Noisy Carrier Modulation for HF RFID

Gerhard P. Hancke
University of Cambridge
gh275@cam.ac.uk

***Abstract* - Radio-frequency tokens are vulnerable to eavesdropping. Several schemes have been proposed that use additional devices to generate cover noise, or bit collisions, in order to protect communication between a reader and a token. I discuss the practical weaknesses in current bit-blocking schemes and propose an alternative implementation where the tokens modulate their reply onto a noisy carrier provided by the reader. I believe that this modification resolves some weaknesses of bit-blocking protocols and is also easier to implement as it does not require additional blocking devices. This method can also be used to simply add noise to the backward communication channel in order to complicate the recovery of eavesdropped data.**

I. INTRODUCTION

RFID systems are vulnerable to eavesdropping and a number of parties have raised privacy concerns with regards to personal data being leaked or specific devices being tracked [1]. Even though confidentiality can be provided by implementing suitable application layer encryption, the cost and hardware constraints limit the amount of logic than can be accommodated. This means that some tokens contain only data storage elements with no security mechanisms. In certain cases key exchange is not possible as the device has no cryptographic means to do so and without a shared key no data can be exchanged confidentially. The traditional way of deriving a session key from the token's unique identifier by using a master key might also not be feasible since the token responds with a random identifier to prevent tracking. Even in systems with application layer security eavesdropping is still a problem, as in the case of e-passports where there is a potential weakness in the session key algorithm, which allows an attacker to store the eavesdropped data and execute a brute force attack offline [2].

Any mechanism that prevents an attacker from eavesdropping data without adding to the hardware complexity of the token would therefore be useful. It has been proposed that additive noise on the communication channel can be used to protect data. Cover noise proposals for the key-less exchange of a secret are especially useful in scenarios involving devices with limited cryptographic resources and can also be applied to ubiquitous environments, where pairing and key-exchange often happen between devices that have never interacted before. A number of protocols have been suggested in the last few years that use bit-collisions, or blocking, in the communication channel to protect an RFID token's privacy [3, 4] and as a method to exchange keys, or data, between a RFID reader and a token [5, 6]. These protocols make the security assumption that tokens, or devices acting like tokens, are indistinguishable to an attacker. The authors argue that distinguishing between different devices are hard and that it would require special hardware, collusion between different attackers or 'fingerprinting' of tokens.

I show that an attacker can distinguish between a response and corresponding cover noise because of simple differences in the devices' communication. The attacker would need no more advanced equipment beyond that needed to perform an eavesdropping attack. I therefore propose an alternative implementation of current bit-blocking schemes, where the characteristics of the cover noise are chosen in such a way that it obscures differences in the devices' phase and modulation depth.

