Preventing Relay Attacks and Providing Perfect Forward Secrecy using PHYSEC on 8-bit μC

Christian T. Zenger, Mario Pietersz, Christof Paar
Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
{christian.zenger, mario.pietersz, christof.paar}@rub.de

***Abstract*—Physical Layer Security (PHYSEC) has the potential to offer substantial advantages for key management for small resource-constrained devices. In this paper, we investigate two PHYSEC primitives: *channel reciprocity based key generation* (CRKG) and *channel reciprocity based relay detection* (CRRD). The security and entry performance of such systems in real-world scenarios are still open research questions. Until now, there are no embedded implementations of CRKG and CRRD systems providing resource requirements such as code size, number of clock cycles, and power consumption. Moreover, there are only few experimental verification of practical use cases that provide information about applicability and that cope with current security issues. We fill this gap with measurements carried out with our embedded prototype. By way of example, we address security critical *remote keyless entry* (RKE) use cases. To achieve this we created the first implementation of both PHYSEC primitives on an 8-bit SoC and provide practice-oriented performance results. Further, we provide a fair comparison to elliptic curve cryptography based key establishment schemes.**

I. Introduction

We are in the midst of a paradigm shift towards the Internet of Things (IoT). While more and more, mostly resource-constrained, devices are equipped with an RF interface, the need to secure IoT applications increases as well. Symmetric algorithms provide confidentiality and integrity. Algorithms such as AES already are a good fit for small embedded systems due to their efficiency. In extreme cases specialized ciphers are available which require even less energy. However, asymmetric cryptography, which is often required to provide key establishment and authentication, can be painfully difficult to realize in embedded nodes. Algorithms such as RSA or elliptic curves require an embedded RNG, are typically 2–3 orders slower, need more memory, and, crucially, they are much more energy hungry than their symmetric counterparts.

Recently, key generation schemes that exploit the uniqueness of the wireless channel between two nodes — for example caused by multipath fading — have been proposed as a potential lightweight solution for IoT devices [1], [2]. However, there are few results about how *Physical Layer Security* (PHYSEC) performs in realistic scenarios and especially on embedded hardware, which is the reason for the paper at hand. We investigate in two PHYSEC primitives: *channel reciprocity based key generation* (CRKG) and *channel reciprocity based relay detection* (CRRD). The key idea of CRKG is extracting keymaterial out of the symmetric wireless channel between two parties (cf. Section II for details). Perfect forward secrecy (PFS) can be achieved with continuously re-keying. Frequent key freshness is an important advantage that usually is provided by public-key cryptography only. If performed frequently enough, it can even defeat side-channel attacks. Because relay attacks are easy to execute and allow car thefts, we also examined CRRD for *remote keyless entry* (RKE) systems. In Section III we briefly describe our PHYSEC demonstrator that is compatible to network layer standards for resource-constrained devices: ZigBee, WirelessHART, and 6LoWPAN. The underlying IEEE 802.15.4 MAC layer is suited for ultra-low-power applications. We implemented and tested the relay attack countermeasure against recent relay attacks and provide results in Section IV. For the first time, we provide results of the performance of 8-bit PHYSEC implementations and of a common *Elliptic Curve Diffie-Hellman* (ECDH) key exchange library in Section V. ECDH key exchange is the most efficient established public-key primitive. There are two main contributions in this paper which are summarized below.

Measurements and evaluation: We performed extensive measurements for three different real-world applications using the IEEE 802.15.4 standard. By way of example, we address security critical RKE use cases. and present performance results of a full CRKG system. Further, we introduce a slight but significant modification on the CRRD protocol of Jain et al. [3]. We evaluate both schemes based on measured channel profiles and present results, such as the number of required channel profiles to achieve a security level of 128-bit.

Implementation and comparison: We describe a comprehensive embedded implementation of a CRKG as well as the CRRD protocol. Both are based on reusing the measured *Received Signal Strength Indicators* (RSSI) and were performed on an 8-bit Intel MCS-51 processor. The '8051' is embedded in a true SoC solution for IEEE 802.15.4 applications - called CC2531. The platform is suited for systems where ultra-low-power consumption is required. We provide exact resource requirements such as code size, number of clock cycles, and energy consumption. For a fair comparison to state-of-the-art key establishment approaches we also investigated in well known ECDH implementations.

