

Channel Reciprocity for KKey Transport (CRiCKET)

- A secure key agreement protocol for IoT networks -

Chrysanthi Paschou, Francesco Raimondo, George Oikonomou, James Pope

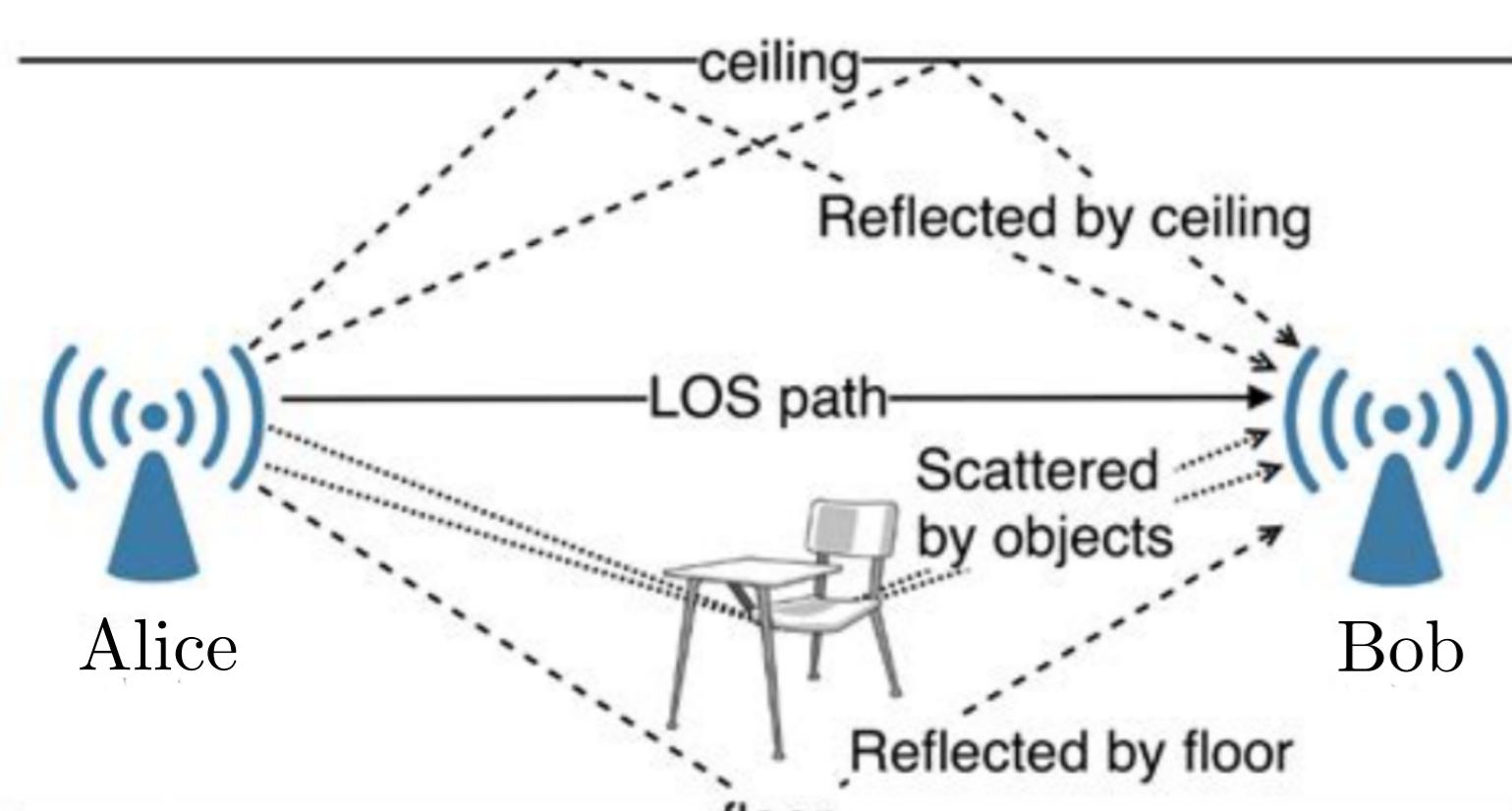
A. Motivation - The key distribution problem

