Effects of PQC on Wireguard protocol

Qiguang Wang, Degree

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Future Network System)

Supervisor: Stephen Farrell

August 2023

Effects of PQC on Wireguard protocol

Qiguang Wang, Master of Science in Computer Science

University of Dublin, Trinity College, 2023

Supervisor: Stephen Farrell

This paper introduces post-quantum WireGuard, a post-quantum instantiation of the WireGuard protocol. The study is based on the KEMTLS protocol (ACM CCS 2020), which uses key-encapsulation mechanisms (KEMs) instead of post quantum signatures to authenticate communication parties. It brings both the TLS and the traditional post-quantum TLS into a more efficient protocol. Notably, this variant does not merely put focus on few aspects of security such as authentication and forward secrecy, as commonly pursued by earlier research at designing post-quantum protocols. Instead, it also offer bilateral post-quantum authentication with moderate cost of computation and communication. To accomplish this, we substitute the existing Elliptic Curve Diffie-Hellman key exchange with a new asymmetric crypto primitive, namely KEM. This thesis explores various combinations of different KEMs to alter existing WireGuard implementations. We propose three incremental modifications, each providing different level of security against quantum threats.

Acknowledgments

Thank you Mum & Dad.

Qiguang Wang

University of Dublin, Trinity College
August 2023

Contents

List of Tables

List of Figures

Chapter 1

Introduction

WireGuard is an innovative secure network protocol that functions as a virtual network interface within the Linux kernel in most cases. Its primary goal is to serve as a drop-in replacement for similar secure network protocol including IPsec and OpenVPN, providing enhanced security, better performance and improved friendliness. The secure tunnel is established by binding the peer's public key with its corresponding source address. The design of WireGuard is directly deriving from the Noise framework, bearing in mind the problem of pervasive surveillance and active tampering. Also, to get over the bandwidth and computational cost , WireGuard streamline the handshake process, utilizing a single round trip to achieve security of both perfect forward secrecy (PFS) and key-compromise impersonation (K-CI) resistance[]. But, all these security properties only holds in the classical computer model. With the arrival of the quantum computers, it is necessary to reimage of WireGuard Diffie-Hellman (DH) style handshake to use key encapsulation mechanisms (KEMs) for authentication, rather than pre-shared static key as a transitional method. **Migrating to post-quantum WireGuard** To achieve higher security levels against quantum adversaries without losing the same security level against the classical adversaries, the original WireGuard protocol needs to be revised. The main focus of this research is on amending the handshake part of the WireGuard. The primary goal of WireGuard is to establish a secure channel between communicating peers. Specifically, a secure channel should guarantee at least the following properties - Authentication: If the the other entity is authenticated, the initiator of authentication request *should only talk to the particularly identified one he think he is talking to* - Secrecy: After a successful handshake, only the communication parties are able to derive the session key or shared secret. Meanwhile, the data sent over the channel should be accessible by the communication parties. - Integrity: Both the entities can be aware of any unexpected modification to the data sent over the channel made by third parties. Since the integrity and secrecy is assured by the combination of public key encryption and secret key encryption, and the session key derived by public key cryptography is the most important component of the subsequent secret key encryption, we should put most effort into how we apply the public key encryption for authentication. Typically, there are two categories of authentications: - Implicit authentication: In most cases, key authentication is referred to as "implicit key authentication" because it does not involve explicit validation of the key itself, but rather relies on the security properties of the underlying cryptographic primitives and the secrecy of the key to ensure its authenticity. It is a property of key agreement protocol that only the other intended entity may be able to get the session key. This property can be either unilateral (authenticated for one entity) or mutual (authenticated for both entities). - To explain the difference between implicit authentication and the other one, the notion "key confirmation" will be introduced. It's a similar property to implicit authentication. It means one entity can be sure that some other entities do get hold of the session key. It does not require "only identified entities" to have the key, it is just rather a guarantee of "having the key". - Explicit authentication: It requires both implicit authentication and key confirmation properties. It means the entity being explicitly authenticated not only is the intended one who is able to acquire the session key but also actually have the key. In the WireGuard scenario, it does not explicitly authenticate all the communication data which restricts its security in regard to eCK-PFS model[]. Therefore , we proposes a more efficient and safer way to leverage the KEM to achieve mutual explicit authentication, in a similar manner to the kemtls and kemtlspdk and achieve both post-quantum authentication and eCK-PFS. The research also instantiates with a variety of post-quantum secure KEMs based on fundamentally different mathematical assumptions.

1.1 Motivation

Style of English An impersonal style keeps the technical factors and ideas to the forefront of the discussion and you in the background. Try to be objective and quantitative in your conclusions. For example, it is not enough to say vaguely "because the compiler was unreliable the code produced was not adequate". It would be much better to say "because the XYZ compiler produced code which ran 2-3 times slower than PQR (see Table x,y), a fast enough scheduler could not be written using this algorithm". The second version is more likely to make the reader think the writer knows what he/she is talking about, since it is a lot more authoritative. Also, you will not be able to write the second version without a modicum of thought and effort.

