SuperTLS: A Diverse and Redundant Secure Communication Channel for Privacy in Cloud

André Joaquim
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
Lisbon, Portugal

Miguel L. Pardal
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
Lisbon, Portugal

Miguel Correia
INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
Lisbon, Portugal

{andre.joaquim, miguel.pardal, miguel.p.correia}@tecnico.ulisboa.pt

ABSTRACT

We present superTLS, a diverse and redundant vulnerability-tolerant secure communication channel for privacy in cloud. There have always been concerns about the strength of some of encryption mechanisms used in SSL/TLS channels and some of them were regarded as insecure at some point in time. SuperTLS is our solution to mitigate the problem of secure communication channels being vulnerable to attacks due to unexpected vulnerabilities in its encryption mechanisms. It is based on diversity and redundancy of cryptographic mechanisms and certificates to provide a secure communication channel even when one or more mechanisms are regarded vulnerable. SuperTLS relies on a combination of k mechanisms/cipher suites, with k being the diversity factor and $k > 1$. Even when $k - 1$ mechanisms are regarded as insecure or considered vulnerable, SuperTLS relies on the remaining secure, diverse and redundant mechanism to maintain the channel secure. We evaluated the performance of our channel by comparing it to a normal TLS channel.

CCS Concepts

•**Networks** → **Application layer protocols;** *Network security;* •**Computer systems organization** → **Redundancy;** •**Security and privacy** → *Security protocols;* Symmetric cryptography and hash functions;

Keywords

Secure communication channels; Diversity; Redundancy; TLS; Vulnerability-Tolerance

1. INTRODUCTION

Secure communication channels are mechanisms that allow two entities to exchange messages or information securely in the Internet. A secure communication channel has three properties: *authenticity*, *confidentiality*, and *integrity*. Regarding authenticity, in an authentic channel, the messages can not be tampered. Regarding confidentiality, in a confidential channel, only the original receiver of a message is able to read that message. Regarding integrity, no one can impersonate another. The information regarding the original sender of a message can not be changed. Several secure communication channels exist nowadays, such as TLS, IPsec or SSH. Each of these examples is used for a different purpose, but with the same finality of securing the communication.

Transport Layer Security (TLS) is a secure communication channel widely used. Originally called Secure Sockets Layer (SSL), its first released version was SSL 2.0, released in 1995. SSL 3.0 was released in 1996, bringing improvements to its predecessor such as allowing forward secrecy and supporting SHA-1. Defined in 1999, TLS did not introduce major changes. Although, the changes introduced were enough to make TLS 1.0 incompatible with SSL 3.0. In order to grant compatibility, a TLS 1.0 connection can be downgraded to SSL 3.0, which brought security issues. TLS 1.1 and TLS 1.2 are upgrades which brought some improvements such as mitigating CBC (cipher block chaining) attacks and supporting more block cipher modes of operation to use with AES. TLS is divided in two sub-protocols, Handshake and Record, constituted by several mechanisms each. The Handshake protocol is used to establish or re-establish a communication between a server and client. The Record protocol is used to process the sent and received messages.

Internet Protocol Security (IPsec) is an Internet layer protocol that protects the communication at a lower level than SSL/TLS, which operates at the Application layer [2].

Secure Shell (SSH) is an Application layer protocol, such as SSL/TLS. SSH is a protocol used for secure remote login and other secure network services over an insecure network [3].

A secure communication channel becomes insecure when a vulnerability is discovered. Vulnerabilities may concern the protocol's specification, cryptographic mechanisms used by the protocol or specific implementations of the protocol. Many vulnerabilities have been discovered in SSL/TLS originating new versions of the protocol with renewed security aspects such as deprecating cryptographic mechanisms or enforcing security measures. Concrete implementations of SSL/TLS have been also considered vulnerable by hav-

WOODSTOCK '97 El Paso, Texas USA

DOI: 10.475/123_4

ing implementation details causing a breach of security and affecting devices worldwide.

- Maybe talk a little about the project's context – the cloud and communication between clouds

- Description

SuperTLS is a secure communication channel tolerant to vulnerabilities which does not rely on only one cryptographic mechanism. It is our belief that *diversity* and *redundancy* of cryptographic mechanisms and certificates can help mitigate existent vulnerabilities. In our project's context, diversity and redundancy consist in using two or more different mechanisms/cipher suites with the same objective. For example, MD5 and SHA-3 are both hash functions used to generate digests. In a real case, where MD5 has become insecure, our diverse and redundant secure communication channel relies upon SHA-3 to keep the communication secure. Using diversity and redundancy of cryptographic mechanisms, when a one of those mechanisms is successfully attacked, another mechanism is able to maintain the security and availability of the communication.

