memory locations

Conclaves
Shared memory
Application fcntl() → "Syscall"

libOS Service

"Syscall"

Operating System Service

Hardware

Memory file

Encrypted on untrusted disk

Conclaves

Shared memory

Enclave

Memory Server

Coordinates locks
Maintains memory locations
Shared memory

- **Application**
  - `fcntl()`
  - Shared Memory

- **libOS**
  - Service

- **Operating System**
  - Service

- **Hardware**
  - Memory file
    - Encrypted on untrusted disk

**Enclave**
- Memory Server
  - Coordinates locks
  - Maintains memory locations
Phoenix: The first truly keyless CDN

Conclaves: Containers of enclaves

Insight: Treat enclaves like a distributed system
Implement services using kernel servers
Enclave
Web server
Cache
Web Application
Firewall

Enclave
Key Server

Enclave
Memory Server

Enclave
File Server

Insight: Treat enclaves like a distributed system
Implement services using kernel servers
The first truly keyless CDN

Containers of enclaves

Insight: Treat enclaves like a distributed system
Implement services using kernel servers
Phoenix: The first truly keyless CDN

Conclaves: Containers of enclaves

Insight: Treat enclaves like a distributed system
Implement services using kernel servers
Phoenix

The first truly keyless CDN

Conclaves

Containers of enclaves

Insight: Treat enclaves like a distributed system
Implement services using kernel servers
Conclaves

File system access

Enclave

Application Code

libOS Service

Enclave File Server

Operating System Service

Hardware

Merkle Tree

Encrypted on untrusted disk
**Application**

```
read()
```
**Application**

**libOS**

```
read()
```

**Syscall**

**Enclave**

**File Server**

**Operating System**

**Hardware**

**Merkle Tree**

Encrypted on untrusted disk
Conclaves

File system access

Enclave

Application

libOS

Service

"Syscall"

RPC

File Server

Operating System

Service

Hardware

Merkle Tree

Encrypted on untrusted disk

read()
Enclave

Application

"Syscall"

read()

libOS

Service

RPC

Enclave

ext2fs server

Block layer

Merkle root

libOS

Operating System

Service

Hardware

Merkle Tree

Encrypted on untrusted disk
Conclaves

File system access

Application

read()

"Syscall"

libOS

Service

Enclave

RPC

ext2fs server

Block layer

Merkle root

Operating System

Service

LibOS

Syscall

Hardware

Merkle Tree

Encrypted on untrusted disk
Conclaves

File system access

Enclave

Application

libOS

Service

"Syscall"

RPC

ext2fs server

Block layer

Merkle root

libOS

Operating System

Service

Hardware

File system access

Merkle Tree

Encrypted on untrusted disk
File system access

Conclaves

Enclave

Application

libOS

Service

"Syscall"

Enclave

ext2fs server

Block layer

Merkle root

libOS

RPC

Syscall

Operating System

Service

Operating System

Service

Hardware

Merkle Tree

Encrypted on untrusted disk
**File system access**

**Enclave**
- **Application**: read()
  - "Syscall"
- **libOS**

**RPC**
- Verifies branches
- Decrypts blocks
- **Enclave**
  - ext2fs server
  - Block layer
  - Merkle root

**Operating System**
- **Service**

**Hardware**
- **Merkle Tree**
  - Encrypted on untrusted disk
Conclaves

File system access

Enclave

Application

libOS

Service

"Syscall"

RPC

Enclave

ext2fs server

Block layer

Merkle root

Operating System

Service

libOS

Syscall

Hardware

Merkle Tree

Encrypted on untrusted disk
Conclaves

File system access

Enclave

Application
read()
Data

libOS
Service

Operating System
Service

Hardware

Merkle Tree
Encrypted on untrusted disk

Verifies branches
Decrypts blocks

Enclave

ext2fs server

Block layer
Merkle root

libOS
Enclave

Web server

Cache

Web Application

Firewall

Enclave

Key Server

Memory Server

File Server

Execution environment is a distributed system of enclaves
Execution environment is a distributed system of enclaves.
Enclave

Web server
Cache
Web Application
Firewall

Enclave
Key Server
Memory Server
File Server

Conclaves supported services:
- fork
- exec
- pipes, signals, semaphores
- Reading & writing files
- Shared memory
- Access to private keys
- Trusted time server

Execution environment is a distributed system of enclaves
Phoenix The first truly keyless CDN
Phoenix
The first truly keyless CDN

Supports multi-tenancy
Both CDN and website can store private data

Other details in the paper
Websites delegate provisioning to CDNs
Phoenix supports many deployment configurations
Phoenix: The first truly keyless CDN

Supports multi-tenancy
Both CDN and website can store private data

Other details in the paper
Websites delegate provisioning to CDNs
Phoenix supports many deployment configurations
Phoenix The first truly keyless CDN

Implemented on top of Graphene-SGX
Evaluated to understand throughput and scalability
What is Phoenix’s request throughput?

