Feasibility study of two GKA protocols on LoRa network

Department of Computer, Control, and Management Engineering Antonio Ruberti

Sapienza
Università di Roma

Stefano Milani

GKA protocols in WSN

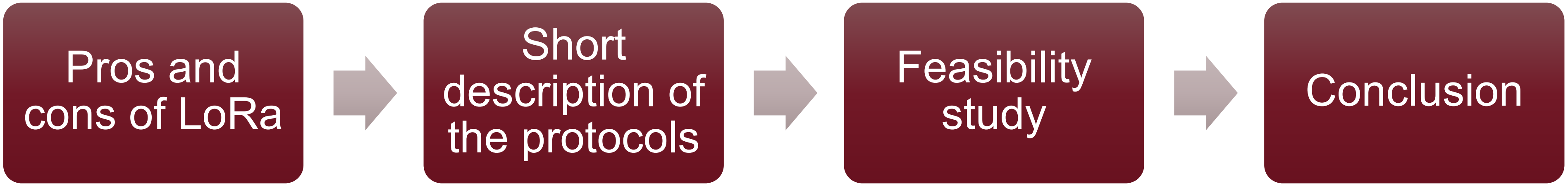
Lossy network

- Reduce the number of packets
- Avoid network congestion due to retransmission

Low bandwidth

- Reduce OTA time
- Avoid retransmission due to conflicts

What we are going to see



Pros and cons of LoRa

Short description of the protocols

Feasibility study

Conclusion

LoRa: Pros

Operates in unlicensed band

Around 10 years battery lifetime

2-5 KM in urban area
Up to 15 KM in suburban area

LoRa: Cons

- Operates in unlicensed band
- Low bit rate
- Reduced packet size

LoRa: some numbers*

- The Spreading factor ($SF = \log_2(R_c/R_s)$) can go from 7 to 12
- The higher the SF, the slowest the bit rate (the shorter the payload size)
- The higher the SF, the longest the communication range
- The bit rate can go from $0{,}3\ kb/s$ to $27\ kb/s$
- The max payload size vary from 51 bytes to 222 bytes

*Data taken from [1]

First GKA protocol* (1/3)

- Step 1

The initiator I generates a random value $r_i \in Z_p^*$

Computes $R_i = r_i G$

Broadcasts $\{Q_i, R_i, U\} + Sign$ (U : list of identities of devices)

- Step 2

The node U_j generates a random value $r_j \in Z_p^*$ and $R_j = r_j G$

Compute $R_{ij} = r_j R_j R_i$

Generate a secret $S_j = d_j Q_i + R_{ij}$ and $Auth_j = h(S_j \mid R_{ij} \mid U_j)$

Send to I: $\{R_j, Q_j, Auth_j, U_j\}$

*Protocol is described in detail in [2] (protocol 1)

First GKA protocol (2/3)

- Step 3

I receive the response from U_j and computes S_j^* as follows

$r_{ij}^* = r_i\, r_j \, mod\, p$ $\quad R_{ij}^* = r_{ij}^* G$ $\quad S_j^* = d_i Q_j + R_{ij}^*$

Checks if $Auth_j$ is valid

- Step 4

$S_j = (x_j, y_j)$ is encoded in (u_j, v_j) as: $u_j = h\left(x_j\right),\ v_j = h(y_j)$

$\forall\, j \in (1, \ldots, n-1)\ \overline{u_j} = \{\oplus_{i \neq j}\, u_i\} \oplus v_j$

$P = (\bar{u}_j | \ldots | \overline{u_j})$ and the secret group key is: $k = h(\oplus_i\, \bar{u}_i)$

Generates $Auth = h\left(k \middle| R_{ij} \middle| P\right)$ and sends $\{Auth, P\} + Sign$

First GKA protocol (3/3)

- Step 5

U_j receives the message and uses S_j to compute (u_j, v_j)

