Record Layer Design

This file provides some guidance on the thinking behind the design of the record layer code to aid future maintenance.

The record layer is divided into a number of components. At the time of writing there are four: SSL3_RECORD, SSL3_BUFFER, DLTS1_BITMAP and RECORD_LAYER. Each of these components is defined by:

1. A struct definition of the same name as the component
2. A set of source files that define the functions for that component
3. A set of accessor macros

All struct definitions are in record.h. The functions and macros are either defined in record.h or record_local.h dependent on whether they are intended to be private to the record layer, or whether they form part of the API to the rest of libssl.

The source files map to components as follows:

```
dtls1_bitmap.c                   -> DTLS1_BITMAP component
ssl3_buffer.c                    -> SSL3_BUFFER component
ssl3_record.c                    -> SSL3_RECORD component
rec_layer_s3.c, rec_layer_d1.c -> RECORD_LAYER component
```

The RECORD_LAYER component is a facade pattern, i.e. it provides a simplified interface to the record layer for the rest of libssl. The other 3 components are entirely private to the record layer and therefore should never be accessed directly by libssl.

Any component can directly access its own members - they are private to that component, e.g. ssl3_buffer.c can access members of the SSL3_BUFFER struct without using a macro. No component can directly access the members of another component, e.g. ssl3_buffer cannot reach inside the RECORD_LAYER component to directly access its members. Instead components use accessor macros, so if code in ssl3_buffer.c wants to access the members of the RECORD_LAYER it uses the RECORD_LAYER_* macros.

Conceptually it looks like this:

```
                      libssl
                         |
-------------------------|-----record.h------------------------------------
                         |
                  _______V______________
                 |                      |
                 |    RECORD_LAYER      |
                 |                      |
                 |    rec_layer_s3.c    |
                 |          ^           |
                 | _________|__________ |
                 ||                    ||
                 || DTLS1_RECORD_LAYER ||
                 ||                    ||
                 || rec_layer_d1.c     ||
                 ||____________________||
                 |______________________|
      record_local.h     ^   ^   ^
```

```
       __________________|    |    |__________________
      |                       |                       |
 _____V_________        ______V________        _______V________
|               |     |               |     |                |
| SSL3_BUFFER   |     | SSL3_RECORD   |     | DTLS1_BITMAP   |
|               |--->|               |     |                |
| ssl3_buffer.c |     | ssl3_record.c |     | dtls1_bitmap.c |
|_______________|     |_______________|     |________________|
```

The two RECORD_LAYER source files build on each other, i.e. the main one is rec_layer_s3.c which provides the core SSL/TLS layer. The second one is rec_layer_d1.c which builds off of the SSL/TLS code to provide DTLS specific capabilities. It uses some DTLS specific RECORD_LAYER component members which should only be accessed from rec_layer_d1.c. These are held in the DTLS1_RECORD_LAYER struct.