Communication Systems & Networks Group

Channel Reciprocity for KEy Transport (CRiCKET)

- A secure key agreement protocol for IoT networks -

Chrysanthi Paschou, Francesco Raimondo, George Oikonomou, James Pope

A. Motivation - The key distribution problem

IoT networks cannot support traditional cryptographic methods for key distribution. Physical Layer Key Generation (PLKG) is a promising solution due to being less computational complex. Yet, typical PLKG protocols are not practical for severely resource-constrained devices.

B. Backround - What is Physical Layer Key Generation?

Stage 1 Channel Probing

- Capturing the unique and **common randomness;**
- Alice and Bob exhange pilot signals and measure radio frequency characteristics such as the fluctuations on the Received Signal Strength (RSS).

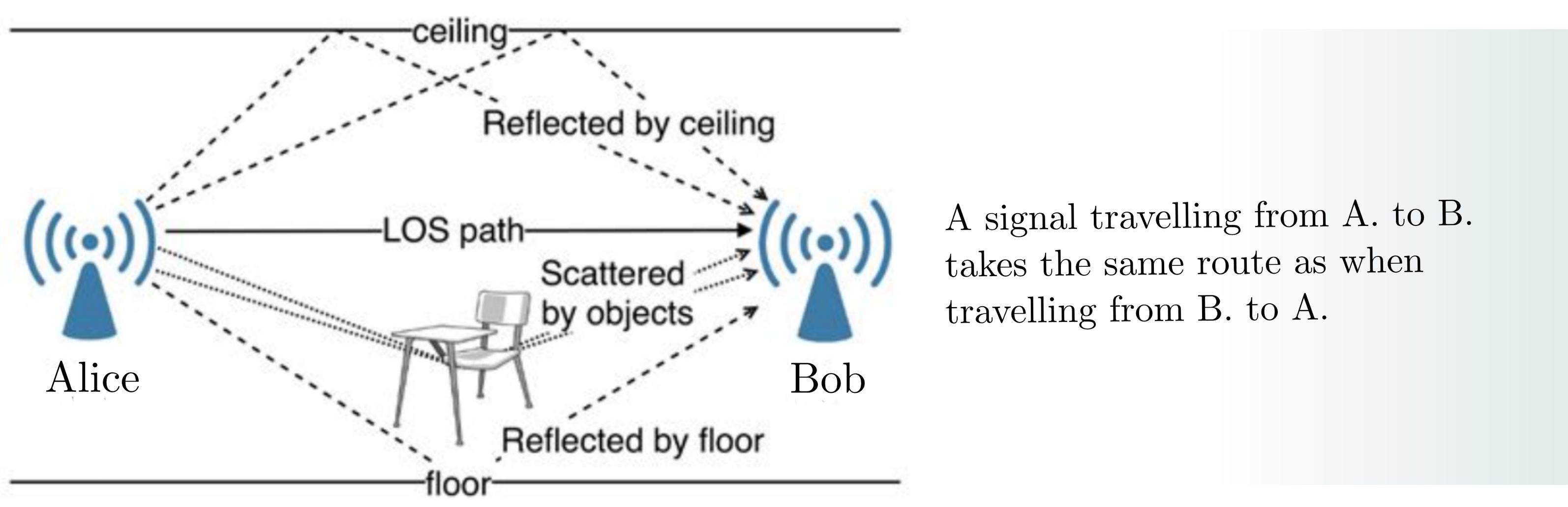
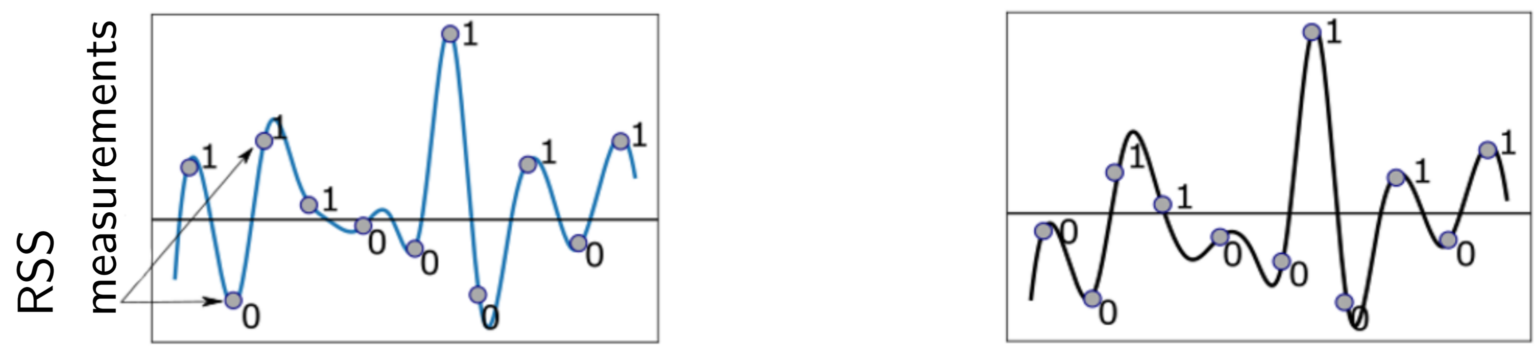


FIGURE I
Alice and Bob wish to agree on a key so that they can use it for upper-layer encryption. They send pilot signal to each other. They both observe similar small-scale fluctuations on the received pilots.

Stage 2 Channel Quantisation

Alice and Bob agree on a quantisation scheme;
They output two highly correlated binary sequences.



