OsmoSGSN - Bug #1582

GEA Encryption is missing

02/23/2016 03:43 PM - laforge

Status:	Resolved	**Start date:**	02/23/2016
Priority:	Urgent	**Due date:**	
Assignee:	laforge	**% Done:**	100%
Category:			
Target version:			
Spec Reference:			

Description

3GPP specifies GPRS encryption at LLC level. We have implemented almost everything required to do this, but have never put all the bits together and tested them. We should support GEA1, GEA2 and GEA3.

Related issues:

Related to OsmoSGSN - Feature #1741: GEA encryption unit tests	**Closed**	**06/01/2016**
Blocked by OsmoSGSN - Bug #1758: segfault in sgsn	**Rejected**	**06/28/2016**
Blocked by OsmoSGSN - Bug #1794: support random IV for GEA (via XID)	**Stalled**	**08/09/2016**

History

#2 - 03/28/2016 09:53 AM - laforge

- Assignee set to spaar

#3 - 04/28/2016 07:27 PM - laforge

- Status changed from New to In Progress

#4 - 05/31/2016 01:10 PM - msuraev

Note: implementation of GEA3 and 4 was once posted to mailing list https://www.freecalypso.org/archive/lists.osmocom.org/baseband-devel/2013-April/003970.html but have not been forward ported.

#5 - 06/01/2016 02:04 PM - laforge

see http://lists.osmocom.org/pipermail/osmocom-net-gprs/2016-May/000614.html for an update, also here for convenience:

```
GEA issues (May 2016):
**********************

libosmo-crypt-gea12/src/osmocom.c
---------------------------------

GEA1_stream() and GEA2_stream() operate with bits and not bytes. This
is how they have to be used:

static int gea1_run(uint8_t *out, uint16_t len, uint64_t kc, uint32_t iv,
             enum gprs_cipher_direction direction)
{
    uint8_t dir;
    uint8_t bits_input[32];
    uint8_t bits_kc[64];
    int i;

    // get 32 input bits

    for(i = 0; i < 32; i++)
    {
        bits_input[i] = (BYTE)((iv >> i) & 1);
    }

    // get 64 key bits
```

```
    for(i = 0; i < 64; i++)
    {
    #if 0 // reverse Kc byte order ("reverse" compared to Calypso test code)
        bits_kc[i] = (BYTE)((kc >> i) & 1);
    #else // "normal" Kc byte order (whatever "normal" is)
        bits_kc[i] = (BYTE)((kc >> (((7 - (i / 8)) * 8) + (i % 8))) & 1);
    #endif
    }

    if (direction == GPRS_CIPH_MS2SGSN)
        dir = DIRECTION_UPLINK;
    else
        dir = DIRECTION_DOWNLINK;

    GEA1_stream(bits_kc, bits_input, dir, len, out);

    return 0;
}

static int gea2_run(uint8_t *out, uint16_t len, uint64_t kc, uint32_t iv,
             enum gprs_cipher_direction direction)
{
    uint8_t dir;
    uint8_t bits_input[32];
    uint8_t bits_kc[64];
    int i;

    // get 32 input bits

    for(i = 0; i < 32; i++)
    {
        bits_input[i] = (BYTE)((iv >> i) & 1);
    }

    // get 64 key bits

    for(i = 0; i < 64; i++)
    {
    #if 0 // reverse Kc byte order ("reverse" compared to Calypso test code)
        bits_kc[i] = (BYTE)((kc >> i) & 1);
    #else // "normal" Kc byte order (whatever "normal" is)
        bits_kc[i] = (BYTE)((kc >> (((7 - (i / 8)) * 8) + (i % 8))) & 1);
    #endif
    }

    if (direction == GPRS_CIPH_MS2SGSN)
        dir = DIRECTION_UPLINK;
    else
        dir = DIRECTION_DOWNLINK;

    GEA2_stream(bits_kc, bits_input, dir, len, out);

    return 0;
}

libosmo-crypt-a53/src/osmocom.c
-------------------------------

The length of the key is in bits and not in bytes:

static int gea3_run(uint8_t *out, uint16_t len, uint64_t kc, uint32_t iv,
             enum gprs_cipher_direction direction)
{
    uint8_t dir;

    if (direction == GPRS_CIPH_MS2SGSN)
        dir = 0;
    else
        dir = 1;

    GEA3((uint8_t *)&kc, sizeof(kc) * 8, iv, dir, out, len);

    return 0;
}
```

