Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

A distributed deployment model for Encrypted Client Hello

Ted Johnson

Supervisor: Dr Stephen Farrell

April 2024

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Master in Computer Science (MCS)

Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at http://tcd-ie.libguides.com/plagiarism/ready-steady-write.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

Signed: _____ Date: _____

A distributed deployment model for Encrypted Client Hello

Ted Johnson, Master in Computer Science

University of Dublin, Trinity College, 2024

Supervisor: Stephen Farrell

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Acknowledgements

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Ted Johnson

University of Dublin, Trinity College
April 2024

Contents

List of Figures

List of Listings

1 Introduction

Encrypted Client Hello (ECH) is a proposed extension to the Transport Layer Security protocol version 1.3 (TLS 1.3) which has begun to see implementation and adoption on the Internet [1, 2]. ECH seeks to allow encryption of the ClientHello message, which can contain potentially sensitive information such as the Service Name Indication (SNI) and Application-Layer Protocol Negotiation (ALPN) extensions. This is partially achieved through serving many private domains behind a common provider to form an anonymity set that conceals the true domain requested by the client.

Due to this, ECH introduces significant centralisation to the Internet. This paper presents a practical model for the distributed deployment of ECH amongst several co-operating TLS servers, where each server operates both as the origin server of its own domains as well as an ECH provider for other participating servers. The model addresses a number of implementation challenges, predominately related to ensuring the security of the protocol is not compromised and minimising the performance impact to the connection while strengthening service availability.

Included in this paper is a review of the background technology and concepts relevant to the discussion of the deployment model. This is followed by a study of the model's design and the complications which influenced it. We then see how this design can be implemented within a practical scenario and discuss some of its deployment considerations. In the subsequent chapter, an analysis and criticism of both the results taken from this implementation and the design as a whole is used to assess the quality of the solution. Finally, I conclude the report with a summary of the work completed and delineate where future contributions could best benefit the further development of the deployment model.

1.1 Motivation

Mark Nottingham has previously cautioned against the introduction of centralisation through Internet standards [3]. Of particular relevance to ECH is his highlight of the adverse effect centralisation can have on infrastructure resilience and service availability through

reliance on a single entity. This is especially detrimental to ECH where the effectiveness of its anonymity set grows with the number of private domains served by a single provider. Nottingham also writes on susceptibility of centralisation to stifle innovation and induce an unhealthy monoculture which may result in less overall technological progress and robustness of the ECH protocol.

Additionally, allowing entirely independent servers to co-operate from across the Internet to provide ECH support for each other enables several distinct organisations to work together to offer improved privacy for their users without the requirement for co-located servers nor the dependence of any on the availability on another. Consider here global networks of whistleblower services, investigative journalists and human rights non-profit organisations who share an interest in protecting the confidentiality of their members and users from persecution and retaliation.

For these reasons, the development of a model for the distributed deployment of ECH across several co-operating providers is a key step towards its application in <perilous> scenarios and broad adoption throughout the Internet.

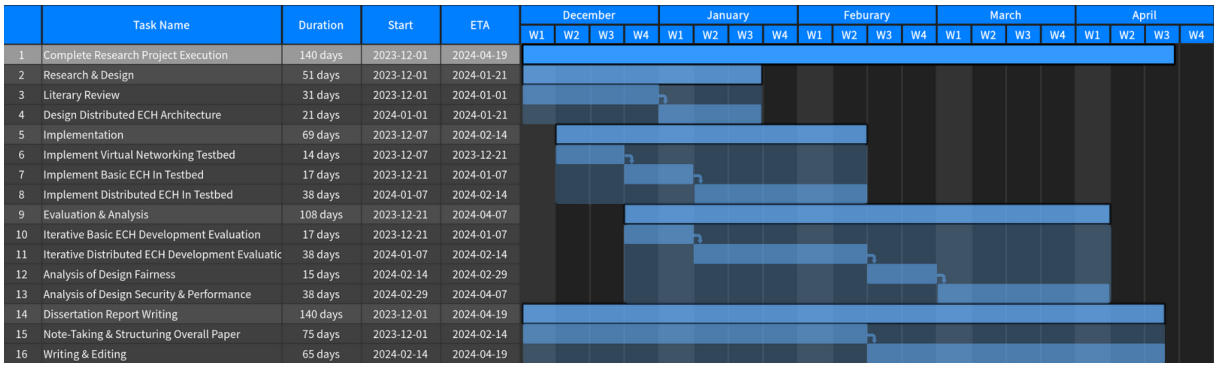
1.2 Project Objectives

The objectives of this research project can be summarised with the following question: "How can Encrypted Client Hello be deployed fairly amongst co-operating Transport Layer Security servers to reduce network centralisation without compromising the security of the protocol?" This task is composed of the following objectives:

1. **Identify principal challenges and appropriate solutions.** Before development can begin proper, we must first understand the environment the system would operate in and explore the technical and logistical issues it might face to determine the dominant criteria for design. We accomplish this through research and experimentation of the functioning of the protocol and its surrounding technologies.
2. **Design, evalute and contrast deployment models.** An iterative development process is used to produce a series of incrementally improved skeletal prototypes, with the goal to rapidly design and test for functionality guided by the design criteria and results of previous work as heuristics.
3. **Analyse model implementatons through simulation.** Promising design solutions are fleshed out into full implementations within deterministic, reproducible and quantifiable simulated environments, where security and performance implications can be easily isolated and compared. It is expected that unforeseeable practical challenges and considerations are to be unveiled during this work.

4. **Conclude findings.** <Learnings from comparisons> <As part of this, a prototype script is to be compiled from the implementation snippets>

In preparation for undertaking this project, I had produced the Gantt chart included in Fig. 1.2.1 to help gauge my progress during the four months of work. While I found more emphasis has been placed on implementation, the overall task structure of the timeline has been followed reasonable well.



	Task Name	Duration	Start	ETA
1	Complete Research Project Execution	140 days	2023-12-01	2024-04-19
2	Research & Design	51 days	2023-12-01	2024-01-21
3	Literary Review	31 days	2023-12-01	2024-01-01
4	Design Distributed ECH Architecture	21 days	2024-01-01	2024-01-21
5	Implementation	69 days	2023-12-07	2024-02-14
6	Implement Virtual Networking Testbed	14 days	2023-12-07	2023-12-21
7	Implement Basic ECH In Testbed	17 days	2023-12-21	2024-01-07
8	Implement Distributed ECH In Testbed	38 days	2024-01-07	2024-02-14
9	Evaluation & Analysis	108 days	2023-12-21	2024-04-07
10	Iterative Basic ECH Development Evaluation	17 days	2023-12-21	2024-01-07
11	Iterative Distributed ECH Development Evaluatic	38 days	2024-01-07	2024-02-14
12	Analysis of Design Fairness	15 days	2024-02-14	2024-02-29
13	Analysis of Design Security & Performance	38 days	2024-02-29	2024-04-07
14	Dissertation Report Writing	140 days	2023-12-01	2024-04-19
15	Note-Taking & Structuring Overall Paper	75 days	2023-12-01	2024-02-14
16	Writing & Editing	65 days	2024-02-14	2024-04-19

Figure 1.2.1: Predicted timeline of project as of the 7th of December, 2023

1.3 Research Contributions

This work provides evidence for the viability of the distributed deployment of ECH between co-operative TLS servers. It supports its argument that the deployment model presented does not compromise the security of the protocol and minimises impact to network performance though analysis of the data produced by implementations within simulated networking environments. Additionally, an evaluation of several traffic masking and normalisation techniques is given to serve as the bases for further work on disrupting traffic correlation attacks applicable to ECH and elsewhere. Finally, the delivered project may also contribute academic value as a deterministic, reproducible tutorial on the deployment and operation of ECH using commonplace software and tooling.

2 Background

This chapter offers an overview of the technology and concepts needed to understand the context and relevance of the work within the broader world. The review is conducted predominately through a networking, security and privacy perspective to best highlight the aspects pertinent to the distributed deployment of ECH. This chapter also represents the bulk of the effort put into investigating and studying the functioning of ECH while identifying and experimenting with different deployment models.

The contents of this chapter include a high level description of the Transport Layer Security protocol and the Domain Name System with a more detailed look at the components that enable ECH functionality. This is followed by an inspection of ECH itself, its security properties and the mechanisms which allow for distributed deployment. Finally, we survey how a variety of traffic analysis techniques that can be used to infer sensitive information from patterns in network activity, as well as the countermeasures which exist to mask these patterns.

2.1 Transport Layer Security

Transport Layer Security (TLS) is a cryptographic protocol proposed by the Internet Engineering Task Force (IETF) which enables secure network communication over public networks. Applications and services can establish an encrypted communication channel to transmit private information such that confidentiality, integrity and authenticity of the data can be ensured. TLS is commonly used to protect Internet traffic, having seen widespread adoption and several revisions since its original inception in 1999, superseding the Secure Sockets Layer (SSL) specification previously defined by Netscape Communications from 1994 [4–6].

TLS is designed to operate on top of a reliable transmission protocol between a client and server, such as the Transmission Control Protocol (TCP) when used over the Internet. In order to prevent eavesdropping, tampering and message forgery, TLS includes a number of security features based on a number of cryptographic mechanisms:

Confidentiality: All service and application data exchanged between the client and server is encrypted as to make it indecipherable to any intermediate party which might be intercepting their communication. For example, consider the importance of protecting customer passwords and banking information when accessing financial services.

