Transport Layer Security	M. Williams	
Internet-Draft	J. Barrett	
Intended status: Standards Track	Nokia	
Expires: September 6, 2009	March 05, 2009	

TOC

Mobile DTLS
draft-barrett-mobile-dtls-00

Status of this Memo

Copyright Notice

Abstract

Mobile DTLS (Mobi-D) is an extension to DTLS that provides host mobility support. After obtaining a new IP address or port, a DTLS client mobile host can continue sending to its DTLS server correspondent host. The mobile host continues to use the existing set of security parameters, from the new address, without re-negotiation. The correspondent host accepts packets from the new IP address or port,

also without re-negotiation. After receiving any valid DTLS packet from the mobile host's new address or port, the correspondent host uses the new address or port to send to the mobile host.

Table of Contents

---

1. Introduction

TOC

Mobility service for hosts is available at the network layer from Mobile IP and from Proxy Mobile IP. For cases when neither of those mobility solutions are available, Mobi-D provides alternative mobility, at the transport layer.
Applications are now carrying audio and video over UDP for the benefits of low latency, help with NAT traversal, and where reliability has to be done at a higher layer. These applications are also being adapted for mobile devices, including multi-access devices such as those with both wired and wireless interfaces, or those with a wireless local area interface and a wireless wide area interface.

Mobi-D extends DTLS (Rescorla, E. and N. Modadugu, “Datagram Transport Layer Security,” April 2006.) [RFC4347] to provide host based mobility for the secure transport provided by DTLS. Mobi-D adds a connection identifier (ID) to the DTLS record layer. The connection ID is used for efficient indexing of the security parameters of a DTLS flow between the hosts. By associating packets to a flow based on security parameters instead of IP address and port, the connection can be maintained across change of the IP address, port, or change of interface.

A basic principle of Mobi-D is that no special messages are required to inform the correspondent host of a mobile host's move. Receipt of a fully valid Mobi-D record from the new address and port is sufficient. The mobile host can move without any re-negotation, and the other end can react immediately upon receipt of the first packet from the new address/port. An additional benefit is that applications on the mobile host can choose to ignore whether or not they are moving, and just make the assumption that the operating system has some address.

In whichever cases DTLS might be useful for securing UDP traffic, Mobi-D extends those cases to support mobile hosts and multi access hosts. The terms "client" and "server" are used as in DTLS. In this document a mobile host may have the role of DTLS client or server, but will typically take the client role. In other words, mobility support is not limited to client roles, either role may be mobile.

1.1. Requirements Language

TOC

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 (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.) [RFC2119].

---

2. Overview

TOC

Mobi-D extends DTLS in four ways:

1. A new DTLS Hello extension (Mobility)
2. A new TLS Hello extension (OP)
3. An additional field in the Record Layer (connection_id)
4. A new type of message (OP)

2.1. Mobility Extension

TOC

The Mobility extension allows a client and server to negotiate and establish mobility support for a DTLS connection.
The client allocates a unique connection_id associated with each DTLS handshake. For each full DTLS handshake, the client includes the connection_id in the Mobility extension, for use by the server when sending to this client. Likewise, the server allocates a unique connection_id and responds with a Mobility extension in ServerHello, including the connection_id for use by the client.
If the server includes the Mobility extension in its ServerHello, the client MUST include the server's connection_id in the very next record sent in the session, and all subsequent records. Likewise, the server MUST include the client's connection_id in the very next record sent in the session, and all subsequent records.
The Mobility extension includes a MobilityType value, which allows the server to declare that its address is fixed. If the server sends MobilityType fixed, then the client MUST NOT adjust its idea of the server's IP or port, even if a datagram is received from a different IP or port. This allows us to prevent the DoS attack described in the Security Considerations (Security Considerations) in one direction.
Clients which include the Mobility extension SHOULD include the OP extension as well, to allow OP messages to be used in the connection. Likewise, a server including Mobility SHOULD include OP as well, if the client did so.
Because TLS is inherently bound to a single host/port quartet, this extension MUST only be used with DTLS, and not with TLS. Any TLS server which receives this extension MUST ignore it.