openbsc/openbsc/src/gprs/gprs_llc.c

Frame encryption in gprs_llc_tx_ui() doesn't work:

```
    /* encrypt information field + FCS, if needed! */
    if (lle->llme->algo != GPRS_ALGO_GEA0) {
        uint32_t iov_ui = 0; /* FIXME: randomly select for TLLI */
        uint16_t crypt_len = (fcs + 3) - (llch + 3);
        uint8_t cipher_out[GSM0464_CIPH_MAX_BLOCK];
        uint32_t iv;
        int rc, i;
        uint64_t kc = *(uint64_t *)&lle->llme->kc;

        /* Compute the 'Input' Paraemeter */
        iv = gprs_cipher_gen_input_ui(iov_ui, sapi, nu, oc);

        /* Compute the keystream that we need to XOR with the data */
        rc = gprs_cipher_run(cipher_out, crypt_len, lle->llme->algo,
                     kc, iv, GPRS_CIPH_SGSN2MS);
        if (rc < 0) {
            LOGP(DLLC, LOGL_ERROR, "Error crypting UI frame: %d\n", rc);
            msgb_free(msg);
            return rc;
        }

        /* Mark frame as encrypted */
        #if 1 // Dieter: recalculate checksum after properly setting the flag
        llch[2] |= 0x02;
        fcs_calc = gprs_llc_fcs(llch, fcs - llch);
        fcs[0] = fcs_calc & 0xff;
        fcs[1] = (fcs_calc >> 8) & 0xff;
        fcs[2] = (fcs_calc >> 16) & 0xff;
        #endif

        /* XOR the cipher output with the information field + FCS */
        for (i = 0; i < crypt_len; i++)
            *(llch + 3 + i) ^= cipher_out[i];

        /* Mark frame as encrypted */
        #if 0 // Dieter: This won't work
        ctrl[1] |= 0x02;
        #endif
    }
```