Data integrity: In a similar manner, cryptographic properties are used to guarantee transferred data cannot be modified during transmission. This is critical for safeguarding against input manipulation in consequential situations, such as while conducting a monetary transaction.

Authentication: TLS provides the ability for both participants to verify the identity of the other, ensuring privileged communication is only performed with the intended recipient. Such a condition is fundamental for establishing trust and confidence in sensitive environments, as is required when interacting with online financial institutions.

TLS 1.3 is the latest defined standard for the protocol, having been published in August 2018 and contributing to the deprecation of TLS 1.0 and TLS 1.1 in March 2021 [7, 8]. <deployment stats>. It introduces many major changes to TLS 1.2, including the addition of a zero round trip time resumption (0-RTT) mode, further encryption and optimisation of the handshake and removal of outdated cryptographic algorithms and security mechanism with all key exchanges now providing forward secrecy. A change of particular relevance to ECH is the encryption of the digital certificate received by the client to authenticate the server.

2.1.1 Digital Certificates

TLS uses digital certificates to make assertions on the identity of entities within the network using a chain of trust model, and are of particular significance during the exchange of public keys. <what public key used for> <"the TLS connection handshake would be susceptible to man-in-the-middle attacks"> <therefore trust must be established>

<chain of trust model is a hierarchical structure of certificates, Fig 2.1.1> <ensures each certificate in the chain is issued by a trusted Certificate Authority (CA)> <root ca is inherently trusted> <These certificates contain public key, parent name, signature and other metadata>

<root ca typically installed by operating system> <it is possible to add your own> <typically only server is authenticated, it sends its cert and public key during the handshake> <its certificate is verified by the client>

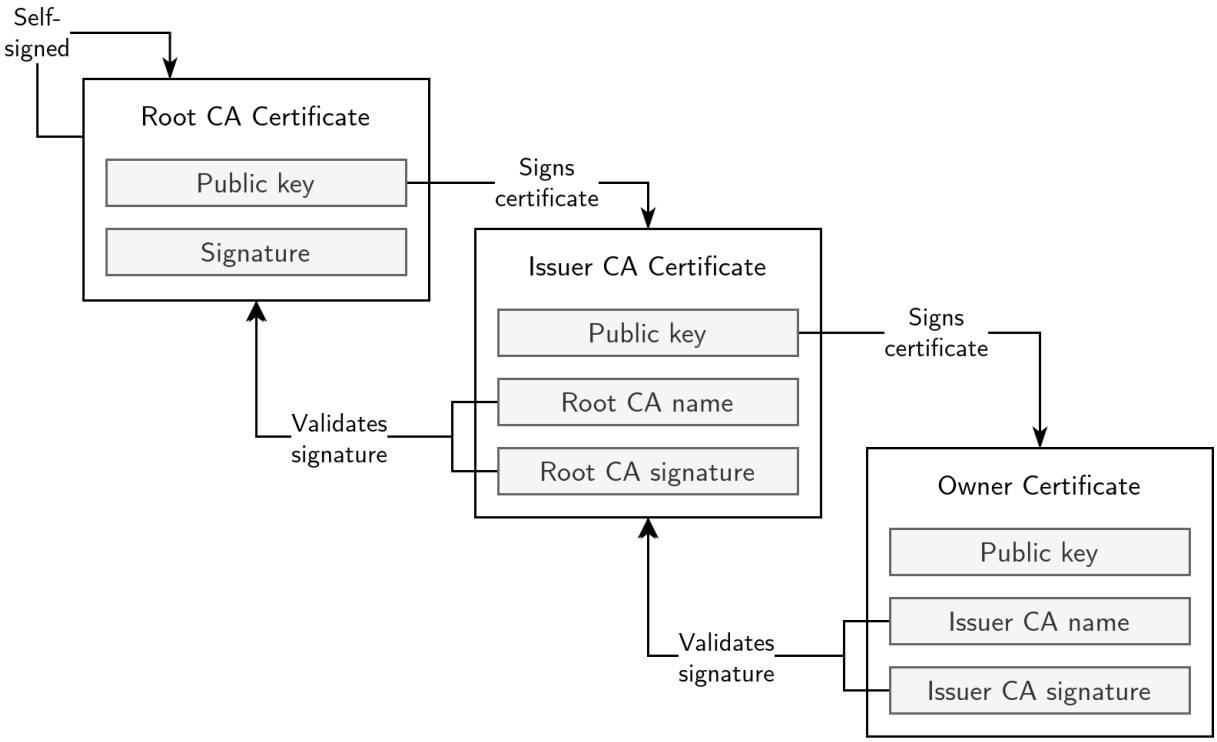


Figure 2.1.1: <>

2.1.2 TLS 1.3 Handshake

<the tls handshake is a procedure executed between the client and server to complete establishment> <tls 1.3 was designed to improve the security and performance of the handshake over 1.2 while reducing overall complexity> <Fig. 2.1.2> <application data>

<The handshake consists of a number of required and optional messages with some plain and others encrypted:>

ClientHello: <clienthello contents+purpose>

ServerHello: <serverhello contents+purpose>

Certificate: <optional certificate, now encrypted>

Finished: <finish protocol>

<in addition to these, the handshake offers extensions>

2.1.3 Extensions

<to remain adaptable, tls 1.3 permits numerous extensions to be inserted during the handshake for additional features and capabilities> <some are mandatory in 1.3 and others are very commonly used> <examples: ems, sni, alpn... and problems they solve> <wider

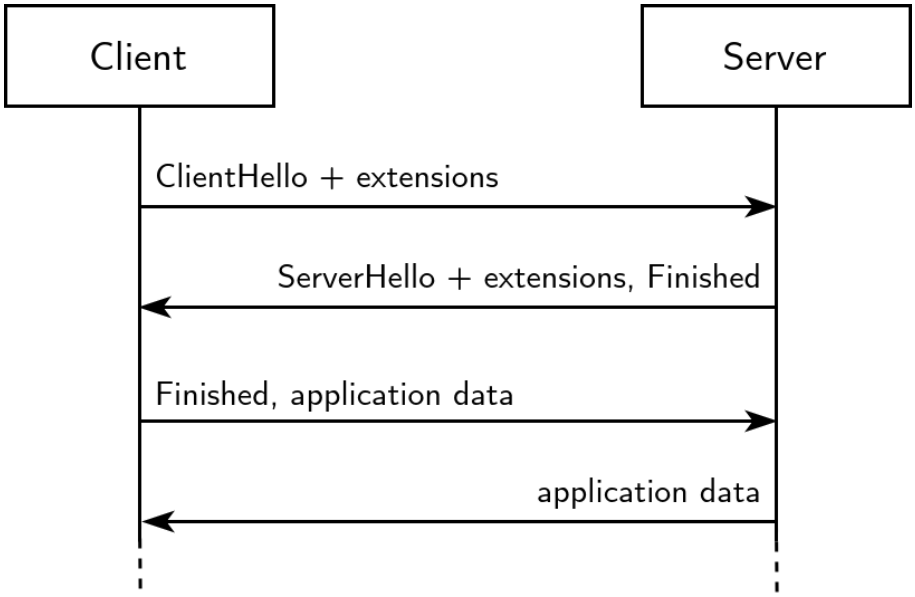


Figure 2.1.2: <>

range of use cases and accommodating evolving requirements> <examples: ech, ...> <problems> <greasing>

2.2 The Domain Name System

<history of ip, why dns needed> <the domain name system (dns) was introduced to add alphanumeric domain names> <originally a big host file> <it now operates as a decentralised hierarchical naming system accessible by networked devices> <registration consists of a number of entities registrars, registrants, registries> <name server>

2.2.1 Name Resolution Process

<consists of root ns, tld ns, authoritative ns> <Fig. 2.2.1> <stub resolver queries local recursive resolver> <if not cached, asks root-tld-etc>

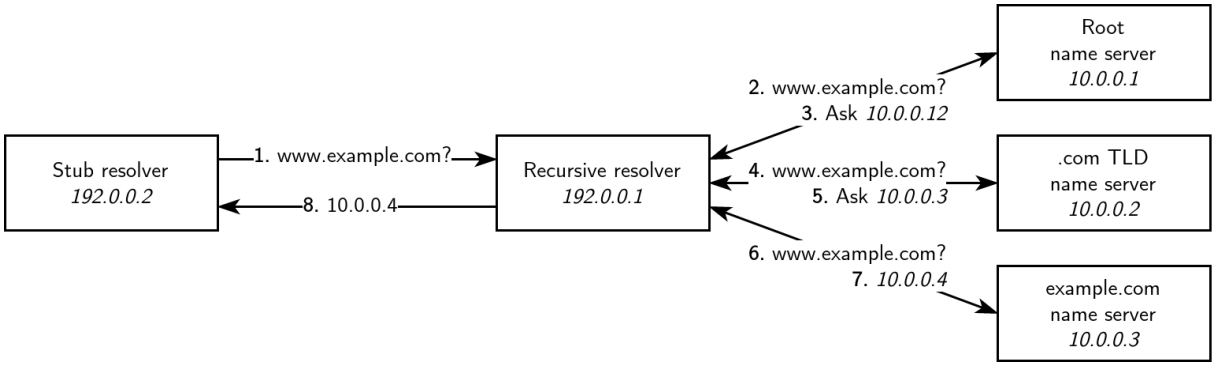


Figure 2.2.1: <example dns query resolution process for www.example.com>

<dns protocol>

2.2.2 DNS Over HTTPS

<security was previously not a concern as the information was considered public> <passive attacks considered bad [**todo**]> <introduction of dot> <problems> <introduction of doh> <doh security>