2.2. OP Extension

TOC

The OP extension allows a client and server to negotiate and establish support for the OP message. The client MAY include the OP extension in its ClientHello. The extension contains a list of OpTypes supported by the client. The server MAY respond with the OP extension in its ServerHello, containing the list of OpTypes to be supported in the connection. The server's list is the intersection of the set of OpTypes supported by the server with the list provided by the client. If the intersection is nil, the server MUST NOT include the OP extension in its ServerHello.

2.3. New Message

TOC

Mobi-D specifies a new type of TLS message to be used for intra-protocol communication. The OP message consists of an OpType and opaque op_data and is encrypted and compressed as per the current connection state (like any other message). The OP message gets a new content type (managed by IANA, see Section 6 (IANA Considerations)).
The OP message is extensible to allow for future use. Mobi-D specifies one OpType, NOP (no-op), which is just an empty message. OP messages MUST be consumed by the receiving DTLS implementation, they are not delivered to the application.
If Mobility has been negotiated for the flow, the recipient of a NOP message MUST use the message's source address and port as the destination address and port for all new messages back to the sender. The NOP message is then discarded.

---

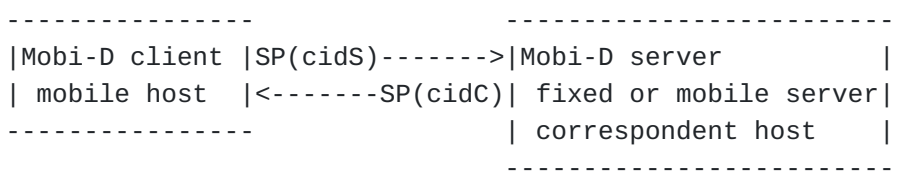
3. Usage Model

TOC

3.1. System Diagram

TOC

TODO: this diagram needs explanation.



The Mobi-D communication is between the DTLS client and server. The server allocates a connection ID (cidS) for use by the client. The client sends that ID in all records it sends to that server. The server receives a record with the connection ID it allocated, and uses that to look up the security parameters (SP) for the DTLS flow from the client, to authenticate and decrypt the record.
The client also allocates a connection ID (cidC) for use by the server. The server sends that ID in all records it sends to that client. The client receives a record with the connection ID it allocated, and uses that to look up the security parameters (SP) for the DTLS flow from the server, to authenticate and decrypt the record.
The terms server and client are used as in DTLS. However, the server and client may both be mobile devices. Note that the flows are not bound to or dependent on particular interfaces of the hosts.

3.2. Example Flow

TOC

In this example, the client changes IP address and the server immediately begins sending traffic to the new address. For the sake of simplicity, the "src" and "sport" examples given here are the values seen by the server. Quite possibly the client is behind a NAT and has a different address on its own interface.

```
            Client                                  Server
            ------                                  ------
            Data                  ------>
            (cid=100)
            (src=208.16.24.2)
            (sport=6723)

                                  <-----              Data
                                                   (cid=200)
                                                   (dst=208.16.24.2)
                                                   (dport=6723)

            Data                  ------>
            (cid=100)
            (src=208.69.36.132)
            (sport=8901)

                                  <-----              Data
                                                   (cid=200)
                                                   (dst=208.69.36.132)
                                                   (dport=8901)

                                  <-----              Data
                                                   (cid=200)
                                                   (dst=208.69.36.132)
                                                   (dport=8901)
```

3.3. Usage: Single Interface Host

TOC

For a single-interface mobile host (MH), when the host moves to a new network and is assigned a new IP address, the MH can continue the secure communication by using the new IP address with the old security

parameters. No signaling is needed, and the transport and application will continue as soon as the MH is ready to use the new IP address.