1.1 Related Work

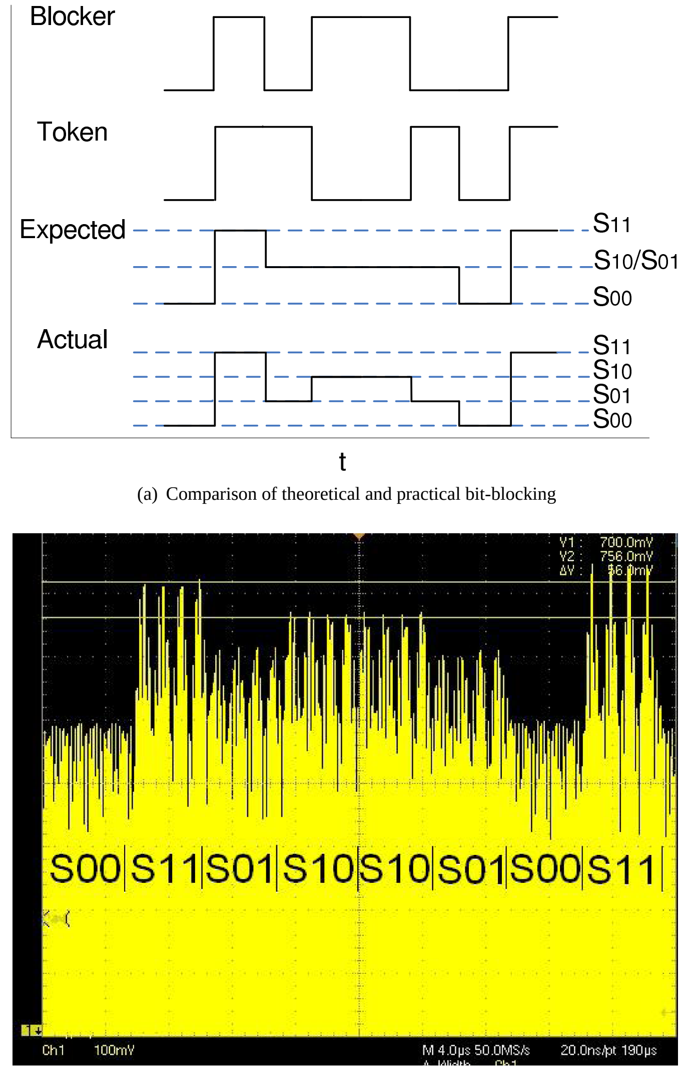
The idea of exchanging data securely by using characteristics of noise on the communication channel, and without the need for a shared secret, has been around for decades, following on from the work of Shannon [7]. In 1975 Wyner [8] described the 'wire-tap' model. The sender transmits some data $y(t)$, which is corrupted by noise $N'(t)$ and $N''(t)$ on the communication channel. The intended recipient receives $x(t) = y(t) + N'(t)$ while the attacker receives $z(t) = y(t) + N''(t)$. The basic idea is that $N'(t) << N''(t)$ and as a result, based on the information theory regarding noise and channel capacity, the intended recipient can recover the data while the attacker cannot. Several ideas, following this model, have been proposed in the RFID environment [9, 10]. The problem with these proposals are that, even though they are theoretically shown to be secure, there are no practical assurances that $N''(t)$ will always be sufficient to prevent an attacker from recovering the data.

It is therefore a logical progression to intentionally add noise to the communication channel. Within the RFID environment there are several papers suggesting that a system should intentionally cause bit collisions on the channel between the reader and the token, thereby scrambling the true value of the token's response. Bit-blocking works as follows (assuming there are two devices): The devices, which are synchronized and identical in terms of their communication channel, transmit a data sequence at the same time. If both transmit a '1' the result is symbol S_{11} and if both both transmit '0' the result is symbol S_{00}. If the devices transmit a '1' and a '0' respectively the result is either S_{01} or S_{10}. Bit-blocking works on the assumption that $S_{01} = S_{10}$ and that the attacker cannot determine who sent the '1' and who sent the '0'.

This principle can be used by token blockers to ensure privacy, or provide access control, by hiding the response from tokens in the presence of an untrusted reader [3,4]. For the purpose of this section I concentrate more on bit-blocking as used in the key-exchange protocols, such as the Noisy Tag (NTP) and NFC Key Agreement (NKA) protocols suggested by Castelluccia, et al. [5] and Haselsteiner, et al. [6] respectively. These two protocols are virtually the same: First there is an exchange phase where the blocker and another device transmit random numbers at the same time. This is followed by a reconciliation phase when all S_{11} and S_{00} symbols are discarded and the secret key is refined from the S_{01} and S_{10} symbols. The two protocols do however differ slightly in terms of practical implementation. In the NTP another token, referred to as the 'noisy tag', is used as the blocker. The reader shares a secret with the blocker, so it can predict the bit-blocking sequence and as a result it can recover the data transmitted by the token. In NKA two NFC enabled devices synchronize and then transmit data at the same time. The receiver knows the blocking sequence it used, so it can recover the data transmitted by the other device.