2.2.3 The HTTPS Resource Record

<these days a number of resource records exist for more advanced functionality and adaptability> <new rr have been added to support performace and some fixes[**https-rr**]> <this has also allowed additional data to be accociated with a host> <ech uses this as the primary mechanism to transmit echconfig out-of-band to clients>

2.3 Encrypted Client Hello

<why need ech> <what ech is "Encrypted Client Hello (ECH) is a proposed extension for TLS 1.3 which has begun to see implementation and adoption on the Internet [1, 2]"> <what ech does "ECH seeks to allow encryption of the ClientHello message, which can contain potentially sensitive information such as the Service Name Indication (SNI) and Application-Layer Protocol Negotiation (ALPN) extensions." <how ech does this "This is partially achieved through serving many private domains behind a common provider to form an anonymity set that conceals the true domain requested by the client.">

The client-facing server first generates an ECH encryption key pair and some associated metadata. This public key and metadata, referred to as an ECH configuration or ECHConfig, may then be shared out-of-band to ECH-enabled clients though secure means like DoH. A client may then use this to construct a ClientHello message, named the ClientHelloOuter, holding unremarkable values for the client-facing server along side an encrypted ClientHello, named the ClientHelloInner, itself holding the real values for a private domain. To establish a TLS connection to the origin server of this domain, the client sends the ClientHelloOuter to the client-facing server, which decrypts and relays the contained ClientHelloInner to the origin server, which itself completes the TLS handshake with the client through the client-facing server.

<in this way, the true domain is concealed>

2.3.1 Hybrid Public Key Encryption

<much like tls itself, the ech extension uses public key cryptography> <shown to be secure>[9] <HPKE explanation>

2.3.2 Split Mode Deployment

<ech defines two modes of network topology> <shared mode is when the client-facing server and backend server are co-located, and is not of interest to us> <split mode is when the client-facing server and backend server are in different boxes pysically separateed> <Fig. 2.3.1>

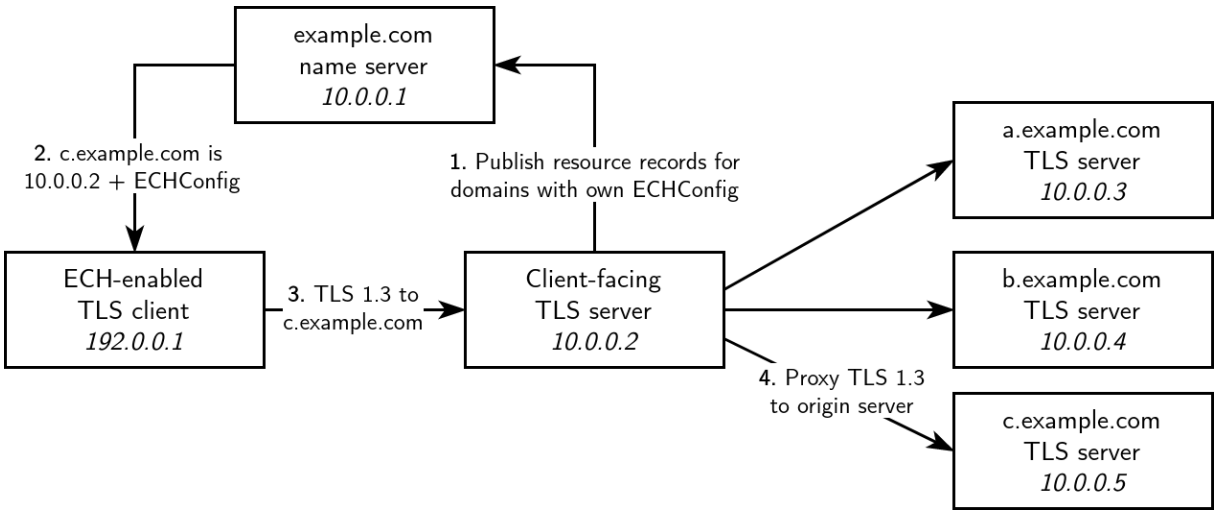


Figure 2.3.1: <>

<this is useful for x,y,z...> <distributed deployment explanation (as opposed to centralised and decentralised)> <however it is susceptible to attacks>

2.4 Traffic Analysis

<what is traffic analysis> <traffic analysis techniques that can be used to infer sensitive information from patterns in network activity> <exploits information unintentionally leaked by the system aka side-channel attack> <example of the act of communication being information itself> <there are a lot of other techniques that can be used: packet inspection>

2.4.1 Correlation Attacks

<is a large category of attacks identified by correlating communication patterns> <can be used to break anonymity, very relevant to ech split mode> <examples: timing, packet size, packet count, packet rate+pattern>

2.4.2 Countermeasures

<to disrupt the patterns> <effectiveness against practicality> <examples: morphing, mixing/pacing and padding, normalisation>

2.5 Summary

<tls and dns continue to evolve with shifting requirements> <ech has been enabled due to this, which allows for the encryption of the clienthello> <ech split mode permits the client-facing server to be pysically separate from the backend server> <this reveals potentially attack surface through traffic correlation, which must be distributed with various countermeasures>

3 Design

<this disseration aims to serve as a guide for security researchers and service operator on the viablity of distributed ech> <during the course of this project, a system design was formulated and iteratively refined> <the main parts of the design considered are its deployment schema and protecting traffic>

<in this chapter, we will study the determined solution as well as the challenges that motivated its design> <this consists of a overview of purposed system followed by an examination of its individual components>

3.1 Problem Overview

<ech split mode serves as a bases for distributed deployment, but makes no attempt to address the implications of servers being operated by separate organisations and located across a public network> <to do so, we must consider a number of challenges faced by this situation to enable co-operation and secure functioning> <this paper purposes a loose network of tls servers all acting as ech providers for each other which proxy connections to the true origin server, as depicted in Fig. 3.1.1>

<parallel to the operation of ech split mode> <loose network must publish appropriate resource records such that clients choose to query any of them for any of their domains> <all providers listens on their public facing network interface for with ClientHelloOuter> <providers also listens on a virtual private network interface they are all a part of for actual connections> <clients resolve requested domain using DoH and then use round robin or other selection process to uniformly choose one of the providers at random with the correct echconfig> <client begins establishment to the requested via the selected provider using the providers echconfig> <as in ech split mode, the provider decrypts the ClientHelloInner to retrieve the actual domain requested> <the provider maps the domain to the true origin server and proxies the clients connection over a backend communication network to the origin server> <the origin server listening on the vpn interface completes the tls handshake with the client through the provider> <this connection is end-to-end encrypted, so the provider does not mitm>

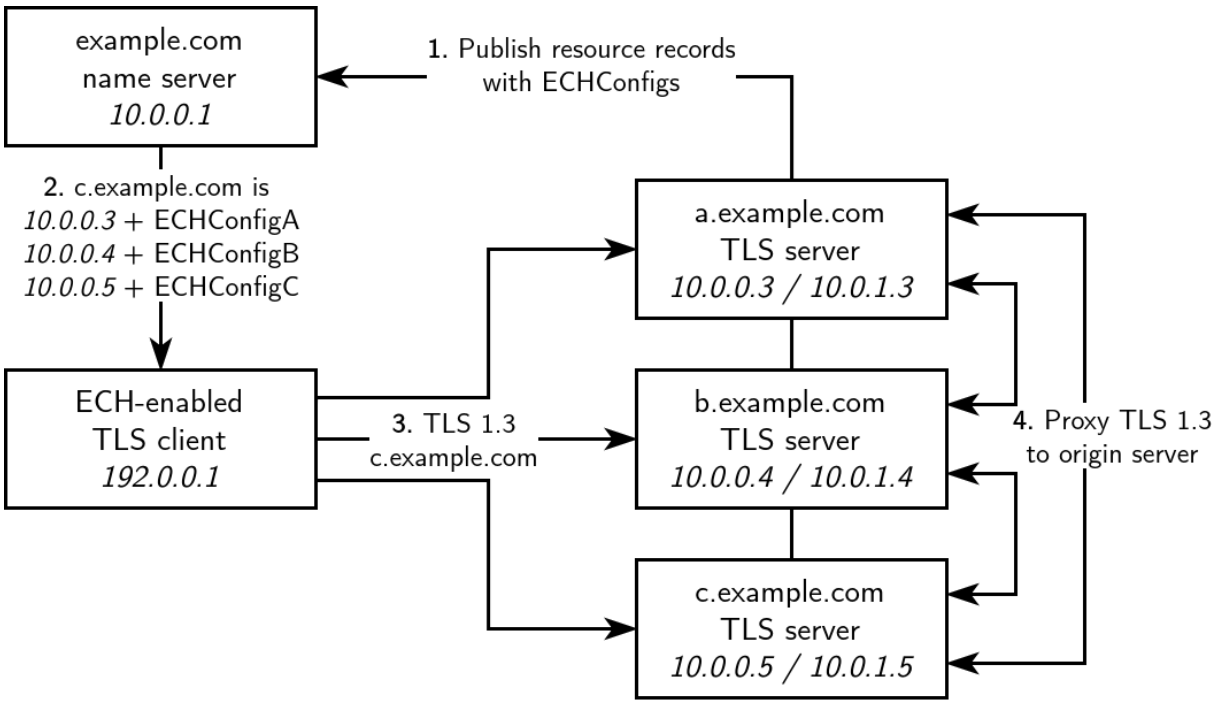


Figure 3.1.1: <>

<the design challenges can be split into two topics, mechanism for distribution and traffic obfuscation> <the distribution mechanism deals with a couple designs for the a dns publication schema so compatible clients follow the above process, and the networking required for tls servers to communicate with each other> <this considers load distribution, echconfig and stream proxying over a virtual private network>