3.4. Usage: Multi-Interface Host

TOC

Mobi-D is not intended to allow two interfaces to carry a single DTLS flow simultaneously. If the MH has a Mobi-D connection and needs to continue that connection on a new interface, and has a different IP address on the new interface, the MH can continue the secure connection on the new interface by using the new IP address with the old security parameters.

If the MH wants to use two or more interfaces at once, it perform the DTLS handshake multiple times, allocating a unique connection_id for each flow on each interface. This way the MH can create separate Mobi-D connections for each of the flows from each interface. Mobility that causes only one of the interfaces to need a change of IP address is supported. For example, consider a mobile device with a small cell radio such as WLAN and a large cell radio such as LTE or WiMAX. Local mobility might cause the Mobi-D connection on the WLAN to move to a new network and configure a new IP address, while the larger cell is still in range and doesn't create a need to modify the IP address. In this case the host can continue to use both Mobi-D connections while updating the IP address on one.

In another multi-radio use case, the host may have two Mobi-D connections on two different interfaces, but may lose coverage on one of the radios. In this case the MH may use the interface that is still in range and the IP address already available on that interface. The MH begins sending the Mobi-D connection of the failing interface over the working interface. There is no need to get a new IP address, or re-handshake on the new interface to accomodate the second flow.

The receiving host of a Mobi-D connection will typically be a server of some kind, but may be another MH. The receiving host accepts Mobi-D packets from any IP address. The connection to which the inbound packet belongs is determined by the connection identifier and corresponding security parameters. In the case of a server receiving two inbound Mobi-D packets from the same client IP address, they will belong to the same connection or to different connections depending on the connection identifier and security parameters used by the MH when transmitting.

If an MH is behind a NAT, the Mobi-D packets may have NATed IP addresses or ports. These will still be delivered based on them having a connection identifier and matching security parameters.

During and after the MH changes IP address, in-bound datagrams may be lost, until the MH sends a packet to the receiving server or host to cause the receiving host to add or update the new IP address to the connection. This packet can be the next packet in the connection, or

can be a special Mobi-D NOP packet to optimize the remote host's detection of the address change.

---

4. Details

TOC

4.1. Extensions

TOC

The Mobility and OP extensions follow the TLS Extension format as defined in TLS (Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” August 2008.) [RFC5246]:

```
struct {
  ExtensionType extension_type;
  opaque extension_data<0..2^16-1>;
} Extension;

enum {
  (65535)
} ExtensionType;
```

The Mobility extension, OP extension, and new ExtensionType are defined as follows:

```
enum {
  mobility(), op(), (65535)
} ExtensionType;

struct {
    uint32 connection_id;
    MobilityType mobile_type;
} Mobility;

enum { mobile(1), fixed(2), (255) } MobilityType;

struct {
    uint8 length;
    OpType<1..256> op_types;
} OP;

enum { nop(1), (255) } OpType;
```

The extension_type for Mobi-D is maintained by IANA as described in the section on IANA considerations (IANA Considerations).

4.2. Record Layer

TOC

Mobi-D adds a connection identifier to the DTLS record layer which each host uses to look up the security parameters for the connection. The identifier (connection_id) is a single 32-bit integer, placed immediately after the content type.
TODO: should this handle collisions with non-Mobi-D DTLS records? For example, certain values of the high order byte of connection_id could be made illegal. The current placement in the record leaves this possibility open.
The modified DTLS record layer is as follows. connection_id is always the ID provided by the receiving party.

```
struct {
    ContentType type;
    uint32 connection_id;                  // New field
    ProtocolVersion version;
    uint16 epoch;
    uint48 sequence_number;
    uint16 length;
    opaque fragment[DTLSPlaintext.length];
} DTLSPlaintext;
```