Frame decryption in gprs_llc_rcvmsg() doesn't work:

```
    /* decrypt information field + FCS, if needed! */
    if (llhp.is_encrypted) {
        uint32_t iov_ui = 0; /* FIXME: randomly select for TLLI */
        uint16_t crypt_len = llhp.data_len + 3;
        uint8_t cipher_out[GSM0464_CIPH_MAX_BLOCK];
        uint32_t iv;
        uint64_t kc = *(uint64_t *)&lle->llme->kc;
        int rc, i;

        if (lle->llme->algo == GPRS_ALGO_GEA0) {
            LOGP(DLLC, LOGL_NOTICE, "encrypted frame for LLC that "
                "has no KC/Algo! Dropping.\n");
            return 0;
        }

        iv = gprs_cipher_gen_input_ui(iov_ui, lle->sapi, llhp.seq_tx,
                        lle->oc_ui_recv);
        rc = gprs_cipher_run(cipher_out, crypt_len, lle->llme->algo,
                     kc, iv, GPRS_CIPH_MS2SGSN);
        if (rc < 0) {
            LOGP(DLLC, LOGL_ERROR, "Error decrypting frame: %d\n",
                 rc);
            return rc;
        }

        /* XOR the cipher output with the information field + FCS */
        for (i = 0; i < crypt_len; i++)
            *(llhp.data + i) ^= cipher_out[i];
```

```
        #if 1 // Dieter: FCS is encrypted, gprs_llc_hdr_parse() should be called again
        llhp.fcs = *(llhp.data + crypt_len - 3);
        llhp.fcs |= *(llhp.data + crypt_len - 2) << 8;
        llhp.fcs |= *(llhp.data + crypt_len - 1) << 16;
        #endif
    } else {
        if (lle->llme->algo != GPRS_ALGO_GEA0) {
            LOGP(DLLC, LOGL_NOTICE, "unencrypted frame for LLC "
                "that is supposed to be encrypted. Dropping.\n");
            return 0;
        }
    }
```

```
Open issues:
############

openbsc/openbsc/src/gprs/gprs_llc.c
-----------------------------------

When creating an LLME "on the fly" in lle_for_rx_by_tlli_sapi() kc and algo
of the LLME have to be set to the ones used for the subscriber.

In general:
-----------

Management of GEA algorithm and KC for a subscriber. The algorithm has to
be selected according to the capabilities of the MS. Some affected functions
are gprs_llgmm_assign() and gsm48_tx_gmm_auth_ciph_req().

Handling of IOV (initialisation vector), currently most of the time IOV is
set to 0, but gprs_llgmm_reset() and gprs_llgmm_reset_oldmsg() use a random
IOV and would probably cause problems when called.

Using uint64_t as the type for Kc in the GEA libraries is not a good choice,
uint8_t[8] would be better and cause less troubles (e.g. endian issues and
incompatibility with the test vectors, where the order of Kc has to be
reversed). Attention: this change might require to adjust the byte order of
Kc in the GEA libraries.
```

#6 - 06/01/2016 02:29 PM - laforge

- Assignee changed from spaar to laforge

- % Done changed from 0 to 30

changes to libosmo-crypt-* have just been pushed. Changes to OsmoSGSN still pending.

#7 - 06/01/2016 02:34 PM - laforge

- Related to Feature #1741: GEA encryption unit tests added

#8 - 06/14/2016 09:31 AM - laforge

- Assignee changed from laforge to msuraev

#9 - 06/14/2016 09:35 AM - laforge

- Priority changed from Normal to High

#10 - 06/14/2016 09:38 AM - laforge

- Priority changed from High to Urgent

#11 - 06/28/2016 10:02 AM - msuraev

- Blocked by Bug #1758: segfault in sgsn added

#12 - 07/01/2016 04:30 PM - msuraev

Testing with osmo-auc - Kc is generated properly and propagated to LLC layer, phone capabilities are checked but activating pdp context fails. Most likely reason: encryption is started at the wrong time (too early?). Unlike GSM in GPRS there seems to be no explicit command to mark the moment when encryption should be applied.

#13 - 07/04/2016 04:28 PM - msuraev

GEA3 encryption working with gerrit #455 but only tested with IOV=0 instead of random due to missing XID IOV negotiation code.

#14 - 07/05/2016 10:15 AM - msuraev

- % Done changed from 30 to 40

Preliminary implementation sent for review as gerrit #460

#15 - 07/18/2016 08:03 AM - msuraev

- % Done changed from 40 to 70

Everything except for XID is merged into master.

#16 - 07/18/2016 08:03 AM - msuraev

- Blocked by Feature #1580: IP header compression added

#17 - 07/19/2016 10:22 AM - msuraev

- Status changed from In Progress to Stalled

#18 - 08/09/2016 02:23 PM - msuraev

- Status changed from Stalled to Resolved

- Assignee changed from msuraev to laforge

- % Done changed from 70 to 100

Random IV generation/negotiation has been split into separate ticket. GEA with current hardcoded IV=0 is working.

#19 - 08/09/2016 02:24 PM - msuraev

- Blocked by Bug #1794: support random IV for GEA (via XID) added

#20 - 08/30/2016 05:05 PM - laforge

- Blocked by deleted (Feature #1580: IP header compression)