3.2 Distribution Mechanism

<distribution is based on two parts> <first, clients need to be able to select from several options provided by a dns query> <second, co-oerating tcp servers need to be able to forward connections to each other based on the decrypted ClientHelloInner>

3.2.1 DNS Publication Schema

<one approach: shared echconfig with round robin A resource records> <load is evenly distributed across servers because client uses round robin on selected A rrs> <but requires shared secrets between all servers as there is no way to specify which ech key is associated with which host> <another approach: using alternative endpoints to associate ech keys with individual servers> <load is evenly distributed across servers due to matching priority> <unfortunately more fine-grained load distribution is not possible without srv resource records adoption> <we can replicate this using a dynamic dns service, where records are regularly substituted such that load is distributed across servers in a fair manner>

3.2.2 TLS Server Co-operation

<tls servers are physically separate from each other and must communicate over the same network as the client when forwarding the client connection> <to do this, a virtual private network is established between participating servers> <servers still listen for normal tls connections on 443 port of the public interface as well as ech-enabled connection> <ech split mode connections have their ClientHelloInner decrypted and private domain mapped to the actual origin> <clients connection is forwarded over a virtual private network to the origin server> <the origin server listening on the vpn interface completes the tls handshake with the client through the provider>

3.3 Traffic Obfuscation

<ech is susceptible>[10] <correlation attacks in low-latency systems like web browsing and video streaming compared to high-latency like email>[11] <rx/tx timing> <packet lengths> <traffic patterns> [12]

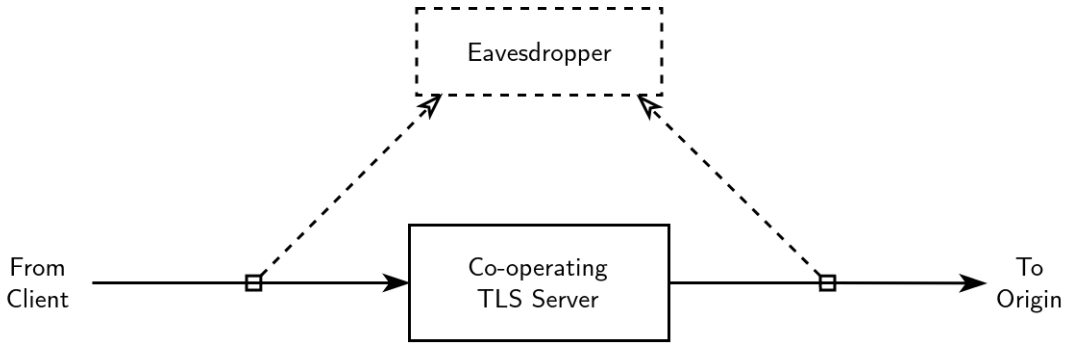


Figure 3.3.1: <TODO>

<traffic obfuscation techniques lead "to disrupt the patterns">

3.3.1 Normalisation

<shannon and perfect secrecy: removal of all identifying features> <injection of dummy traffic to fill bandwidth gaps> <impracticality in civilian environments due to required bandwidth>

3.3.2 Pacing and Mixing

<while not perfect, many practical techniques exist to mask traffic with minimal impacts> <padding packets themselves to prevent packet length correlation> [13] <to disrupt timing-based correlation attacks, delays and slotting (aka pacing) can be used> <to disrupt deep packet inspection, other packet-based correlation attacks (traffic patterns), traffic

mixing can be used> [14, 15] <a simple technique is packet duplication, where a similar packet is sent to every peer>

3.4 Summary

<it fulfills the desired design criteria> <we use dns https rrs to distributed load across multiple ech providers> <separate echconfigs can be used be each server as the ech key is associated with the server> <an alternative strategy exists that is more robust but requires shared echconfig> <a virtual private network between co-operating servers is used to allow peer communication> <ech split mode is vulnerable to correlation attacks> <but techniques exist to defend against attacks>

4 Implementation

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.1 Simulation

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.1.1 Virtualisation

qemu+debvm Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
ssh-keygen -N "" -t ed25519 -f ssh.key
debvm-create -h builder -o builder.img -r unstable -z 2GB -k ssh.key.pub -- \
    --include ca-certificates,build-essential,dh-autoreconf,git,e2fsprogs \
    --include libpsl-dev,libpcre3-dev,libz-dev,libnghttp2-dev

qemu-img create build.img 2G
debvm-run --image builder.img --sshport 2222 --graphical -- \
    -display none -drive file=build.img,format=raw,if=virtio,readonly=off &
debvm-waitssh 2222

ssh -o NoHostAuthenticationForLocalhost=yes -i ssh.key -p 2222 root@127.0.0.1 "
    mkfs.ext4 -L build /dev/vdb
    mount /dev/vdb /mnt

    git clone -b ECH-draft-13c https://github.com/sftcd/openssl.git /mnt/src/openssl
    cd /mnt/src/openssl
    ./config --prefix=/mnt/openssl --openssldir=/mnt/openssl
    make -j8
    make -j8 install

    cd / && umount /mnt
    shutdown now"
wait
```

Listing 4.1.1: TODO builder

4.1.2 Networking

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
sudo ip link add name br0 type bridge
sudo ip addr add 172.0.0.1/24 dev br0
sudo ip link set dev br0 up

debvm-run --image host.img -- \
    -device virtio-net-pci,netdev=net1,mac=00:00:00:00:00:01 \
    -netdev bridge,id=net1,br=br0
```

Listing 4.1.2: TODO br0

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected

font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
[Match]
MACAddress=00:00:00:00:00:01

[Network]
DNS=172.0.0.254
Address=172.0.0.5/24

[Route]
Gateway=0.0.0.0
Destination=0.0.0.0/0
Metric=9999
```

Listing 4.1.3: TODO /etc/systemd/network/00-br0.network contents

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl req -x509 \
    -newkey ec -pkeyopt ec_paramgen_curve:secp384r1 -days 3650 -nodes \
    -keyout /keys/root.key -out /keys/root.crt -subj '/CN=example.com'
```

Listing 4.1.4: TODO root ca

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.2 DNS Server

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no

information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl req \
    -newkey ec -pkeyopt ec_paramgen_curve:secp384r1 -nodes \
    -keyout /keys/dns.key -out /keys/dns.csr -subj '/CN=ns.example.com'

LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl x509 -req \
    -CA /keys/root.crt -CAkey /keys/root.key -days 3650 -CAcreateserial \
    -extfile <(printf 'subjectAltName=DNS:ns.example.com') \
    -in /keys/dns.csr -out /keys/dns.crt
```

Listing 4.2.1: TODO dns

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
tls tlspair {
    key-file "/keys/dns.key";
    cert-file "/keys/dns.crt";
};

options {
    directory "/var/cache/bind";
    recursion no;
    dnssec-validation auto;
    allow-transfer { none; };
    listen-on { any; };
    listen-on port 443 tls tlspair http default { any; };
};

zone "example.com" {
    type master;
    file "/var/lib/bind/db.example.com";
};
```

Listing 4.2.2: TODO /etc/bind/named.conf contents

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no

information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
$TTL 3600
@ IN SOA dns root.dns 2024040100 3600 600 86400 600
@ IN NS dns

dcu.ech.example.com. IN A 172.0.0.2
tcd.ech.example.com. IN A 172.0.0.5
ucd.ech.example.com. IN A 172.0.0.8

dcu.example.com. IN HTTPS 1 dcu.ech.example.com. ech=<DCU ECHConfig>
dcu.example.com. IN HTTPS 1 tcd.ech.example.com. ech=<TCD ECHConfig>
dcu.example.com. IN HTTPS 1 ucd.ech.example.com. ech=<UCD ECHConfig>

tcd.example.com. IN HTTPS 1 dcu.ech.example.com. ech=<DCU ECHConfig>
tcd.example.com. IN HTTPS 1 tcd.ech.example.com. ech=<TCD ECHConfig>
tcd.example.com. IN HTTPS 1 ucd.ech.example.com. ech=<UCD ECHConfig>

ucd.example.com. IN HTTPS 1 dcu.ech.example.com. ech=<DCU ECHConfig>
ucd.example.com. IN HTTPS 1 tcd.ech.example.com. ech=<TCD ECHConfig>
ucd.example.com. IN HTTPS 1 ucd.ech.example.com. ech=<UCD ECHConfig>
```

Listing 4.2.3

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.3 TLS Server

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.3.1 Peer Communication

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
wg genkey | tee /keys/tcd/wg.key | wg pubkey > /keys/tcd/wg.key.pub
```