II. System Model

Now the system model of potential RKE systems using IEEE 802.15.4 communication on 2.4 GHz[1] follows. We describe the problem definition and threat model by using an

[1] Several established RKE systems for cars are communicating using an asymmetric channel (125 kHz car to key, 315/433 MHz key to car). That might have historical reasons. It is conceivable that early immobilizer (first approaches used 125 kHz for communication) and RKE system (first approaches used 433 MHz) realizations were joined. — However, our approach focuses on novel state-of-the-art ultra-low-power communication on 2.4 GHz. These RKE are getting more and more established for alarm systems in smart home environments and maybe also for car keys systems.

examples with RKE for cars. However, the model is also valid for IoT applications. Further, some basic on the wireless channel as well as on PHYSEC are given.

A. Problem Definition and Threat Model

There are two types of modern RKE systems: The first one is called *remote active open* system, where the user presses a button to lock/unlock a (car) door. The second one is called *passive keyless entry and start* (PKES) system, here the user only needs to carry the key in the vicinity of the car to be able to open the door. As shown repeatedly in media and academic, both systems have weaknesses. For securing and authenticating the communication between car and resource-constrained car key both are preloaded usually with a symmetric cryptographic key (a master or group secret). This reduces the effort for key management, but leads to potential successful attacks which scale well for the attacker. Once one node is compromised, the security posture of the entire system (maybe of the entire product batch) collapses. As a result, successful attacks causing automobile manufacturers problems due to economical and reputational damage.

The implications of recent attacks are especially serious because of the scaleability factor of these attacks. For example, Eisenbarth et al. [4] attacked the KeeLoq system using low-cost side-channel attacks and demonstrated that the extraction of a group key compromises a large group of door locking applications. Strobel et al. [5] applied implementation attacks on a state-of-the-art electronic door locking systems. The attack on the master secret compromised the entire product batch. Verdult et al. [6] introduced three reverse engineering based attacks. The attacks addressed an impressive list of potentially target-vehicles. Weak or no random number generators were a critical point in these systems. Also critical are so called insider attacks, where malicious staff of manufacturers or maintenance companies leak key material to criminal organizations. The aim of all attacks is usually to steal the car.

PKES systems permit a low-skill relay attack. Francillon et al. [7] demonstrated successfully attacks on 10 car models from 8 manufacturers. The relay between car and car key is realized with an analog setup to limit processing time, i.e., approximately $100\,\mathrm{ns}$ for a distance of $30\,\mathrm{m}$. The attackers relay the signals of the car to the car key and vice versa to fake the required proximity to open and start the car. The problem is that instead verifying that the correct key is in car's physical proximity, the car verifies if it can communicate with the correct key, assuming that the ability to communicate implies proximity [7]. We define the problem of secure door-to-key relationship as follows: the channel between the car door and the car key requires **authentication**, **relay attack resistant proximity verification** (esp. for the case of PKES), and ***perfect forward secrecy*** (PFS)[2]. Compromising the (initial) key successfully, e.g., by implementation attacks, would not scale to other cars and the window of opportunity for a car thief is limited to the PFS interval.

Maximizing the battery lifetime is another important goal for most manufacturers. Car key hardware consists of a low-power bidirectional communication interface with an internal small $8/16$-bit SoC. Moreover, requirements like PFS cannot be met using symmetric cryptography only. However, while asymmetric approaches allow for improved key management they result in long (sometimes unacceptably long) processing times, a large code size, and considerably energy consumption during encryption and transmission. Additionally, it is a known problem that obtaining good random numbers in constrained devices is a difficult problem that often leads to insecure constructions. Finally, constraints on the battery life makes use of many cryptographic solutions even more difficult.

B. The Wireless Channel as Keying Variable

For wireless communication we consider a time-division duplex radio channel at a carrier frequency of $2.4\,\mathrm{GHz}$.

Reciprocity The wireless channel is symmetric. This can be exploited and utilized by (nearly) simultaneous channel measurement by both transceiver A (e.g., the car) and transceiver B (e.g., the car key). Without taking noise, interference and non-linear components into account the symmetry relies on the principle of *antenna reciprocity* and *channel reciprocity*. For most practical channels the reciprocity property holds and is easily measurable [10].