It can derive the key $k = h(\bar{u}_j \oplus u_j \oplus v_j)$

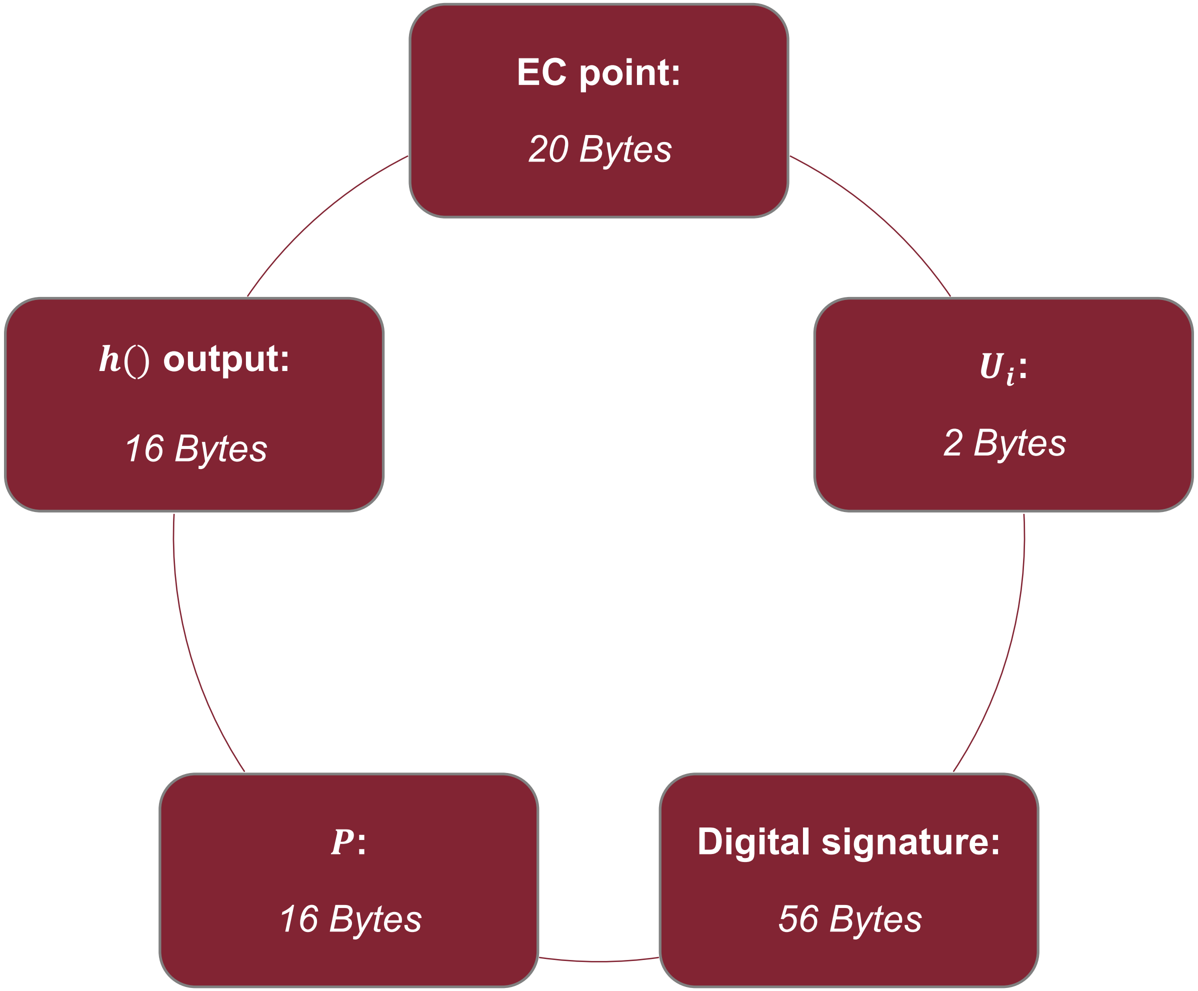
It verifies $Auth$ to check the correctness of the key

- Step 6

Each sensor send an ACK: $\{k, Q_i\}$ to the initiator to terminate the handshake.