Listing 4.3.1: TODO host wg

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
[NetDev]
Name=wg0
Kind=wireguard

[WireGuard]
ListenPort=51820
PrivateKey=<TCD WireGuard Private Key>

[WireGuardPeer]
PublicKey=<DCU WireGuard Public Key>
AllowedIPs=172.0.1.2/32
Endpoint=172.0.0.2:51820

[WireGuardPeer]
PublicKey=<UCD WireGuard Public Key>
AllowedIPs=172.0.1.8/32
Endpoint=172.0.0.8:51820
```

Listing 4.3.2: TODO host wg

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected

```
[Match]
Name=wg0

[Network]
Address=172.0.1.5/24
```

Listing 4.3.3: TODO host wg0

font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.3.2 Web Server

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl ech \
    -public_name tcd.example.com -pemout /keys/tcd/key.ech
```

Listing 4.3.4: TODO host+site tls

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
stream {
    ssl_preread on;
    ssl_echkeydir /keys/tcd;
    server {
        listen 172.0.0.5:443;
        proxy_pass $backend;
    }
    map $ssl_preread_server_name $backend {
        dcu.example.com 172.0.1.2:443;
        ucd.example.com 172.0.1.8:443;
    }
}

http {
    server {
        root /site/tcd;
        server_name tcd.example.com;
        listen 172.0.1.5:443 ssl;
        http2 on;
        ssl_certificate /keys/tcd/tcd.crt;
        ssl_certificate_key /keys/tcd/tcd.key;
        ssl_protocols TLSv1.3;
        location / {
            ssi on;
            index index.html index.htm;
        }
    }
}
```

Listing 4.3.5: TODO nginx

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.4 TLS Client

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
tc qdisc replace dev enp0s6 root netem slot 100ms 200ms

tcpdump -i wg0 -nnqt ip and src 172.0.1.5 | while read _ _ _ dst _ len; do
    [ "172.0.1.2" != "${dst%.*}" ] &&
        dd if=/dev/urandom bs=$len count=1 >/dev/udp/172.0.1.2/1234 &
    [ "172.0.1.8" != "${dst%.*}" ] &&
        dd if=/dev/urandom bs=$len count=1 >/dev/udp/172.0.1.8/1234 &
done
```

Listing 4.3.6

4.4.1 curl

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

```
LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/curl/bin/curl \
    --verbose --cacert /keys/root.crt --ech hard \
    --doh-url https://ns.example.com/dns-query https://tcd.example.com
```

Listing 4.4.1: TODO curl

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.4.2 Mozilla Firefox

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

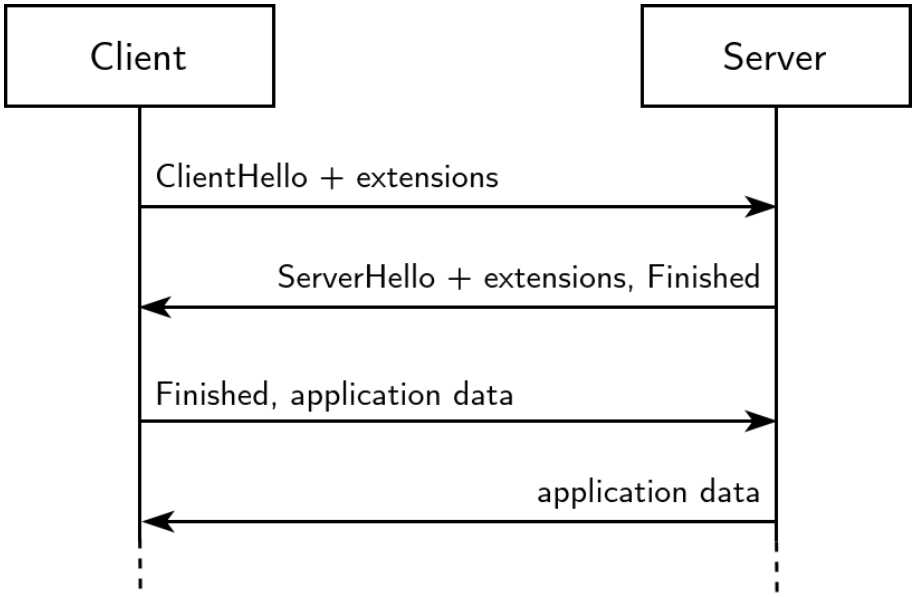


Figure 4.4.1: <TODO>

4.4.3 Google Chrome

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

4.5 Summary

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

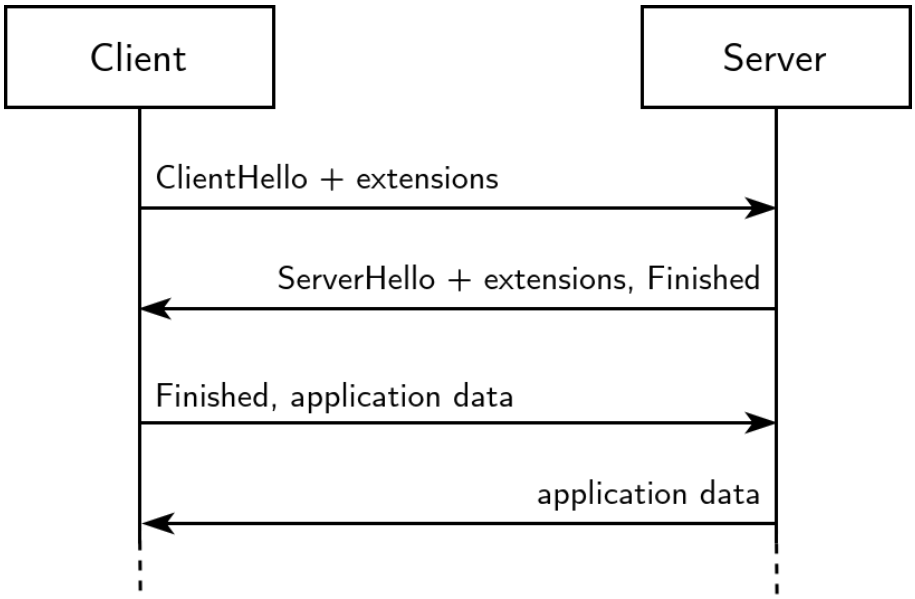


Figure 4.4.2: <TODO>

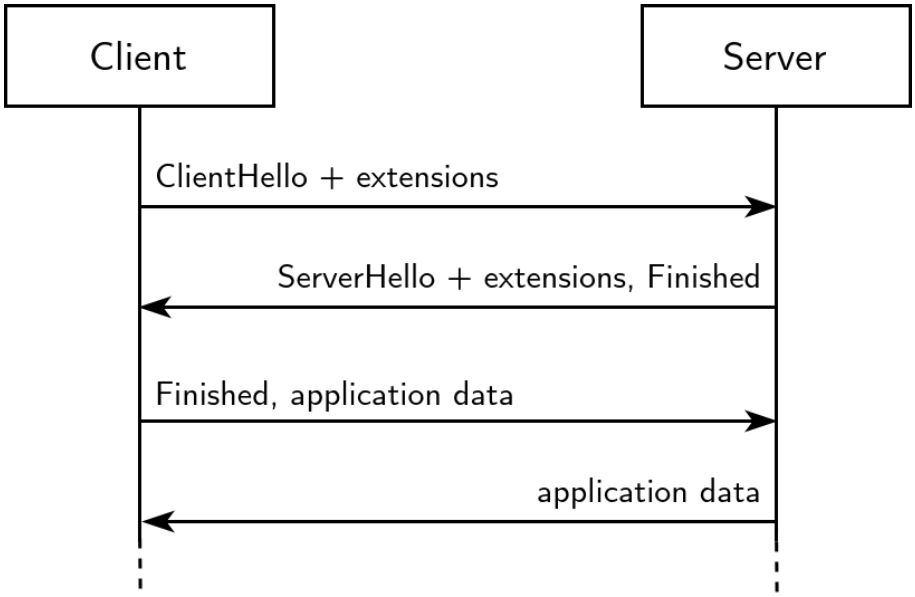


Figure 4.4.3: <TODO>

5 Results and Discussion

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.1 Data Collection

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.2 Evaluation

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

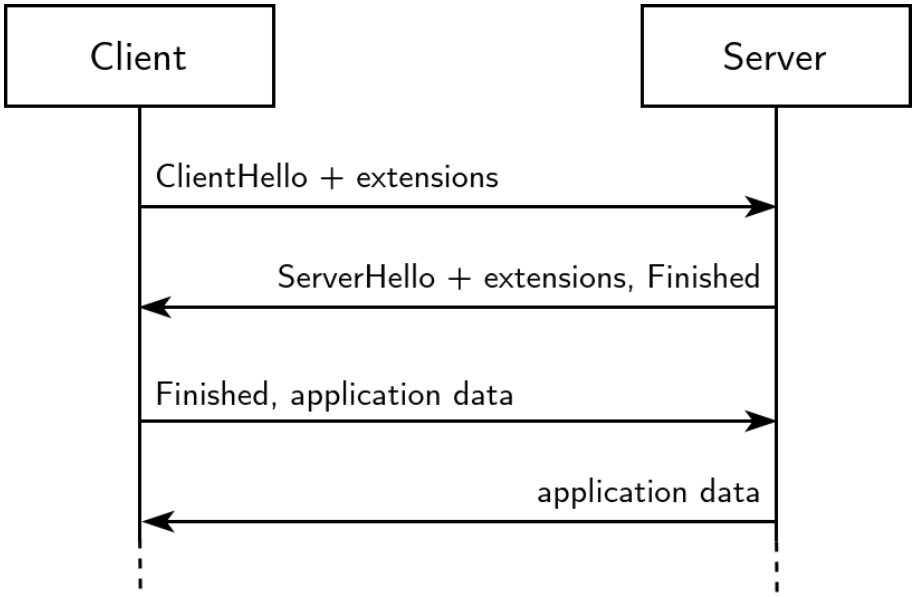


Figure 5.1.1: <>

5.2.1 Performance

compare against normal ech and no ech Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