Both the NTP and NKA protocols are useful assuming that it is practical to ensure that the $S_{01} = S_{10}$ condition holds. As the authors mention themselves, this requires that both devices' data must match in amplitude and phase. Figure 1(a) shows an example where S_{01} is not equivalent to S_{10}. I looked at several ISO 14443A tokens, all containing a NXP Mifare 1K IC, to see if the communication of commonly used RFID tokens vary in amplitude and phase:

- **Amplitude**: A difference in amplitude of S_{01} and S_{10} is likely to occur if there is a difference in the modulation depth of the blocker



(a) Comparison of theoretical and practical bit-blocking

(b) Bit collision between the replies of two ISO 14443A tokens

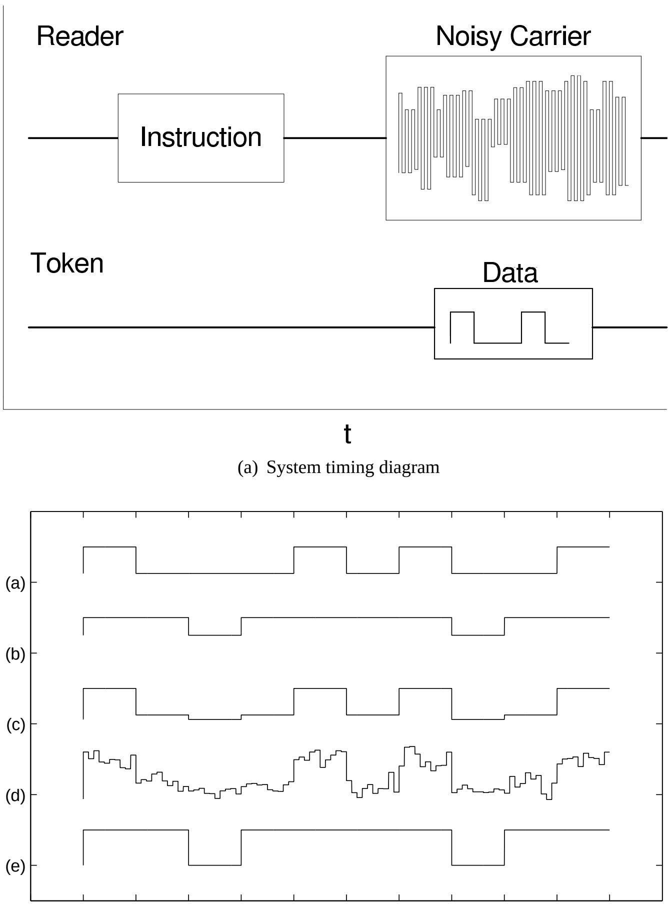
FIGURE 1 - DISTINGUISHING BETWEEN COVER NOISE AND THE TOKEN'S RESPONSE

and the token. The modulation depth, or the change in amplitude of the carrier during data modulation, is determined by the antenna inductance, the resonant capacitor, modulation impedance and even orientation (since it effects antenna coupling). Figure 1(b) shows that the synchronized response of two of these tokens clearly has four distinct levels for S_{01}, S_{10}, S_{11} and S_{00} respectively, with up to 250 mV difference between S_{11} symbols for different cards. As a result an attacker with eavesdropping equipment, in this case a tuned copper loop antenna and an amplifier, might be able to distinguish between the two sequences.

- **Phase**: I found that phase was less of a practical issue. Tokens have the ability to synchronize relatively well - as illustrated by the anti-collision procedure in ISO 14443A cards [11]. The tokens I tested all responded within 0.1 μs of each other, which is roughly equivalent to 1% of a bit period. A determined attacker could probably fingerprint a card in this way, but at this stage variability in the amplitude of the modulatation depth is an easier option.

II. NOISY CARRIER MODULATION

The NKA protocol suggests that each device synchronizes phase and amplitude before commencing with the rest of the protocol. In the NTP protocol a noisy token is used, which I assume is similar to the other tokens to ensure that $S_{01} \approx S_{10}$. Suggesting that two devices synchronize in time is feasible, but matching modulation depth is more tricky, since it involves varying the RF carrier or tuning parameters. An alternative solution is to introduce randomness to the communication. This can prevent the attacker from determining whether the symbols S_{01} or S_{10} were transmitted. Randomizing the characteristics of the physical communication medium, by continuously moving devices involved in key paring, have been suggested to prevent an attacker from distinguishing between devices using received signal strength [12]. In the RFID environment, however, it is hard to move the reader, so I had to look for another solution.