Once established, the connection_id remains in use throughout the life of the DTLS session, including in a re-handshake. The client MUST include the server's connection_id in every record sent to the server, and likewise, the server MUST include the client's connection_id in every record sent to the client.
If a host receives a valid Mobi-D record with a valid connection_id from a new IP address or port, the host MUST use the message's source address and port as the destination address and port for all new messages back to the sender. A host MUST NOT change the destination address and port for the correspondent host until after successfully authenticating and verifying the sequence number of the DTLS record according to the connection_id contained within.
A new handshake (either for session resumption or a full handshake) MUST NOT use connection_ids in ClientHello and ServerHello because the IDs will point to existing security parameters that do not yet apply to the flow. The client and server MAY agree to re-use the same connection_ids via the Mobility extension negotiation in the handshake. Multiple connections between the same client and server MUST use different connection_ids and security parameters.

However, a re-handshake during an established session MUST include connection_ids in all records, as the connection_ids are necessary for each party to find the current security parameters.
The introduction of the connection identifier creates the possibility for an attacker with access to the packet flow to forward a packet after modifying the IP address or port, causing the return packet to be misaddressed. This attack is discussed further in the Security Considerations section (Security Considerations) below.

4.3. Message

TOC

Mobi-D adds a new message to TLS: the OP message. OP MUST be consumed by the receiving DTLS implementation and MUST NOT be delivered to the application. OP is an extensible message; other extensions may add OP message types. OP is implemented using a new content type, op, provided by IANA (see Section 6 (IANA Considerations)). Like other messages, op messages are encrypted and compressed, as specified by the current connection state. Mobi-D defines the NOP (no-op) OP message type.
Only OP message types negotiated in the OP Hello extension may be sent in a connection.
The extended ContentType is as follows. The OP content type is maintained by IANA as described in Section 6 (IANA Considerations).

```
enum {
    change_cipher_spec(20), alert(21), handshake(22),
    application_data(23), op(), (255)
} ContentType;
```

The OP message is defined as follows:

```
enum { nop(1), (255) } OpType;

struct {
    OpType op_type;
    opaque op_data<0..2^14-1>;
} OP;
```

---

5. Security Considerations

TOC

5.1. Cryptography

TOC

Mobi-D makes no changes to the cryptography of DTLS.

5.2. Connection ID

TOC

The connection_id has no cryptographic properties. An attacker able to capture packets can modify the connection_id. A Mobi-D host that receives a DTLS record with an incorrect (but valid) connection_id will be unable to authenticate the record and will reject it. A Mobi-D host that receives a record with an invalid connection_id (one it does not recognize) MUST discard the record.

5.3. Denial of Service

TOC

Mobi-D introduces new DoS attack. An attacker able to capture datagrams from the client will be able to send a valid datagram to the server from a forged IP address. This will cause the server to transmit packets to the forged address until the server receives a new packet from the real client. The same attack can occur in the other direction, but can be mitigated by the use of the fixed MobilityType in the server's Mobility extension.
Such an attack could deny service to the victim(s), but would not compromise the integrity of the encrypted data, nor allow the attacker to inject data of his own choosing, or to replay datagrams. Further, this attack is not substantially different from other denial of service attacks.

---

6. IANA Considerations

TOC

The Mobility extension number, OP extension number, and the OP content type will need to be registered with IANA.

---

7. Acknowledgements

TOC

The authors want to thank Pasi Eronen, Hannes Tschofenig, Basaravaj (Raj) Patil, Dan Wing, Teemu Savolainen, Lars Eggert, and Eric Rescorla

8. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC4347]	Rescorla, E. and N. Modadugu, “Datagram Transport Layer Security,” RFC 4347, April 2006 (TXT).
[RFC5246]	Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” RFC 5246, August 2008 (TXT).

Appendix A. Additional Stuff

TOC

This becomes an Appendix.

Authors' Addresses

TOC

	Michael Williams
	Nokia
Email:	michael.g.williams@nokia.com
	Jeremey Barrett
	Nokia
Email:	jeremey.barrett@nokia.com