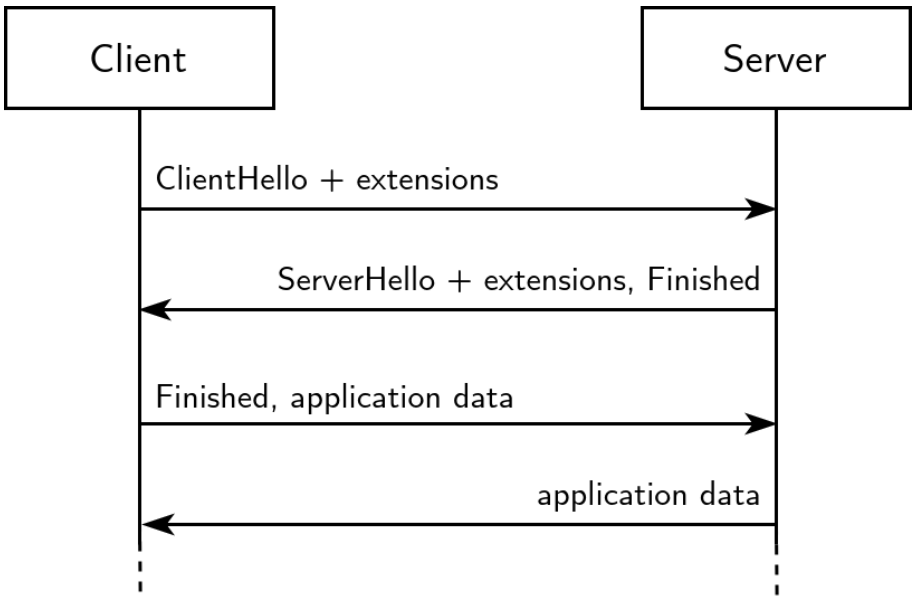


Figure 5.2.1: <>

5.2.2 Security

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

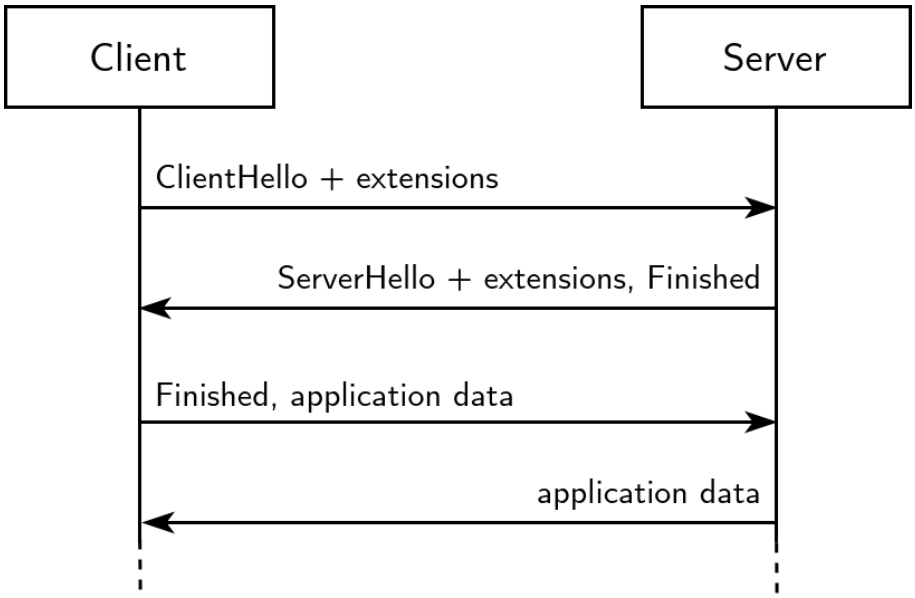


Figure 5.2.2: <>

5.3 Limitations

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.3.1 Load Balancing

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest

gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.3.2 Traffic Padding

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.4 Summary

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

6 Conclusion

<this research has been a first effort to create a distributed ech deployment model> <we have seen background> <we have seen design> <we have seen implementation> <we have seen results>

<in this final chapter, present key learnings> <then outline future work> <finish with a reflection of the project as a whole>

6.1 Learnings

<ECH Split Mode topology permits distributed deployment> <https rr enable static load distribution> <A dynamic DNS service allows for fair load distribution> <minimal performance impact when compared to normal> <security is ok and techniques exist to defend against correlation attacks>

6.2 Future Work

<shared dns publication strategy (with ech key rotation)> <dynamic traffic flow analysis for load balancing (instead of just distribution)> <more in-depth study on disrupting traffic correlation attacks> <test deployment on actual hardware with realistic traffic>

6.3 Reflection

<what i think went well> <reliable, reproducable and deterministic virtual environment test bed> <functioning distributed ech prototype> <if i was to do this again, what would i change> <dedicate more time to theoretical instead of practical (e.g. got firefox working but unsure if actually compatible with spec> <however, the work completed is still applicable to future effort>

Bibliography

[1] Zisis Tsiatsikas, Georgios Karopoulos, and Georgios Kambourakis. "Measuring the adoption of TLS encrypted client hello extension and its forebear in the wild". In: *European Symposium on Research in Computer Security*. Springer. 2022, pp. 177–190.

[2] Christopher Wood Achiel van der Mandele Alessandro Ghedini and Rushil Mehra. *Encrypted Client Hello - the last puzzle piece to privacy*. Sept. 2023. URL: https://blog.cloudflare.com/announcing-encrypted-client-hello (visited on 03/24/2024).

[3] Mark Nottingham. *Centralization, Decentralization, and Internet Standards*. RFC 9518. Dec. 2023. DOI: 10.17487/RFC9518. URL: https://www.rfc-editor.org/info/rfc9518.

[4] Chia-ling Chan et al. "Monitoring TLS adoption using backbone and edge traffic". In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2018, pp. 208–213.

[5] Let's Encrypt Stats. *Percentage of Web Pages Loaded by Firefox Using HTTPS*. URL: https://letsencrypt.org/stats/#percent-pageloads (visited on 04/01/2024).

[6] Christopher Allen and Tim Dierks. *The TLS Protocol Version 1.0*. RFC 2246. Jan. 1999. DOI: 10.17487/RFC2246. URL: https://www.rfc-editor.org/info/rfc2246.

[7] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. Aug. 2018. DOI: 10.17487/RFC8446. URL: https://www.rfc-editor.org/info/rfc8446.

[8] Kathleen Moriarty and Stephen Farrell. *Deprecating TLS 1.0 and TLS 1.1*. RFC 8996. Mar. 2021. DOI: 10.17487/RFC8996. URL: https://www.rfc-editor.org/info/rfc8996.

[9] Karthikeyan Bhargavan, Vincent Cheval, and Christopher Wood. "A symbolic analysis of privacy for tls 1.3 with encrypted client hello". In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2022, pp. 365–379.

[10] Martino Trevisan et al. "Attacking DoH and ECH: Does Server Name Encryption Protect Users' Privacy?" In: *ACM Transactions on Internet Technology* 23.1 (2023), pp. 1–22.

[11] Brian N Levine et al. "Timing attacks in low-latency mix systems". In: *Financial Cryptography: 8th International Conference, FC 2004, Key West, FL, USA, February 9-12, 2004. Revised Papers 8*. Springer. 2004, pp. 251–265.

[12] Samuel Padraic DeFabbia-Kane. "Analyzing the effectiveness of passive correlation attacks on the tor anonymity network". In: (2011).

[13] Shui Yu et al. "Predicted packet padding for anonymous web browsing against traffic analysis attacks". In: *IEEE Transactions on Information Forensics and Security* 7.4 (2012), pp. 1381–1393.

[14] Xinwen Fu et al. "Analytical and empirical analysis of countermeasures to traffic analysis attacks". In: *2003 International Conference on Parallel Processing, 2003. Proceedings.* IEEE. 2003, pp. 483–492.

[15] Xinwen Fu et al. "On effectiveness of link padding for statistical traffic analysis attacks". In: *23rd International Conference on Distributed Computing Systems, 2003. Proceedings.* IEEE. 2003, pp. 340–347.