Diversity: The wireless channel has also the property of *spatial decorrelation*: If uniformly distributed scatterers are given and channel variations occur, such as due to moving scatterers, transmitter, and receiver nodes, the spatial decorrelation is a zero-order Bessel function. The first zero correlation is given after $\approx \lambda/2$ where λ is the wavelength of the carrier [11]. In more realistic scenarios the decorrelation occurs after several λs [12].

Randomness: The radio channel is expected to provide a random behaviour. A complex and dynamic environment leads to unpredictable evolution of wave propagation effects, such as diffraction, scattering, and reflection. Note that temporal independence in the channel measurements is depending on the coherence time of the channel. The coherence time is an approximation based on scenario-dependent assumptions, such as the average speed of moving scatters. For human motion driven scenarios we assume a channel coherence time of approximately $100\,\mathrm{ms}$. We address the resulting security of correlated radio channel profiles by applying on-line entropy estimation, as we show later.

C. Channel-Reciprocity Based Key Generation

Recently PHYSEC and especially CRKG has been investigated as a potential lightweight solution for IoT devices [1], [2]. CRKG was introduced 1995 by Hershey et al. [13] as an alternative paradigm for generating shared secret keys. The approach is based on common measurements of the wireless channel by the sender and receiver, whereby symmetric secret keys are derived from common channel parameters. Besides common randomness the scheme is also inherently secure to attacks. If an attacker's distance to both legitimated nodes is large enough, the channel parameter he observes to each node is uncorrelated and an attack is not possible. The channel measurements are then post-processed, quantized, and relieved

[2] A secure communication protocol is said to have PFS if it protects future sessions against past compromises of secret keys. Achieving PFS means that compromising the authentication key or a session key does not compromise past or future session keys [8]. Therefore, PFS prevents scaling of potential successful attacks [9].

from noise/interference parts. The resulting entropy is collected and utilized as a shared symmetric key.

D. Channel-Reciprocity Based Relay Detection

To prevent relatively simple and generic relay attacks on PKES systems in modern cars, e.g., as demonstrated by Francillon et al. [7], we implemented a modified version of the CRRD protocol by Jain et al. [3]. Jain introduced the idea of using channel reciprocity (and implicit diversity) to detect wormholes. Instead of applying a quantization scheme to verify equality of quantized channel profiles, we simply use raw channel profiles and calculate the Pearson coefficient as a metric for linear correlation. Therefore, the bidirectional channel only needs to be sampled several times by each party. Afterwards, one party sends its measured values encrypted with his session key to the other party. There the received values and the measured values are compared using cross-correlation.

III. Demonstrator Platform

For mobile, long-time channel measurements we implemented the data exchange protocol on three Raspberry Pi 2 platforms. All devices are equipped with a CC2531 USB enabled IEEE $802.15.4$ communication interface[3]. In order to establish common channel probing, A periodically sends data frames to B and waits for acknowledgements. A passive attacker E also receives these request/response pairs. When receiving a probe, all three devices extract RSSI values and, thus, can measure a channel-dependent sequence over time.

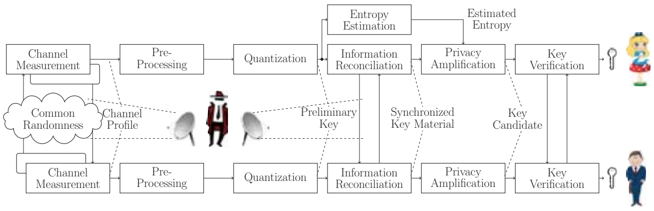


Fig. 1: Modular structure of the CRKG security architecture [14].

For evaluation of the bidirectional channel measurements, the measured sequences are stored and processed locally on a monitor laptop with Matlab. The protocol execution presented ensures synchronized channel measurements within the probing duration of $r_p^{-1} \leq 5\,\mathrm{ms}$. To achieve uncorrelated measurements in time, the sampling rate is adopted to the typical coherence time and therefore $r_s \approx (100\,\mathrm{ms})^{-1}$ for subsequent rounds of channel probing. To evaluate the performance and energy consuption, we simulate different channel sample intervals by downsampling the measurements. We have implemented the key generation protocol illustrated in Fig. 1 [14]. The pre-processing step is optional and comprises reciprocity enhancement and decorrelation schemes. The multilevel quantization scheme proposed in [10] is applyed with a blocklength of 128 samples. The chosen quantization was recommended by Zenger et al. [15] as a result of their security evaluation. In order to cope with disagreements due to imperfect channel-reciprocity, we utilize syndrome-based secure sketch with BCH codes as proposed for CRKG in [16]. Here, we consider a (255, 131, 37)-BCH code. To achieve a security level of 128-bit from the collected key material gained from the common channel, we applied the bit-wise on-line entropy estimation based on SP-draft [17] by NIST. The draft was designed to estimate the entropy of *non-independent and identically distributed* (non-IID) random variables. To determine the security level we adopt the design criteria proposed in [18]. Since information for error correction is exchanged over the public channel, further enhancement of entropy is done in the privacy amplification stage. For privacy amplification we used CBC-MAC as suggested in [19] in combination with the AES block cipher.