Fetch a file 10,000 times over non-persistent HTTPS connections from among 128 concurrent clients.
What is Phoenix’s request throughput?

Fetch a file 10,000 times over non-persistent HTTPS connections from among 128 concurrent clients.

What is Phoenix’s request throughput?
What is Phoenix’s request throughput?

Fetch a file 10,000 times over non-persistent HTTPS connections from among 128 concurrent clients.

- Confidentiality
- Confidentiality & integrity

Throughput (requests/sec)

- Linux
- Phoenix-crypt
- Phoenix-vericrypt

Downloaded file size

- 1 KiB
- 10 KiB
- 100 KiB

# Workers

Throughput (requests/sec)

- 1
- 2
- 4
- 8

0

1000

2000

3000

4000

5000

6000

7000

8000

9000

10000

11000

12000

13000

14000

15000

16000

17000

18000

19000

20000

21000

22000

23000

24000

25000

26000

27000

28000

29000

30000

31000

32000

33000

34000

35000

Downloaded file size
How does Phoenix scale to multiple tenants?

![Graph showing time per request (ms) vs. number of tenants]

- Linux (shared NGINX)
  - 1 tenant: 40 ms
  - 2 tenants: 264 ms
  - 4 tenants: (bar not shown)
  - 6 tenants: (bar not shown)
How does Phoenix scale to multiple tenants?

The diagram compares the time per request (ms) for different numbers of tenants using two different systems: Linux (shared NGINX) and Phoenix-crypt (shared nothing).

- **Linux (shared NGINX)**
  - 8 tenants: 40 ms
  - 16 tenants: 127 ms
  - 32 tenants: 326 ms
  - 48 tenants: 1437 ms

- **Phoenix-crypt (shared nothing)**
  - 8 tenants: 127 ms
  - 16 tenants: 264 ms
  - 32 tenants: 481 ms

The diagram shows that Phoenix-crypt maintains a lower time per request as the number of tenants increases compared to Linux (shared NGINX).
How does Phoenix scale to multiple tenants?

The diagram shows the performance of Phoenix in different configurations with varying numbers of tenants and enclaves. The x-axis represents the number of tenants, while the y-axis shows the time per request in milliseconds. The configurations include:

- Linux (shared NGINX)
- Phoenix-crypt (shared NGINX)
- Phoenix-crypt (shared nothing)

Key observations:

- With 1 tenant, Phoenix-crypt (shared nothing) has a low latency of 40 ms.
- As the number of tenants increases, the time per request also increases for all configurations.
- Phoenix-crypt (shared nothing) consistently shows lower latency than the other configurations for all numbers of tenants and enclaves.

The diagram highlights the efficiency of Phoenix in managing multiple tenants and enclaves without compromising performance.
How does Phoenix scale to multiple tenants?

- **Linux (shared NGINX)**
- **Phoenix-crypt (shared NGINX)**
- **Phoenix-crypt (shared nothing)**

<table>
<thead>
<tr>
<th>Number of tenants</th>
<th>Time per request (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40 ms</td>
</tr>
<tr>
<td>2</td>
<td>128 ms</td>
</tr>
<tr>
<td>4</td>
<td>264 ms</td>
</tr>
<tr>
<td>6</td>
<td>806 ms, 1437 ms</td>
</tr>
</tbody>
</table>

- Number of enclaves:
  - 8

- SGX paging events:
  - 1K
  - 10K
  - 1M
  - 10M
  - 100K
  - 1M
  - 10M
Other results

Benchmark overhead of running WAFs (ModSecurity) in SGX (overhead about the same as in Linux)

Perform detailed micro-benchmarks of each kernel server

Compare standard ocalls to exitless ocalls (not always better)
**Phoenix**  The first truly *keyless* CDN

**Conclaves**  Containers of *enclaves*

---

Run legacy apps in *enclaves*

**Moderate performance overheads**

Throughput (requests/sec)

<table>
<thead>
<tr>
<th># Workers</th>
<th>1 KiB</th>
<th>10 KiB</th>
<th>100 KiB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Downloaded file size

- Linux
- Phoenix-crypt
- Phoenix-vericrypt

**https://phoenix.cs.umd.edu/**