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

B. Background - What is Physical Layer Key Generation?

Stage 1 Channel Probing

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

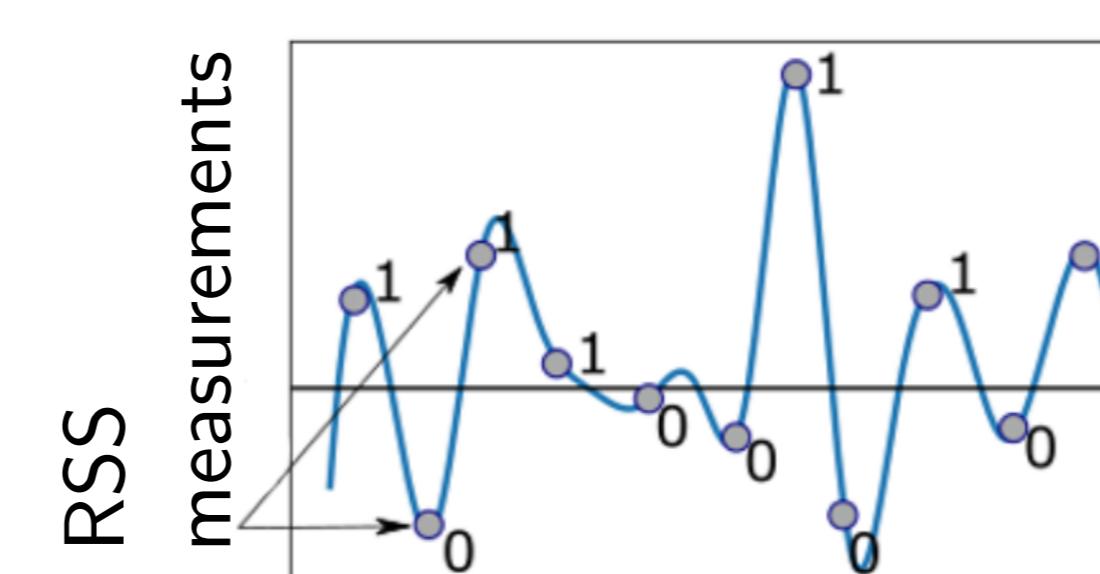


A signal travelling from A. to B. takes the same route as when travelling from B. to A.

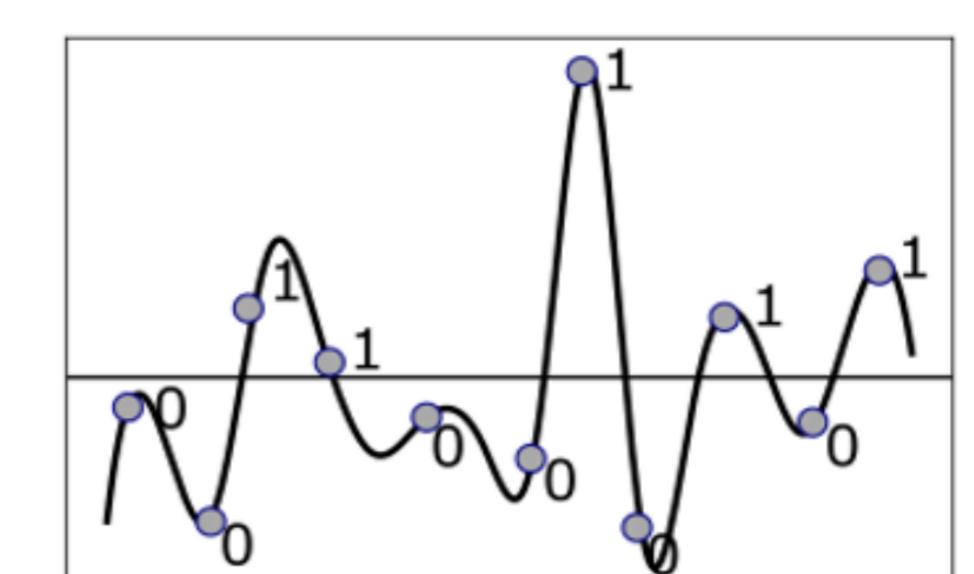
FIGURE I
Alice and Bob wish to agree on a key so that they can use it for upper-layer encryption. They send pilot signal to each other. They both observe similar small-scale fluctuations on the received pilots.