IV. Channel Measurements & Results

We ran measurement campaigns for two potential PKES applications. In the first scenario we placed one measurement platform consecutively at several cars. The corresponding counterpart (i.e., the key) was carried by the user who performed natural actions that happen daily in the real-world. The user walks from outside the transmission range to the the car, opens the door being near the door lock, and stays for a period of time close to the car, and drives around.

The second application is a digital locking and access control system usually used in buildings. A digital door lock has usually two types of distinctive communication partners: corresponding keys and the *central control system* (CCS) [5]. For both measurement series, the testbed is applied at the premises of our research group, which is an office area in a modern building (completed in 2010). We mounted a Raspberry Pi platform at different doors on a floor whereby the communication interface was located as close as possible to its most probably position. The corresponding other platform simulates an usual working day of an employee, carrying around the key in his pocket, opening doors, walking around the floors, and finally working in his office in transmission range to his door lock. For the door-to-CCS scenario, we performed an additional long-term measurement between the door locks and the wireless interface of the CCS. The wireless interface is mounted at a predestined access point position that is pretty much the center of the office area in our case. The main difference between both scenarios is that in the door-to-CCS setup both parties are stationary and therefore fluctuations of the channel only achieve due to external motion, whereby in the door-to-key setup mostly high fluctuations are experienced due to key's mobility.

As introduced before, all measurements were done for three parties, whereby the third party is a potential passive attacker. The attacker has always a fixed location within the close proximity of $50\,\mathrm{cm}$ of one of both legitimate devices. For each scenario, we measured the bidirectional channels approximately 1 million times.

The security level of the collected key material is determined by multiplying the on-line estimated conditional bit-entropy $H_{est.}(X)$ with the number of collected bits (cf. [18]). The conditional bit-entropy $H(X|Y) = H_{est.}(X) \cdot \frac{k}{n}$ is derived by weighting the estimated bit-entropy $H_{est.}(X)$ with the remaining entropy factor k/n (due to reconciliation overhead,

[3] http://www.ti.com/tool/cc2531emk

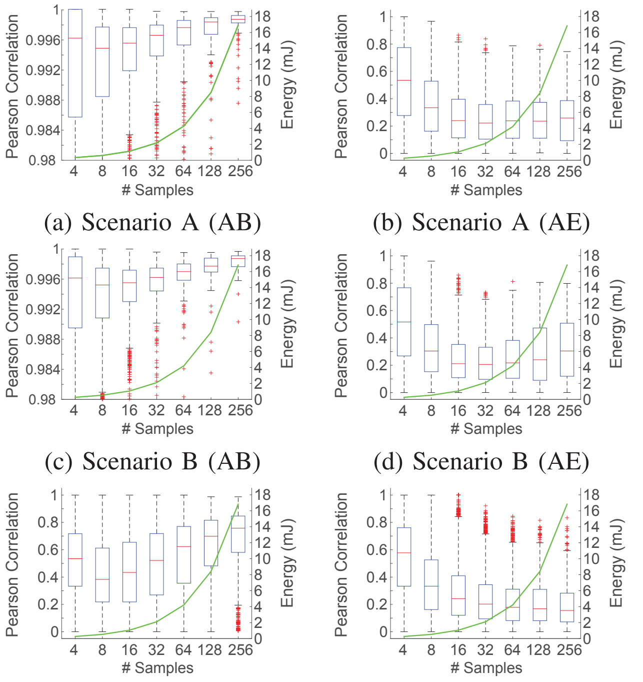
an information loss of $n - k$ occurs). The resulting *entropy extraction rate* (EER) as well as the number of required channel measurements to achieve a security level of 128-bit for each scenario is summarized in Table I. Both results depend on the fading properties of the channel. In general it is true that if the transceivers or the scatters in the environment are moving, a higher EER can be achieved and a lower number of measurement is required than if a static scenario is given. This behavior is also reflected by our results. Here the static scenario C) requires between 25 to 38 times more channel measurements than the scenarios with movement A) and B). This is because of the lower entropy, due to temporal correlations. If low channel fluctuations are given, noise and interference are dominating which leads to additional bit disagreements.