- The contributions of this paper are: x,y,z

Using diversity and redundancy, SuperTLS aims at increasing security over current secure communication channels. Guaranteeing a diversity factor $k > 1$, which implies having k different cipher suites with optimally k different mechanisms for each of the following:

- Key exchange;

- Authentication;

- Encryption;

- MAC: HMAC/AEAD (used for data integrity). **NOTE: In TLS, HMAC is used for CBC and stream cipher. AEAD is used for GCM and CCM (MtE)**

Although SSL/TLS supporting strong encryption mechanisms, such as AES and RSA, there are other factors than mathematical complexity that can contribute to vulnerabilities. Diversifying encryption mechanisms includes diversifying certificates and consequently keys (public, private, shared). Diversity of certificates is a direct consequence of diversifying encryption mechanisms due to the fact that each certificate is related to an authentication and key exchange mechanism.

One of the challenges of this project is to study if diversity and redundancy have a real impact on increasing security of a communication channel while having reasonable performance and time-related costs which required a precise measurement of our solution.

- The rest of the paper is organized as followed: z,y,x

The rest of the document is organized in the following way: Section 2 presents the related work. Section 3 presents the architecture of the proposed solution. Section 4 presents the evaluation. Section 5 presents some conclusions.

2. RELATED WORK

What are the current issues/vulnerabilities in the existing systems/mechanisms?

- Brief TLS

- Mechanisms vulnerabilities

- Brief combining mechanisms OR brief diversity

3. OVERVIEW

SuperTLS is a diverse and redundant secure communication channel. It aims at increasing security using diverse and redundant cryptographic mechanisms and certificates, and it is based on the TLS protocol. Although being an independent secure communication channel, SuperTLS is compatible with TLS 1.2 but does not support all of its cipher suites. (**NOTA: devia dizer isto?** –such as the ones containing PSK and SRP mechanisms)

This project aims to solve the main problem originated by having only one cipher suite negotiated between client and server: when one of the cipher suite's mechanisms becomes insecure, the secure communication channels using that cipher suite may become vulnerable. Although most cipher suites used by TLS 1.2 are regarded as secure, as stated in Section 2, there is not any assurance that a governmental agency or a company with high computational power and financial resources is not able to break one of those cryptographic mechanisms in the near future.

Unlike a TLS communication channel, a SuperTLS communication channel does not rely in only one cipher suite. SuperTLS negotiates more than one cipher suite between client and server and, consequently, more than one cryptographic mechanism will be used for each **phase/purpose** – key exchange, authentication, encryption and MAC.

Diversity and redundancy's first entry point in SuperTLS is in the SuperTLS' Handshake, where client and server negotiate k cipher suites to be used in the communication, where k, $k \geq 1$ is called the *diversity factor*. In an abnormal case where the diversity factor $k = 1$, it is considered that the communication channel has no diversity nor redundancy. Nevertheless, in this case, SuperTLS works as a regular TLS 1.2 channel with one cipher suite.

The strength of SuperTLS resides in the fact that, even when $(k-1)$ cipher suites become insecure, because one of its cryptographic mechanisms is insecure, our proposal remains invulnerable. The remaining secure, diverse and redundant cipher suite ensures that the communication channel is secured by remaining invulnerable.

Nevertheless, not all ciphers suites are compatible with one another. As referred in Section 2, cipher suites must be combined in a way that security is increased. For that to happen, we want to maximize diversity. In order to fulfil these constraints, some research was made to determine and quantify diversity among some cryptographic mechanisms [1].

The server chooses the best combination of k cipher suites according to the cipher suites server and client have available. The choice of the cipher suites might be conditioned by the certificates of both server and client. Diversity is measured using different metrics for hash functions or public-key cryptographic functions. Hash functions' metrics include origin, year, digest size, structure, rounds and weaknesses

(collisions, second preimage and preimage). After comparing several hash functions using the metrics stated above, the authors concluded that the best three combinations are the following:

- SHA-1 + SHA-3: This combination is not possible in SuperTLS. TLS 1.2 does not support SHA-3;
- SHA-1 + Whirlpool: This combination is not possible in SuperTLS. TLS 1.2 does not support Whirlpool;
- SHA-2 + SHA-3: This combination is not possible in SuperTLS. TLS 1.2 does not support SHA-3.