A1 Code Listing

<what this code is in relation to project> <why it was created> <it does this that and the other thing> <requirements>

```
#!/bin/bash

hash sudo ip ssh ssh-keygen debvm-create debvm-run debvm-waitssh || exit 1
msg() { printf "\n\033[1;33m$@\033[0m\n"; }
pkill qemu-system-x86

# Parse provided config
dir="$1"
network_cfg="$(<"$2")" || exit 1
server_cfgs="$(<"$3")" || exit 1
read -d "" domain dns_host dns_mac dns_ip <<< "$network_cfg"
mkdir -p "$dir" || exit 1

# Generate SSH keypair
[ -f "$dir/ssh.key" ] && [ -f "$dir/ssh.key.pub" ] || {
  msg "Generating ssh keypair for VMs..."
  ssh-keygen -N "" -t ed25519 -f "$dir/ssh.key" || exit 1
}

# Setup host network bridge
sudo ip link show br0 1>/dev/null 2>&1 || {
  msg "Creating network bridge br0..."
  sudo ip link add name br0 type bridge || exit 1
  sudo ip addr add 172.0.0.1/24 dev br0 || exit 1
}
[ -z "$(ip link show br0 up)" ] && {
  msg "Bringing up network bridge br0..."
  sudo ip link set dev br0 up || exit 1
}

# Generate build with builder
[ -f "$dir/builder.img" ] || {
```

```
  msg "Generating builder.img..."
  debvm-create -h builder -o "$dir/builder.img" -r unstable -z 2GB -k
  ↪  "$dir/ssh.key.pub" -- \
    --include ca-certificates,build-essential,dh-autoreconf,git,e2fsprogs \
    --include libpsl-dev,libpcre3-dev,libz-dev,libnghttp2-dev || exit 1
}
[ -f "$dir/build.img" ] || {
  msg "Generating build.img:"

  cmds="
  # Format and mount build.img
  mkfs.ext4 -L build /dev/vdb || exit 1
  mount /dev/vdb /mnt || exit 1

  # Build OpenSSL patched with ECH support
  git clone -b ECH-draft-13c https://github.com/sftcd/openssl.git /mnt/src/openssl \
    && cd /mnt/src/openssl || exit 1
  ./config --prefix=/mnt/openssl --openssldir=/mnt/openssl || exit 1
  make -j8 || exit 1
  make -j8 install || exit 1

  # Build curl patched with ECH support
  git clone -b ECH-experimental https://github.com/sftcd/curl.git /mnt/src/curl \
    && cd /mnt/src/curl || exit 1
  autoreconf -fi || exit 1
  CPPFLAGS=-I/mnt/openssl/include LDFLAGS=-L/mnt/openssl/lib64 ./configure \\
    --prefix=/mnt/curl --with-openssl --enable-ech --enable-httpsrr || exit 1
  LD_LIBRARY_PATH=/mnt/openssl/lib64 make -j8 || exit 1
  make -j8 install || exit 1

  # Build NGINX patched with ECH support
  git clone -b ECH-experimental https://github.com/sftcd/nginx.git /mnt/src/nginx \
    && cd /mnt/src/nginx || exit 1
  ./auto/configure --prefix=/mnt/nginx \\
    --with-cc-opt=-I/mnt/openssl/include --with-ld-opt=-L/mnt/openssl/lib64 \\
    --with-stream --with-stream_ssl_module --with-stream_ssl_preread_module \\
    --with-http_ssl_module --with-http_v2_module || exit 1
  LD_LIBRARY_PATH=/mnt/openssl/lib64 make -j8 || exit 1
  make -j8 install || exit 1
  sed 's/\\/usr\\/sbin\\/nginx/\\/mnt\\/nginx\\/sbin\\/nginx -c \\/site\\/nginx.conf
  ↪  -p \\/site\\/nginx/' \\
    /mnt/src/nginx/debian/nginx-common.nginx.service > /mnt/nginx/nginx.service ||
    ↪  exit 1
  >>/mnt/nginx/nginx.service echo '
  [Service]
  Environment=LD_LIBRARY_PATH=/mnt/openssl/lib64' || exit 1
```

```
76
77    # Graceful shutdown
78    cd / && umount /mnt || exit 1
79    shutdown now"
80
81    echo "$cmds"
82    qemu-img create "$dir/build.img" 2G || exit 1
83    debvm-run --image "$dir/builder.img" --sshport 2222 --graphical -- \
84      -display none -drive file="$dir/build.img",format=raw,if=virtio,readonly=off &
85    debvm-waitssh 2222 || exit 1
86    ssh -o NoHostAuthenticationForLocalhost=yes -i "$dir/ssh.key" -p 2222
      ↪  root@127.0.0.1 "$cmds" || exit 1
87    wait
88  }
89
90  # Generate base VM image
91  [ -f "$dir/base.img" ] || {
92    msg "Generating base.img:"
93
94    cmds="
95    # Mount build.img
96    mount -o ro /dev/disk/by-label/build /mnt || exit 1
97
98    # Install some debugging tools
99    apt-get --yes install vim dnsutils iproute2 || exit 1
100
101   # Generate CA root and DNS key+certificate
102   mkdir -p /keys || exit 1
103   LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl req -x509 \\
104     -newkey ec -pkeyopt ec_paramgen_curve:secp384r1 -days 3650 -nodes \\
105     -keyout /keys/root.key -out /keys/root.crt -subj '/CN=root.$domain' || exit 1
106   LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl req \\
107     -newkey ec -pkeyopt ec_paramgen_curve:secp384r1 -nodes \\
108     -keyout /keys/$dns_host.key -out /keys/$dns_host.csr -subj
      ↪  '/CN=$dns_host.$domain' || exit 1
109   LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl x509 -req \\
110     -CA /keys/root.crt -CAkey /keys/root.key -days 3650 -CAcreateserial \\
111     -extfile <(printf 'subjectAltName=DNS:$dns_host.$domain,IP:$dns_ip') \\
112     -in /keys/$dns_host.csr -out /keys/$dns_host.crt || exit 1
113   chmod +r /keys/{root,$dns_host}.key || exit 1"
114
115   for server_cfg in $server_cfgs; do IFS=, read host _ ip _ sites <<< $server_cfg
116     cmds="$cmds
117     # Generate $host WireGuard and ECH keypair
118     mkdir -p /keys/$host || exit 1
119     wg genkey | tee /keys/$host/wg.key | wg pubkey > /keys/$host/wg.key.pub || exit 1
```

```
    LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl ech \\
      -public_name $host.$domain -pemout /keys/$host/key.ech"
    for site in ${sites//,/ }; do
      cmds="$cmds
      # Generate $site.$domain key+certificate
      LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl req \\
        -newkey ec -pkeyopt ec_paramgen_curve:secp384r1 -nodes \\
        -keyout /keys/$host/$site.key -out /keys/$host/$site.csr -subj
        ↪  '/CN=$site.$domain' || exit 1
      LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl x509 -req \\
        -CA /keys/root.crt -CAkey /keys/root.key -days 3650 -CAcreateserial \\
        -extfile <(printf 'subjectAltName=DNS:$site.$domain,IP:$ip') \\
        -in /keys/$host/$site.csr -out /keys/$host/$site.crt || exit 1
      chmod +r /keys/$host/$site.key || exit 1"
    done
  done

  cmds="$cmds
  # Graceful shutdown
  cd && umount /mnt || exit 1
  shutdown now"

  echo "$cmds"
  debvm-create -h base -o "$dir/base.img" -r unstable -z 1GB -k "$dir/ssh.key.pub"
  ↪  -- \
    --include ca-certificates,wireguard,libpsl5,libpcre3,libnghttp2-14 || exit 1
  debvm-run --image "$dir/base.img" --sshport 2222 --graphical -- \
    -display none -drive file="$dir/build.img",format=raw,if=virtio,readonly=on &
  debvm-waitssh 2222 || exit 1
  ssh -o NoHostAuthenticationForLocalhost=yes -i "$dir/ssh.key" -p 2222
  ↪  root@127.0.0.1 "$cmds" || exit 1
  wait
}

# Set DNS server configuration
cmds="
# Install dependencies
apt-get --yes install bind9 || exit 1

# Configure BIND9 for DoH usage
>/etc/bind/named.conf.options echo '
tls tlspair {
  key-file \"/keys/$dns_host.key\";
  cert-file \"/keys/$dns_host.crt\";
};
options {
```

```
  directory \"/var/cache/bind\";
  recursion no;
  dnssec-validation auto;
  allow-transfer { none; };
  listen-on { any; };
  listen-on port 443 tls tlspair http default { any; };
};' || exit 1

# Configure BIND9 to be dns.example.com
>/etc/bind/named.conf.local echo '
zone \"$domain\" {
  type master;
  file \"/var/lib/bind/db.$domain\";
};' || exit 1

# Configure BIND9 with RRs for dns.example.com
>/var/lib/bind/db.$domain echo '
\$TTL 60
@ IN SOA $dns_host root.$dns_host 2007010401 3600 600 86400 600
@ IN NS $dns_host
$dns_host IN A $dns_ip"
for server_cfg in $server_cfgs; do IFS=, read host _ ip _ sites <<< $server_cfg
  cmds="$cmds"$'\n'"$host.ech IN A $ip"
  for site in ${sites//,/ }; do
    for p_server_cfg in $server_cfgs; do IFS=, read p_host _ <<< $p_server_cfg