TABLE I: Experimental results of key extraction. Shown are the number of required channel profiles as well as the resulting EER (based on [17]) to achieve a security level of 128-bit.

Scenario	EER [bit/s]	# of measurements
A) Key-to-Car ($k = 199$)	1.011	1264
B) Key-to-Door ($k = 171$)	1.555	819
C) Key-to-CCS ($k = 131$)	0.041	31328

We performed for each scenario the relay attack described in [7]. Therefore, we synchronized the channel measurement and placed the two relays between both key and car (or door). With slight position modifications we rerun the experiment ten times for each scenario. Our evaluation results for relay attack detection are provided in Fig. 2. The key factor by using the Pearson coefficient is the number of required samples to achieve a reliable detection rate (and a low false-positive rate). The distribution of the correlation coefficients for each scenario (A,B,C) is illustrated in boxplots. Therefore, we divided the 1 million measurements per scenario by different block sizes $(4, 8, ..., 256)$. Fig.s 2(a),(c),(e) show the results of the legitimate users, whereby Fig. 2(b),(d),(f) show the results of a potential relay attacker. For the mobile scenarios A) and B) the median correlation between key and lock is very high (corrcoef$>$ 0.98) independent of the chosen number of samples and the median correlation for an relay attacker is lower than 0.55. These results are the basis for a binary hypothesis test that leads to high success rates and low false-positive rates independent of the block size. Therefore, reliable detection performance can be reached with only four measurements, which require approximately 400 ms. Note that for mobile scenarios (the key needs to be carried in a regular way) the duration of the CRRD mechanism is within the acceptance threshold. For the static scenario C), the course of the 'legitimate' median correlation for increasing block sizes starts by 0.54 and mostly linearly increases to 0.76 for the largest evaluated block size (256), whereby the course of the 'attacker' median correlation starts by 0.58 and decreases to 0.16. For a block size larger than 32 the interquartile ranges of the boxplots do not overlap and therefore the probability of the reliability of high success rates and low false-positive rates increases with increasing number of samples.

As pointed out in [20], Jain et al.'s scheme cannot detect *adaptive wormholes*. Adaptive wormholes require digital processing and real-time adoption of the transmission power. Such an attack is not feasible with an analog setup as demonstrated by Francillon et al. [7]. However, an open research question is if the attacker can easily influence RSSI readings. Therefore, detecting adaptive wormholes is left for future work.



(a) Scenario A (AB)

(b) Scenario A (AE)

(c) Scenario B (AB)

(d) Scenario B (AE)

(e) Scenario C (AB)

(f) Scenario C (AE)

Fig. 2: Evaluation results of the CRRD scheme: The distribution of the correlation coefficient is given for the *no-attack* cases (a, b, c) and for the *relay attack* cases (b, d, f). Further, the required energy consumption for relay detection is given.

V. Embedded Implementation & Results

In many cases, IoT-platforms are small embedded devices without continuous power supply. The selection of algorithms therefore is more restricted compared to platforms with external power supply and strong CPUs. In particular, resource requirements such as code size, the number of clock cycles, and power consumption are the crucial factors. The targeted platform CC2531 is a true SoC solution for IEEE 802.15.4 applications, such as ZigBee, WirelessHART, 6LoWPAN. By combining an industry-standard enhanced 8-bit Intel MCS-51, from the 8051 familiy, with an IEEE 802.15.4 RF transceiver and providing various operating modes, the chip is suited for systems where ultra-low-power consumption is required. The microcontroller has 32-KB in-system programmable flash memory and 8-KB RAM[4]. Basis for the embedded implementations on the 8051 MCU itself is a firmware provided by Texas Instruments with a tiny operating system and an IEEE 802.15.4 compliant MAC. We created a new thread that executes the algorithms only when the system is active anyway and does not influence the sleep cycles of the system in order to save energy. The algorithms for quantization, reconciliation, and privacy amplification (cf. Section III) of CRKG protocol as well as the countermeasure against relay-attacks were implemented in

[4] http://www.ti.com/lit/ds/symlink/cc2531.pdf

plain C and are optimized for 8-bit processor architectures. We verified each part of the protocol implementation with test vectors from Matlab and the communication with Wireshark.