The following points are couple of *Do's & Dont's* that I have noted down as feedback to reports over the years. The focus of this list is to encourage writers to be specific in writing reports - some of this is motivated by Strunk and White's The Elements of Style (Strunk and White (1999)). Regarding reports that are submitted as part of a degree, examiners have to read and mark these reports - make it easy for these examiners to give good marks by following a number of simple points:

Acronyms: Acronyms should be introduced by the words they represent followed by the acronym in capitals enclosed in brackets e.g. "...TCP (Transmission Control Protocol)..." ⇒ "... Transmission Control Protocol (TCP)..."

Contractions: I would generally suggest to avoid contractions such as "I'd", "They've", etc in reports. In some cases, they are ambiguous e.g. "I'd" ⇒ "I would" or "I had" and can lead to misunderstandings.

Avoid "do": Be specific and use specific verbs to describe actions.

Adverbs: Adverbs and adjectives such as "easily", "generally", etc should be removed because they are unspecific e.g. the statement "can be easily implemented" depends very much on the developer.

Articles: "A" and "an" are indefinite articles; they should be used if the subject is unknown. "The" is a definite article; which should be used if a specific subject is referred to. For example, the subject referred to in "allocated by the coordinator" is not determined at the time of writing and so the sentences should be changed to "allocated by a coordinator".

Avoid brackets: Brackets should not be used to hide sub-sentences, examples or alternatives. The problem with this use of brackets is that it is not specific and keeps the reader guessing the exact meaning that is intended. For example "... system entities (users, networks and services) through ..." should be replaced by "... system entities such as users, networks, and services through ...".

Figures: Figures and graphs should have sufficient resolution; figures with low resolution appear blurred and require the reader to make assumptions.

Captions: Use captions to describe a figure or table to the reader. The reader should not be forced to search through text to find a description of a figure or table. If you do not provide an interpretation of a figure or table, the reader will make up their own interpretation and given Murphy's law, will arrive at the polar opposite of what was intended by the figure or table.

Backgrounds: Backgrounds of figures and snapshots of screens should be light. Developers often use terminals or development environments with dark backgrounds. Snapshots of these terminals or developments are difficult to read when placed into a report.

Titles: Titles of section should never be followed immediately by another title e.g. a title of a chapter should be followed by text describing the content and relevance of the sections of the chapter and could then be followed by the title of the first section of the chapter.

Punctuation: A statement is concluded with a period; a question with a question mark.

Spellcheckers: Use a spellchecker!

1.2 Figures

The arranging of figures in Latex can lead to spending a lot of time on minor issues e.g. positioning a figure in a specific location on a page, fixing minor issues with an exact size of a figure, etc. Figure 1.1 provides a simple example that demonstrates the use of one of two macros for handling figures, called *includefigure*; the other macro, *includescalefigure*, is demonstrated in chapter 5. Figures should always be readable without magnification when printed and the resolution of an image should be sufficient to provide a clear picture when printed.

Figure 1.1: A caption should describe the figure to the reader and explain to the reader the meaning of the figure. A Sub-clause of Murphy's Law: If the interpretation of a figure is left to a reader, the reader will misinterpret the figure, feel insulted or decide to ignore it. Do not leave it up to the reader!

1.3 Structure & Contents

At the end of the introduction, a layout of the structure and the contents of the following chapters should be provided for the reader. The overall goal of all descriptions of contents that follows these descriptions is to prepare the reader. The reader should not be surprised by any content that is being presented and should always know how content that is currently being read fits within an overall dissertation.

Chapter 2

State of the Art

At the beginning of each chapter, a description should introduce the reader to the content of the chapter. The description should explain to the reader the layout of the chapter, the contribution that the chapter makes to the overall dissertation and the contribution of the individual sections towards the overall chapter.

From the perspective of this document forming part of your degree, this chapter should demonstrate to the reader your knowledge of the area of your dissertation project. It should present your knowledge in a coherent and detailed form. The reader should understand that you have in-depth knowledge of the area of the dissertation without being overloaded with information.

2.1 Background

A section on the background of the dissertation should provide the reader with an introduction to existing technologies and concepts that form the basis of the work presented in the dissertation.

2.2 Closely-Related Work

Work in research areas tends to address a number of specific aspects. Ideally, the discussion of published research should focus on the aspects that have been addressed by various publications - and not a discussion of the individual publications.

For example, if the topic would be a discussion of work on programming languages, the subsections of the related work could be discussions of object orientation and its realisation in various languages or the use of lambda functions by these languages.

2.2.1 Aspect #1

2.2.2 Aspect #2

2.3 Summary

Summarize the chapter and present a comparison of the projects that you reviewed.