All the remaining suggested combinations cannot also be used because TLS 1.2 does not support SHA-3. These results have a direct impact in SuperTLS due to the fact that, being SuperTLS based on OpenSSL, and compatible with TLS 1.2, it also does not support SHA-3 nor Whirlpool. All of SuperTLS' cipher suites use either AEAD (MAC-then-Encrypt mode using a SHA-2 variant) or SHA-2 (SHA-256 or SHA-384).

Having a small range of available hash functions limits the maximum diversity factor achievable concerning hash functions. SHA-3 is relatively recent, having been selected ad the winner of the NIST hash function competition on 2012. In a near future, it is expected that a new TLS protocol version supports SHA-3 and makes possible the use of diverse hash functions.

Regarding public-key functions, the metrics used include origin, year, mathematical hard problems, perfect forward secrecy, semantic security and known attacks. After comparing several public-key encryption mechanisms, using the metrics stated above, the authors concluded that the best four combinations are the following:

- DSA + RSA: This combination is possible as TLS 1.2 suports both functions for *authentication*. However, TLS 1.2 specific cipher suites only support DSA with elliptic curves (ECDSA);
- DSA + Rabin-Williams: This combination is not possible. TLS 1.2 does not support Rabin-Williams;
- RSA + ECDH: This combination is possible as TLS 1.2 supports both functions for *key exchange*;
- RSA + ECDSA: This combination is possible as TLS 1.2 supports both functions for *authentication*.

Regarding authentication, although DSA + RSA is stated as the most diverse combination, TLS 1.2 preferred functions use ECDSA instead of DSA. Using elliptic curves results in a faster computation and lower power consumption **NOTA: encontrar referencia para isto. ver tese do ricardo. pg. 28, ultimo paragrafo.**. With that being said, the preferred combination for authentication is RSA + ECDSA.

Regarding key exchange, the most diverse combination is RSA + ECDH. Although, in order to grant perfect forward secrecy, the ECDH must be employed using ephemeral keys (ECDHE). Concluding, the preferred combination for key exchange is RSA + ECDHE.

The study did not presented any conclusions regarding symmetric-key encryption, such as AES. Therefore, considering the metrics employed for public-key encryption functions, and considering an additional metric – the mode of operation – we obtained combinations of diverse symmetric-key encryption:

- AES256-GCM + CAMELLIA128-CBC:
- AES256-CBC + CAMELLIA128-GCM:
- AES128-GCM + CAMELLIA256-CBC:
- AES128-CBC + CAMELLIA256-GCM:

As the cipher suites list is not very extensive, we are able to select a diverse group of k cipher suites without generating a considerable amount of overhead. As the cipher suites are presented in order of preference, the first cipher suite chosen is, ideally, the first of the list.

Diversity and redundancy will also be introduced in the following communication between client and server. It is our intent to use a subset of the k cipher suites defined in the Handshake Protocol to cipher the messages. While performance must be taken into account, we will proceed to estimate the reasonable k cipher suites and also the reasonable subset of k to cipher the messages with.

In terms of security, our solution must tolerate all the attacks given that at least one diverse redundant mechanism of the one attacked should not to be vulnerable that attack. On the contrary, our solution should never be vulnerable to attacks to which TLS 1.2 is not vulnerable. This fact would indicate a decrease of security.

3.1 Protocol Specification

- Images updated from the thesis' project

3.2 Implementation

- I started by implementing diversity in the cipher.
- The interface is the same as OpenSSL v1.0.2g, with the addition of SuperTLS specific functions
- Explain second cipher suite [X-Factor] audition and selection
- State my interface – new functions (most important)

Functions:

- *SSL_CIPHER *ssl3_choose_sec_cipher (SSL* s, STACK_OF(SSL_CIPHER) *clnt, STACK_OF(SSL_CIPHER) *srvr)*: Chooses a second cipher suite according to the second certificate and the first chosen cipher suite. This function tries to create maximum diversity.

4. EXPERIMENTAL EVALUATION

5. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LaTeX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

6. FUTURE WORK

- Usar duas libraries compativeis e usar funcoes duma e doutra

7. REFERENCES

[1] R. Carvalho. Authentication Security through Diversity and Redundancy for Cloud Computing. Master's thesis, Instituto Superior Técnico, Lisbon, Portugal, 2014.

[2] S. Kent and K. Seo. Security Architecture for the Internet Protocol (RFC 4301), 2005.

[3] T. Ylonen and C. Lonvick. The Secure Shell (SSH) Protocol Architecture (RFC 4251), 2006.