After measuring each protocol step separately in time, we calculated the actual energy consumption by multiplying the elapsed time with the power consumption for different modes provided by the data sheet[5], i.e., computing $19.5\,\mathrm{mW}$, receiving $61.5\,\mathrm{mW}$, and sending $86.1\,\mathrm{mW}$. For each mode, we verified the plausibility by measuring the power consumption using a shunt circuit and a PicoScope. Therefore, we measured the current and integrate over time. The required energy overhead for additional bidirectional channel measurements using standard frames increases linearly with its number, as illustrated in green and on the right y-axis in Fig. 2. For the mobile scenarios A) and B), four channel samples are needed which requires $263\,\mu\mathrm{J}$ only. In the static scenario C) 64 channel samples are needed which leads to an energy overhead of $4.207\,\mathrm{mJ}$. Because energy constraints are essential for IoT applications, the proposed relay detection works very well for mobile scenarios but might be not suitable for static scenarios.

In state-of-the-art IoT-systems, (lightweight) public key cryptography (PKC) has been implemented to establish dynamic key management for resource-constrained devices. In particular, elliptic curve cryptography (ECC) is the most efficient algorithm among the established PKC algorithms. The efficient implementation on embedded systems has been well investigated, e.g., [21]. However, the known results on 8-bit implementations do not include the energy consumption of transmitting, listening for and receiving data (see [22], [23] for a detailed discussion). For key-establishment we implemented the ECDH protocol that requires to perform four point multiplications and to send two messages (cf. [23]). The results of the ECDH reference implementation[5] are summarized in Table II. Note that ECDH requires additionally a strong random number generator and statistical tests in each device, which requires also additional energy. Nevertheless, for simplicity this is out of scope in this work. As highlighted in the table, our goal is to achieve a 128-bit security level. The approximate resource overhead of the computation of each component of the CRKG scheme is provided in Table III. The results of our implementations are based on input vectors of 128 RSSI values for the quantization schemes and, therefore, on 256 bit input for reconciliation and privacy amplification. The code size of ECDH is 7.7 times larger than the one of CRKG. The RAM required are distinguished by approximately $210\,\mathrm{byte}$.

TABLE II: Approximate resource overhead of each of the PKC blocks of reference implementation on a 8-bit MCS-51.

Block name	Size [Kb]	RAM [Kb]	# of cycles	Energy [mJ]
secp128r1	7.264	0.160	263, 111, 112	80.167
secp192r1	5.856	0.240	572, 444, 446	174.417
secp256r1	**8.749**	**0.320**	**1,734,400,000**	**528.45**
secp384r1	5.643	0.480	4, 096, 000, 000	1, 248

Putting all together, the overall energy cost of CRKG is between $5.4\,\mathrm{mJ}$ and $209.6\,\mathrm{mJ}$, while ECDH requires an energy of $528.5\,\mathrm{mJ}$ (see Table IV). The results show that in mobile scenarios the energy consumption for CRKG is only between $1\% - 1.6\%$ of ECDH; and even in a static scenario the energy consumption is only 39% of ECDH. However, the variance of the number of required channel profiles for the scenarios is large and no guarantee of a reliable duration can be made. The number of profiles is also connected with the duration time of rekeying. Where ECDH can be performed within $27.1\,\mathrm{s}$ (on one 8051's side), CRKG's duration time depends strongly on the sampling interval and the entropy gained from the environment or movement. Assuming that 20 channel profiles are collected within one second and each time the user locks/unlocks his car. Than rekeying (128-bit) takes about 80-130 transactions with the car key (tremendously longer in static scenarios).

TABLE III: Approximate resource overhead of each of the component blocks of reference impl. on a 8-bit MCS-51.