      [ "$host" != "$p_host" ] && {
        cmds="$cmds"$'\n'"$site IN HTTPS 1 $p_host.ech ech='\$(tail -2
        ↪  /keys/$p_host/key.ech | head -1)'"
      }
    done
  done
done
cmds="$cmds' || exit 1"

declare "${dns_host}_cmds=$cmds"

# Set TLS servers configuration
for server_cfg in $server_cfgs; do IFS=, read host _ ip wg sites <<< $server_cfg
  cmds="
  # Install dependencies
  apt-get --yes install wireguard tcpdump || exit 1

  # Configure WireGuard
  >/etc/systemd/network/00-wg0.netdev echo '
  [NetDev]
  Name=wg0
```

```
208   Kind=wireguard
209   [WireGuard]
210   ListenPort=51820
211   PrivateKey='\"\$(cat /keys/$host/wg.key)\"'"
212   for p_server_cfg in $server_cfgs; do IFS=, read p_host _ p_ip p_wg _ <<<
      ↪  $p_server_cfg
213     [ "$host" != "$p_host" ] && {
214       cmds="$cmds
215       [WireGuardPeer]
216       PublicKey='\"\$(cat /keys/$p_host/wg.key.pub)\"'
217       AllowedIPs=$p_wg/32
218       Endpoint=$p_ip:51820"
219     }
220   done
221   cmds="$cmds' || exit 1
222   >/etc/systemd/network/00-wg0.network echo '
223   [Match]
224   Name=wg0
225   [Network]
226   Address=$wg/24' || exit 1
227
228   # Configure NGINX
229   mkdir -p /site/nginx/logs || exit 1
230   >/site/nginx.conf echo '
231   pid /run/nginx.pid;
232   worker_processes 1;
233   events { worker_connections 1024; }
234
235   # ECH client-facing server as proxy for each WireGuard peer
236   stream {
237     log_format basic \"\$remote_addr [\$time_local] \$protocol \$status \$bytes_sent
      ↪  \$bytes_received \$session_time\";
238     access_log logs/access.log basic;
239     ssl_preread on;
240     ssl_echkeydir /keys/$host;
241     server { listen $ip:443; proxy_pass \$backend; }
242     map \$ssl_preread_server_name \$backend {"
243   for p_server_cfg in $server_cfgs; do IFS=, read _ _ _ p_wg p_sites <<<
      ↪  $p_server_cfg
244     for p_site in ${p_sites//,/ }; do
245       cmds="$cmds $p_site.$domain $p_wg:443;"
246     done
247   done
248   cmds="$cmds
249     }
250   }
```

```
251
252   # ECH backend server listening only through WireGuard
253   http {"
254   for site in ${sites//,/ }; do
255     cmds="$cmds
256     server {
257       root /site/$site;
258       server_name $site.$domain;
259       listen $wg:443 ssl;
260       http2 on;
261       ssl_certificate /keys/$host/$site.crt;
262       ssl_certificate_key /keys/$host/$site.key;
263       ssl_protocols TLSv1.3;
264       location / { ssi on; index index.html index.htm; }
265     }"
266   done
267   cmds="$cmds
268   }' || exit 1"
269
270   for site in ${sites//,/ }; do
271     cmds="$cmds
272     # Generate $site index.html
273     mkdir -p /site/$site || exit 1
274     >/site/$site/index.html echo '\
275     <!doctype html>
276     <html lang=en>
277       <head>
278         <meta charset=utf-8>
279         <title>$site.$domain</title>
280       </head>
281       <body>
282         <img src=\"/image.png\" width=\"300\" height=\"300\">
283         <p>
284           Welcome to <b>$site.$domain</b><br/>
285           Got here via <i><!--# echo var=\"remote_addr\" --></i>
286         </p>
287         <ul>
288           <li>SNI: <!--# echo var=\"ssl_server_name\" --></li>
289           <li>HTTP host: <!--# echo var=\"http_host\" --></li>
290           <li>ALPN protocol: <!--# echo var=\"ssl_alpn_protocol\" --></li>
291         </ul>
292         <form action=\"/pkglist\">
293           <input type=\"submit\" value=\"Download pkglist\" />
294         </form>"
295     for p_server_cfg in $server_cfgs; do IFS=, read p_host _ p_ip _ p_sites <<<
      ↪  $p_server_cfg
```

```
      cmds="$cmds<p>Sites on $p_host ($p_ip):"
      for p_site in ${p_sites//,/ }; do
        cmds="$cmds<br/><a href=\"https://$p_site.$domain\">$p_site.$domain</a>"
        [ "$site" = "$p_site" ] && cmds="$cmds *" || true
      done
      cmds="$cmds</p>"
    done
    cmds="$cmds
      </body>
    </html>' || exit 1
    ln -s /mnt/src/openssl/doc/images/openssl-square-nontransparent.png
    ↪  /site/$site/image.png || exit 1
    ln -s /var/lib/apt/lists/deb.debian.org_debian_dists_unstable_main_binary-amd64_
    ↪  Packages /site/$site/pkglist || exit 1"
  done

  cmds="$cmds
  # WireGuard traffic padding service
  >/site/padding.sh echo '#!/bin/bash
  tc qdisc replace dev enp0s6 root netem slot 100ms 200ms
  tcpdump -i wg0 -nnqt ip and src $wg | while read _ _ _ dst _ len; do"
  for p_server_cfg in $server_cfgs; do IFS=, read _ _ _ p_wg _ <<< $p_server_cfg
    cmds="$cmds
    [ \"$p_wg\" != \"\${dst%.*}\" ] && dd status=none if=/dev/urandom bs=\$len
    ↪  count=1 >/dev/udp/$p_wg/12345 &"
  done
  cmds="$cmds
  done' || exit 1
  >/site/padding.service echo '
  [Unit]
  After=network-online.target
  [Service]
  ExecStart=/site/padding.sh
  Restart=always
  [Install]
  WantedBy=multi-user.target' || exit 1
  chmod +x /site/padding.sh || exit 1

  # Install services
  cp /site/padding.service /mnt/nginx/nginx.service /etc/systemd/system || exit 1
  systemctl daemon-reload && systemctl enable padding nginx || exit 1"

  declare "${host}_cmds=$cmds"
done

# Generate all VM images in parallel
```

```
port=2222
for cfg in "$dns_host,$dns_mac,$dns_ip" $server_cfgs; do IFS=, read host mac ip _
↪  <<< $cfg
  port="$((port+1))"
  [ -f "$dir/$host.img" ] || {
    msg "Generating $host.img:"

    cmds_var="${host}_cmds"
    cmds="
    # Mount build.img
    >>/etc/fstab echo 'LABEL=build /mnt ext4 defaults 0 0' || exit 1
    mount -o ro /dev/disk/by-label/build /mnt || exit 1

    # Useful aliases
    >~/.profile echo '
    alias openssl=\"LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/openssl/bin/openssl\"
    alias curl=\"LD_LIBRARY_PATH=/mnt/openssl/lib64 /mnt/curl/bin/curl\"
    echo \"dig +https @dns.example.com tcd.example.com https\"
    echo \"curl --verbose --cacert /keys/root.crt --ech hard --doh-url
    ↪  https://dns.example.com/dns-query https://tcd.example.com\"
    ' || exit 1

    # Configure networking
    hostnamectl set-hostname $host || exit 1
    sed -i 's/base/$host/g' /etc/hosts || exit 1
    >/etc/systemd/network/00-br0.network echo '
    [Match]
    MACAddress=$mac
    [Network]
    DNS=$dns_ip
    Address=$ip/24
    [Route]
    Gateway=0.0.0.0
    Destination=0.0.0.0/0
    Metric=9999' || exit 1

    # Execute $host-specific commands
    ${!cmds_var}

    # Graceful shutdown
    cd && umount /mnt || exit 1
    shutdown now"

    echo "$cmds"
    cp "$dir/base.img" "$dir/$host.img" || exit 1
    debvm-run --image "$dir/$host.img" --sshport "$port" --graphical -- \
```

```
      -display none -drive file="$dir/build.img",format=raw,if=virtio,readonly=on &
    debvm-waitssh "$port" || exit 1
    ssh -o NoHostAuthenticationForLocalhost=yes -i "$dir/ssh.key" -p "$port"
    ↪ root@127.0.0.1 "$cmds" || exit 1
  } &
done
wait

port=2222
for cfg in "$dns_host,$dns_mac" $server_cfgs; do IFS=, read host mac _ <<< $cfg
  sleep 1
  port="$((port+1))"
  {
    msg "Booting up host $host..."
    debvm-run --image "$dir/$host.img" --sshport "$port" --graphical -- \
      -display none -drive file="$dir/build.img",format=raw,if=virtio,readonly=on \
      -device virtio-net-pci,netdev=net1,mac=$mac -netdev bridge,id=net1,br=br0 &
    debvm-waitssh "$port" || exit 1
    msg "Host $host is up and running"
    echo "ssh -o NoHostAuthenticationForLocalhost=yes -i '$dir/ssh.key' -p $port
    ↪ root@127.0.0.1"
    wait
    msg "Host $host has shutdown"
  } &
done
wait
killall debvm-run qemu-system-x86_64
```