	Aspect #1	**Aspect #2**
Row 1	Item 1	Item 2
Row 2	Item 1	Item 2
Row 3	Item 1	Item 2
Row 4	Item 1	Item 2

Table 2.1: Caption that explains the table to the reader

Chapter 3

Design

At the beginning of each chapter, a description should introduce the reader to the content of the chapter. The description should explain to the reader the layout of the chapter, the contribution that the chapter makes to the overall dissertation and the contribution of the individual sections towards the overall chapter.

3.1 Problem Formulation

This section should provide the reader with an overall description of the problem that will be addressed in the dissertation. In contrast to a generic discussion of the dissertation topic in the Introduction chapter, this section should provide a detailed discussion of the problem that has been identified based on the existing work that has been discussed in the preceeding chapter.

In some dissertations, it may make sense to convert this section into a short chapter of its own which follows the discussion of the existing work and preceeds the discussion of the work of the dissertation.

3.1.1 Identified Challenges

This section should present a short description of the gaps in the existing work and the relationship of these gaps to the work described in this dissertation.

3.1.2 Proposed Work

This section should provide a thorough description of the problem and an overview of the work proposed to address the problem.

3.2 Overview of the Design

A description of the approach that addresses the problem identified above.

3.3 Summary

Every chapter aside from the first and last chapter should conclude with a summary.

Chapter 4

Implementation

Guess what? At the beginning of each chapter, a description should introduce the reader to the content of the chapter. The description should explain to the reader the layout of the chapter, the contribution that the chapter makes to the overall dissertation and the contribution of the individual sections towards the overall chapter.

4.1 Overview of the Solution

```
x = 1
if x == 1:
    # indented four spaces
    print("x is 1.")
```

Listing 4.1: Lengthy caption explaining the code to the reader

The code in listing 4.1 is a demonstration how to include a file with code into the template.

4.2 Component One

The code in listing 4.2 is a demonstration how to include code in the template.

```
x = 1
if x == 1:
    # indented four spaces
    print("x is 1.")
```

Listing 4.2: Second Lengthy caption

4.3 Summary

Every chapter aside from the first and last chapter should conclude with a summary.

Chapter 5

Evaluation

The Evaluation chapter should present a comparison of the work that forms the basis of the dissertation and existing work. At a higher level, it should demonstrate an awareness of the relationship of the dissertation work to the research area that it is based in.

5.1 Experiments

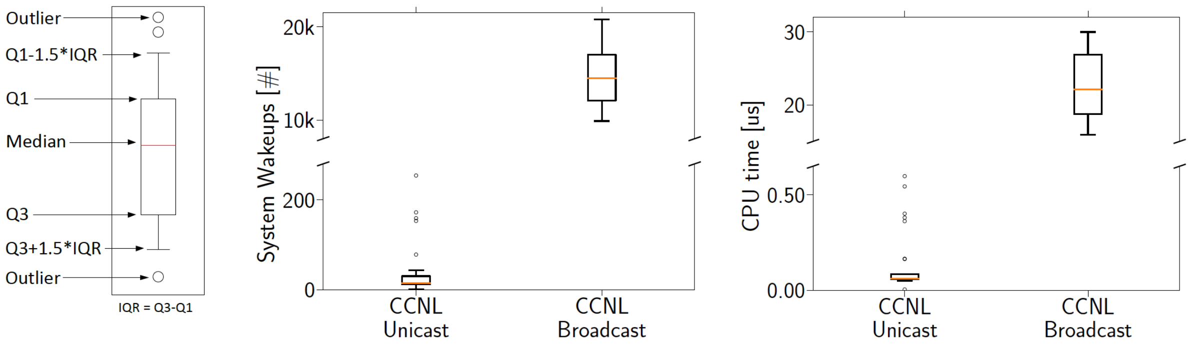
In the case where experiments have been carried out, the experimental setup and the values that were defined for the variables need to be presented in a table e.g. table 5.1.

Column 1	**Column 2**	**Column 3**
Row 1	Item 1	Item 2
Row 2	Item 1	Item 2
Row 3	Item 1	Item 2
Row 4	Item 1	Item 2

Table 5.1: Caption that explains the table to the reader

5.2 Results

Figures that present results such as figure 5.1 need to display descriptions of the axes, the units and scales of the measurements, statistical values, etc. Where measurements were taken from experiments, error bars or confidence intervals need to be provided to give the reader an indication of the spread of the measurements.



(b) System wakeups (c) CPU usage

Figure 5.1: Long caption that describes the figure to the reader

5.3 Summary

Every chapter aside from the first and last chapter should conclude with a summary that presents the outcome of the chapter in a short, accessible form.

Chapter 6

Conclusions & Future Work

This chapter should summarize the work presented in the dissertation and discuss the conclusions that can be drawn from the work and the results presented in chapter 5.

6.1 Future Work

The section may present a list of items that were beyond the scope of the dissertation.

Bibliography

Strunk, Jr., W. and White, E. B. (1999). *The Elements of Style.* Pearson Education, 4th edition.