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: **CRiCKET** (Channel Reciprocity for KKey Transport)

- There is no need to reconcile the sequences from Stage 2;
- The correlated sequences from Stage 2 are used for "hiding" a key;
- Alice decides on a key and encodes it on her sequence;
- Bob compares the received codewords with his sequence and retrieves the key.

Interested in the analytical work?



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

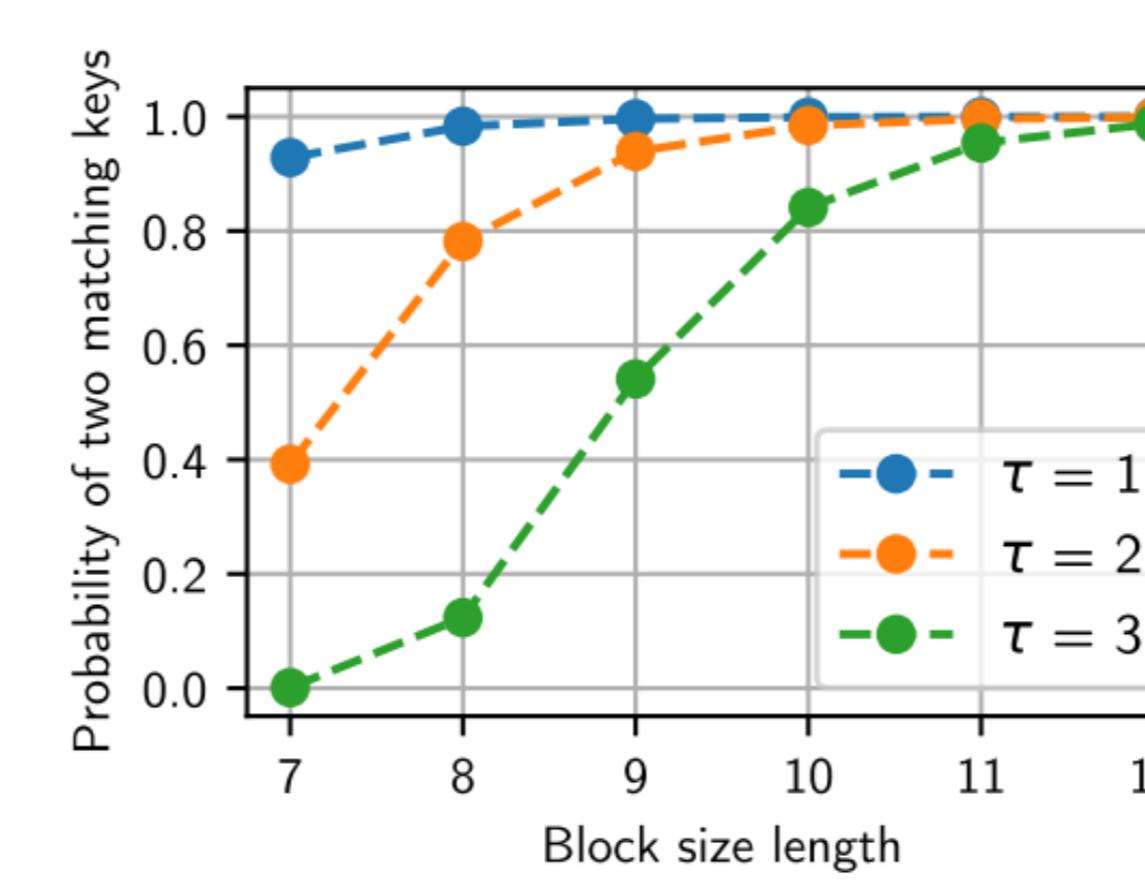


FIGURE II Probability of two matching 128-bit keys when the binary sequences from the quantisation stage disagree by 20%. Variables " τ " and "Block size length" are encoding/decoding parameters.