Alice: 10110010101 Bob: 00110011101

Stage 3 Key Reconciliation -> Corrects/discards mismatches

Common approaches:
Error-Detection-Code (EDC) based (borrowed from Communication Theory)
Error-Correction-Code (ECC) based (borrowed from Quantum Cryptography)

C. Current limitations and our solution

1

The Key Reconciliation phase can be "heavy";
When there are many bit-mismatches, common approaches are NOT practical in IoT networks.

Our solution: Channel Reciprocity for KEy Transport (CRicKET)

- CRicKET is an encoding that exploits channel reciprocity for "hiding" a key;
- CRicKET is applied after the quantisation stage;
- There is no need to reconcile the correlated sequences;
- Alice decides on a key and encodes it on her sequence;
- Bob compares the received codewords with his sequence and retrieves the key.

2 Key attributes

- CRicKET has **low reconciliation cost** (Table I);
- The probability of two matching keys can be mathematically predetermined (Fig.II);
- Analytical study that allows design optimisation and **information theoretic guarantees.**

Approach	Comm.Overhead	Complexity	Leakage
EDC-based	High	Low	Low
ECC-based	Low	High	High
CRicKET	Low	Low	Low

TABLE I
Reconciliation cost

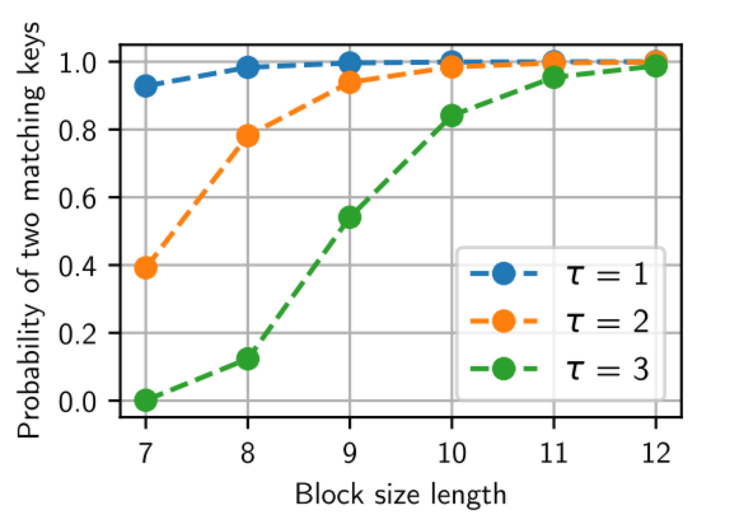


FIGURE II Probability of two matching 128-bit keys when the binary sequences from the quantisation stage disagree by 20%. Variables "τ" and "Block size length" are encoding/decoding parameters.

3 Implementation

- Practicabillity has been successfully tested using nodes implemented on nRF52840 board;
  Work in progress: Implement CRicKET seemlessly on ongoing traffic:
- Piggyback pilot signals onto outgoing data frames;
- Measure and quantify RSSI measurements in each received data frame.

4

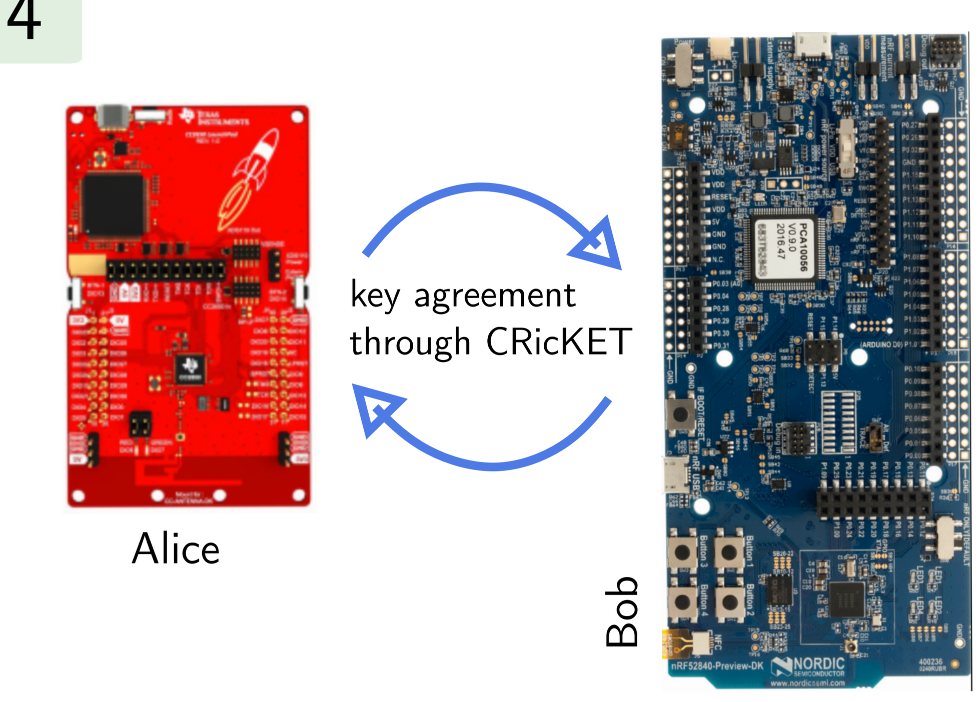


FIGURE III
CRicKET is implemented in single-chip devices

Alice: TIC cc2650
- 48MHz 32-bit processor
- 128 KB flash
- 20kB RAM

Bob: Nordic nRF52850
- 64 MHz 32-processor
- 1MB flash
- 256kB RAM

Interested in the analytical work?

Scan for email