Block name	Size [Kb]	RAM [Kb]	# of cycles	Energy [μJ]
Quantization	0.208	0.029	11, 876	7.235
Reconciliation	0.771	0.018	1, 325, 556	801.665
Privacy amp.	0.158	0.064	7, 773	4.719
Σ	1.137	0.111	1, 345, 205	803.619

Related work on public-key implementations used Curve25519 implementations (instead of secp256r1) on an 8-bit AVR ATmega μC [24] (instead of 8051). The paper provides the number of clock cycles, code size, and RAM usage. Unfortunately no power consumption is provided. Nevertheless, the paper shows that the code size is between $9,912$ and $17,710$ bytes, which is 9 to 17 times bigger than CRKG. The number of clock cycles is between $14,146,844$ and $13,900,397$. Based on the execution time and the power model of the 8051, we estimate the energy consumption to $8.6\,\mathrm{mJ}$, which is only slightly worse than CRKG.

TABLE IV: Energy analysis of CRKG and ECDH (128-bit).

Protocol	Scenario	Comp.	Msg. transfer	Total energy [mJ]
CRKG	A	0.193	8.034	8.227
	B	0.187	5.206	5.393
	C	199.133	10.512	209.645
ECDH	A,B,C	528.45	0.064	528.514

VI. Related Work

Re-keying is a possible countermeasure against key extraction attacks, such as many side-channel attacks on symmetric ciphers. However, it is not easy to achieve in resource-constraint applications. Related threats we like to address with this work are low-cost attacks, such as [4]–[6]. In [4] and [6] car entry and start systems were reverse-engineered, weaknesses were identified and successfully exploited. Proposal [5] pinpoints various security vulnerabilities of a widespread digital locking system, e.g., the application of master keys as well as the fact that obtaining 'good' random numbers in constrained embedded systems is hard. Prevention of relay attacks on low-cost hardware is challenging, too. Several attacks point out that today's countermeasures are not sufficient [7], [25]. Francillon et al. presented in [7] low-cost relay attacks on 10 car models from 8 manufacturers. To avoid attack

[5] https://github.com/iSECPartners/nano-ecc

detection due to long signal delays, an analog setup without digital signal processing was applied. Wormhole attacks[6] in mobile ad-hoc networks were introduced in [25]. Several countermeasures against relay attacks, such as *shielding the key*, *removing battery from the key*, *software only modifications*, *access control restrictions*, and *hardware modifications*, are presented in [7]. However, all countermeasures inflict usability somehow. Further countermeasures based on signal strength, signal propagation properties, or out-of-band communication are identified as weak or not suited for embedded systems. Approaches based on measuring the round trip time (RTT) from the car to the key to the car to verify an upper-bound on its distance to another are called *distance bounding* and have been extensively studied. There are surveys that explore the fundamentals and requirements of distance bounding protocols [7]. The previous work can be summarized as follows. RF distance bounding is the only viable option, since ultrasonic distance bounding is vulnerable to relay attacks. RF distance bounding is challenging because of the the processing time and the time variance of the processing time at the low-resource key. Given a maximum distance of $1\,\mathrm{m}$ at which the key should be able to open the door, the RTT should be less than $6\,\mathrm{ns}$. Furthermore, a relay attack over a distance of $30\,\mathrm{m}$ adds approximately $200\,\mathrm{ns}$ to the RTT. Assuming a standard SoC with $32\,\mathrm{MHz}$, one cycle is $31.1\,\mathrm{ns}$ which makes the required reaction and processing time unrealizable on today's platforms. Recently, wormhole detection in mobile ad-hoc networks based on channel-reciprocity has been proposed by Jain et al. [3]. Krentz et al. [20] presented a wormhole detection mechanism based on Jain et al. [3] for 6LoWPAN networks. This version might be suitable for RKE systems as well, however, it requires transmission power adoption and frequency hopping which introduces new security and energy issues. Nevertheless, it might be a good solution for static scenarios. We provide a simple extension of the work of Jain et al. and provide measurement results in RKE scenarios.

VII. Conclusion

Energy efficient dynamic key management and PFS realizations for resource-constrained battery-powered devices are crucial problems in embedded applications. Recent attacks underline the fact that a solution is urgently required. We investigate PHYSEC — which has been proposed as a potential lightweight solution for IoT devices [1], [2] — as an extension framework for small embedded platforms to address the problem of dynamic key management as well as of relay attacks. To address real-world requirements, we presented an ultra-low-power IEEE 802.15.4 testbed and an extensive experimental evaluation for different security critical IoT-applications. We identified resource-efficient PHYSEC architectures — especially for CRKG — and demonstrate the first implementation of a CRKE and a CRRD schemes on an 8-bit processor to verify the applicability of PHYSEC for resource-constrained platforms. By providing resource requirements, we present a comparison between CRKG and ECDH implementations and demonstrate that PHYSEC can be an efficient, lightweight alternative to conventional schemes in diverse scenarios. The environment has a strong impact on the extracted entropy and on the reliability of PHYSEC performance. Future work might identify static scenarios and classify channel profiles regarding its potential amount of entropy to reduce energy.

