TLS Secure Man in the Middle

TLS Secure MITM

<series>draft-hallambaker-tls-mallet

<stream>ietf

<status>info

<ipr>trust200902

<layout tor=”true”>

<author>Phillip Hallam-Baker

<surname>Hallam-Baker

<initials>P. M.

<firstname>Phillip

<organization>Comodo Group Inc.

<email>[philliph@comodo.com](mailto:philliph@comodo.com)

# Abstract

By intent, TLS provides end-to-end encryption. While this is usually desired, there are circumstances in which it is desirable or a regulatory requirement for a client to establish a TLS connection through an intermediary that is capable of creating an archival record. This document describes a modification to the TLS protocol that allows an aware client to provide a recovery record to such an intermediary. The recovery record permits the intermediary to verify that the encrypted recovery records can be decrypted by a party that holds the corresponding private key but does not enable decryption by the intermediary.

# Introduction

The MALLET protocol is a modification of the TLS protocol to enable a MALLET aware client to negotiate a TLS tunnel through an intermediary such that the intermediary can determine that the communication can be decrypted by use of an offline key.

# Definitions

## Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [!RFC2119].

# Mechanism

TLS-MALLET, is a modification of the TLS interaction between the client and the intermediary that replaces the Perfect Forward Secrecy mechanism with a new mechanism that makes use of partial Diffie-Hellman key agreement.

In its most general form, a Diffie-Hellman agreement consists of function F(x) and a private key x such that F(x) forms a sub-group of order g of some space of possible values such that

* A function ‘.’ Exists and may be efficiently calculated such that F(a).F(b) = F(|a+b|g)
* It follows from the above that F(x) may be efficiently calculated
* Calculation of x given F(x) is infeasible.

For the sake of making the following argument clear to the reader, the implementation of the protocol is described using traditional Diffie Hellman (i.e. in a finite field). The preferred implementation however is to use Diffie Hellman on the elliptic Curve Ed25519 or Ed448.

In the normal TLS Perfect Forward Secrecy exchange, the client generates a Diffie Hellman key *x*, calculates |ex|g and sends the latter to the server for calculation of the Diffie Hellman agreement value.

In TLS-MALLET, the intermediary generates two public key pairs:

* Online keypair: Used to encrypt client escrow keys.
* Offline keypair: Used to decrypt archived encrypted communications.

The client is pre-configured with an interception record that contains the public components of the intermediary keys.

To establish a TLS-MALLET

# Security Considerations

# IANA Considerations

# Acknowledgements