This ensure that every nodes has correctly delivered the key

Some sizes *(secpr160r1 curve)*



EC point:

20 Bytes

U_i:

2 Bytes

Digital signature:

56 Bytes

P:

16 Bytes

$h()$ output:

16 Bytes

Messages length

Step 1 msg: $\{Q_i, r_i, U\} + sign$ => $(2n + 96)$ bytes

Step 2 msg: $\{R_j, Q_i, Auth_j, U_j\}$ => 58 bytes

Step 4 msg: $\{Auth, P\} + Sign$ => 88 bytes

Step 5 msg: $h(k, Q_i)$ => 16 bytes

Feasibility on LoRa

Worst case: $SF = 12$, payload max size: 51 bytes

Step 2 and step 4 message needs two LoRa packets to be transmitted.

Step 1 message at least 3 packets (if $n = 4$), further fragmentation needed with greater group.

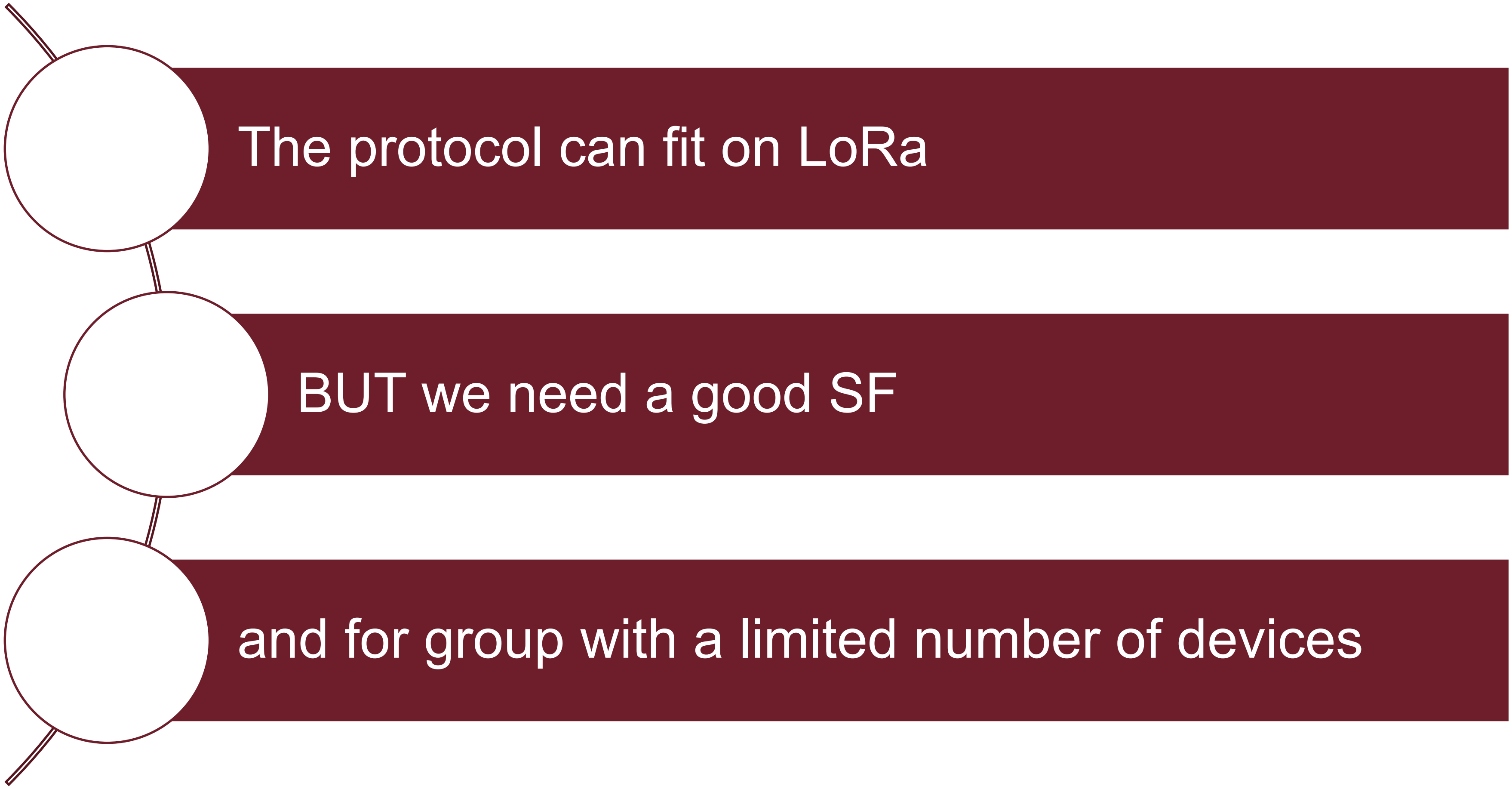
Best case: $SF = 7$, payload max size: 222 bytes