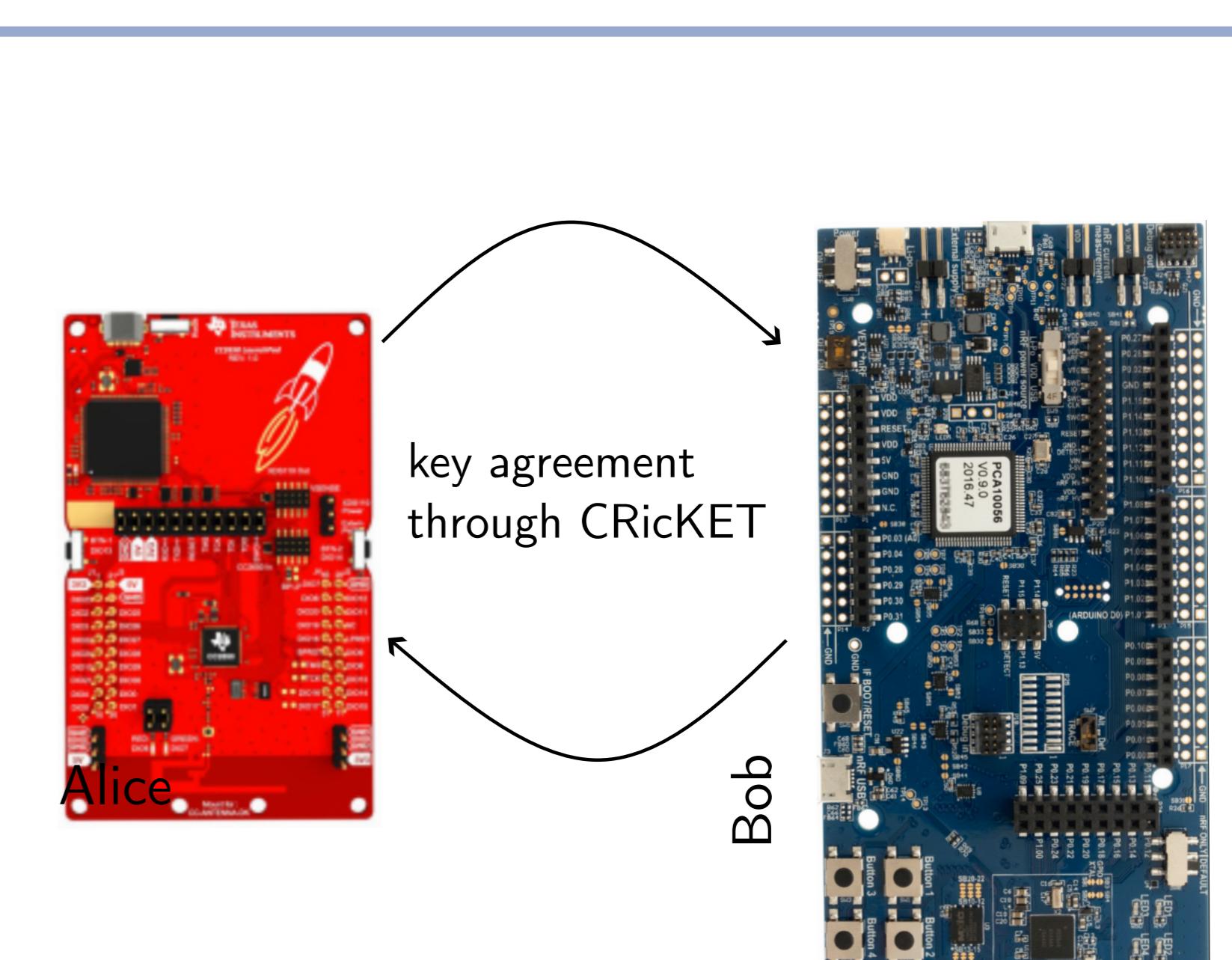


FIGURE III
CRiCKET is implemented in single-chip devices

3 Implementation

- Practicability 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.

- Alice:** TIC cc2650
- 48MHz 32-bit processor
- 128 KB flash
- 20kB RAM
- Bob:** Nordic nRF52840
- 64 MHz 32-bit processor
- 1MB flash
- 256kB RAM