[6] We note that relay attacks have been similarly applied in mobile ad-hoc networks (MANET) and are called wormhole attacks [26].

References

[1] W. Trappe *et al.*, "Low-energy security: Limits and opportunities in the internet of things," *IEEE Security & Privacy*, 2015.

[2] C. T. Zenger *et al.*, "Exploiting the physical environment for securing the internet of things," in *New Security Paradigms Workshop*, 2015.

[3] S. Jain *et al.*, "Preventing wormhole attacks using physical layer authentication," in *IEEE Wireless Communications and Networking Conference*, 2012.

[4] Eisenbarth *et al.*, "On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme," in *Advances in Cryptology*, 2008.

[5] D. Strobel *et al.*, "Fuming acid and cryptanalysis: Handy tools for overcoming a digital locking and access control system," in *Advances in Cryptology - 33rd Annual Cryptology Conference*, 2013.

[6] R. Verdult *et al.*, "Dismantling megamos crypto: Wirelessly lockpicking a vehicle immobilizer," in *USENIX Security Symposium*, 2013.

[7] A. Francillon *et al.*, "Relay attacks on passive keyless entry and start systems in modern cars," in *Network and Distributed System Security Symposium*, 2011.

[8] A. Menezes *et al.*, *Handbook of Applied Cryptography*, 1996.

[9] C. T. Zenger *et al.*, "Preventing scaling of successful attacks: A cross-layer security architecture for resource-constrained platforms," in *Cryptography and Information Security in the Balkans*, 2014.

[10] S. Jana *et al.*, "On the effectiveness of secret key extraction from wireless signal strength in real environments," in *Mobile Computing and Networking, MOBICOM*, 2009.

[11] A. Goldsmith *et al.*, *MIMO Wireless Communications*, 2010.

[12] S. Eberz *et al.*, "A practical man-in-the-middle attack on signal-based key generation protocols," in *Computer Security, ESORICS*, 2012.

[13] J. E. Hershey *et al.*, "Unconventional cryptographic keying variable management," in *IEEE Transactions on Communications*, 1995.

[14] C. T. Zenger *et al.*, "A novel key generating architecture for wireless low-resource devices," in *Secure Internet of Things (SIoT)*, 2014.

[15] C. Zenger *et al.*, "Security analysis of quantization schemes for channel-based key extraction," in *Wireless Communication Security at the Physical Layer*, 2015.

[16] M. Edman, A. Kiayias, Q. Tang, and B. Yener, "On the security of key extraction from measuring physical quantities," *CoRR*, 2013.

[17] E. Barker *et al.*, "Recommendation for the entropy sources used for random bit generation," *Draft NIST Special Publication*, 2012.

[18] C. T. Zenger *et al.*, "On-line entropy estimation for secure information reconciliation," in *Wireless Communication Security at the Physical Layer*, 2015.

[19] Y. Dodis *et al.*, "Randomness extraction and key derivation using the cbc, cascade and HMAC modes," in *Advances in Cryptology*, 2004.

[20] K. Krentz *et al.*, "6lowpan security: Avoiding hidden wormholes using channel reciprocity," in *Trustworthy Embedded Devices*, 2014.

[21] A. Liu *et al.*, "Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks," in *Information Processing in Sensor Networks*, 2008.

[22] D. Galindo *et al.*, "On the energy cost of authenticated key agreement in wireless sensor networks," in *Wireless Communications and Mobile Computing*, 2012.

[23] J. Großschädl *et al.*, "The energy cost of cryptographic key establishment in wireless sensor networks," in *ACM Symposium on Information, Computer and Communications Security*, 2007.

[24] M. Düll *et al.*, "High-speed curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers," *Des. Codes Cryptography*, 2015.

[25] A. K. Abdelaziz *et al.*, "Survey of routing attacks and countermeasures in mobile ad hoc networks," in *Computer Modelling and Simulation*, 2013.

[26] Y. Hu *et al.*, "Wormhole attacks in wireless networks," in *IEEE Journal on Selected Areas in Communications*, 2006.