Fragmentation needed on step 1 message if group have more than 63 nodes.

Avg case: payload max size: 136 bytes

Fragmentation needed on step 1 message if group have more than 20 nodes.

Opinion



The protocol can fit on LoRa

BUT we need a good SF

and for group with a limited number of devices

Second GKA protocol* (1/2)

Given a group composed by n nodes

Every nodes generates a secret k_i.
M_1 selects a point P
Sends to M_2 $Q_1 = k_1 P$.

M_2 sends to M_3 $Q_2 = k_2 Q_1 = k_2 k_1 P$.
This process is repeated until the node M_n is reached. M_n generates $Q_n = k_1 \ldots k_n P$ the shared secret.

For each node M_i, k_i is the private key and Q_i is the public key.

*Protocol is described in detail in [3]

Second GKA protocol (2/2)

M_n encrypts Q_n with M_{n-1} public key Q_{n-1} and sends it to M_{n-1}

M_{n-1} decrypt the message and acquire Q_n.

Then it can repeat the procedure and sends the secret key to M_{n-2}, and so on so forth until every nodes receive the secret key.

Message length *(secpr160r1 curve)*

All the messages contain only an EC point:

- Public key of the previous node in the first step
- Encrypted secret group key in the second step

Plus an eventual flag to determine the message type (See the complete description of the protocol)

So the message length for all the messages is:

21 bytes

Feasibility on LoRa

In the worst case we have $SF = 12$ with a maximum payload size of 51 bytes.

So all the messages fit in one LoRa packet also in the worst case.

This allows us to use a “bigger” curve like the secp256k1.

In this case the EC point has size 32 bytes.

The messages are 33 bytes long, but they fit on LoRa also in the worst case.

Opinion

- The protocol fit on LoRa
- ALSO with $SF = 12$
- The message size does not depend on the number of device in the group

Conclusion

First protocol

- Msg size depends on the number of devices
- Fits LoRa under determinate conditions

Second protocol

- Msg size does NOT depend on the number of devices
- Fits LoRa also in the worst case

References

[1] Adelantado, Ferran, et al. **"Understanding the limits of LoRaWAN."** *IEEE Communications magazine* 55.9 (2017): 34-40.

[2] Porambage, Pawani, et al. **"Group key establishment for enabling secure multicast communication in wireless sensor networks deployed for IoT applications."** *IEEE Access* 3 (2015): 1503-1511.

[3] Chatzigiannakis, Ioannis, et al. **"Design, analysis and performance evaluation of group key establishment in wireless sensor networks."** *Electronic Notes in Theoretical Computer Science* 171.1 (2007): 17-31.