(a) System timing diagram

(b) Example of bit-blocking with additional noise

FIGURE 2 - NOISY BIT-BLOCKING PROTOCOL

I propose that the blocker uses a layer of band-limited AWGN in addition to bit-blocking to hide differences in the physical characteristics of the tokens' communication. Figure 2(b) shows an example of how this works: (a) and (b) are the blocking sequence and data and (c) is the combination of the two. The fact that (c) has two distinct levels for S_{01} and S_{10} is hidden by adding random noise (d), but the data can still be recovered (e). In a way this merges bit-blocking with the concept of hiding data in random noise. The exchange phase of my protocol is shown in Figure 2(a). This is followed by a resolution phase where S_{11} and S_{00} are discarded and a key K_t is refined from the remaining symbols. This phase is the same as described in the NTP and NKA protocols and $2n$ bits need to exchanged to refine a n bit secret.

I assume that the reader is trusted and that it attempts to exchange a key with a trusted token in the presence of a passive attacker. I do not consider active attacks and my scheme does not prevent unauthorized readers from communication with a token. It could, however, be used to exchange a secret between a RFID blocker/proxy device and a reader, which can then be used to set up authentication and access control conditions. If the user already carries an intelligent RFID proxy device, which uses bit-blocking for privacy, the scheme can be modified so that this device can randomize its blocking bits by adding AWGN, therefore making it difficult for an attacker to distinguish the difference between the tokens and the proxy. NFC has already been advocated as a out-of-band method in ubiquitous environments for setting up communication parameters before communication commences on another medium [13]. My proposal can be extended to active devices, such as mobile phones, that use the 'passive' mode described in the NFC standard [14]. My scheme can also be used, in addition to conventional cryptography, to provide eavesdropping resistance. For example, it will make brute force key searches on e-passports much harder if some of the attacker's eavesdropped cipher text bits are incorrect. In this case the reader will transmit blocking-bits whenever the token responds with data.

2.1 Practical Implementation

Practically my scheme differs from current blocker implementations. I also propose that the reader itself acts as the blocker. This makes the system simpler as the user does not need to carry an additional device, which shares a secret with all readers that are encountered. near-field communication differs from conventional RF communication, since the token does not transmit a signal in the conventional sense. The token modulates data onto a carrier transmitted by the reader in very close proximity, by changing its impedance [15]. Cover noise can therefore by added by generating a 'noisy carrier' onto which the token's data is modulated. The only additional hardware required by the reader is a noise generator that combines the output of a PRN, which generates the bit-blocking sequence, and an AWGN noise source. The result is modulated onto the carrier at the same time as the token's data. After the reader removes the carrier and subtracts the noise the data can be recovered. This does not require a special token. In fact, tokens adhering to ISO standards that specifies near-field communication can be used, as the bit-collision process is transparent to the token.

Implementing the cover noise in this way also provides protection against attackers who try to recover data with the help of directional antennas. When the noise is generated by a third party that is not in close proximity, e.g., a device covering the whole room, or if two devices both transmit data, e.g., the NKA protocol with 'active' NFC, then an attacker can possibly isolate the data response (or the cover noise sequence) by aiming his antenna at a specific device. In near-field communication the token's response is modulated onto the signal originating from the reader. The attacker eavesdrops this signal, not a signal from the token, so the cover noise sequence and the data response should have degraded equivalently irrespective of the spatial orientation of the reader and the token relative to the attacker. This means that the attacker has minimal chance to separate cover noise and response data because of differences in the positioning between the token and reader. In the case of 'active' NFC the very short operating distance might also make it difficult for the attacker to distinguish between the blocking and data sequence.

III. Results

The attacker does not know the noisy-bit blocking sequence, so he has to try and recover the data by removing the noise through alternative means. A common way to reduce the effect of noise is to average several recordings of the same signals. I do not consider this option, because the attacker does not have multiple recording as the transaction is run only once. For my simulation I integrate over an entire bit period and make a decision about the symbol based on the result. This is a special case of the correlation demodulator when the base functions are rectangular and is an optimum receiver used for data recovery in the presence of AWGN so it works well for testing the effectiveness of the noisy addition [16, pp 233–244]. I assume that the attacker knows exactly when the data is sent and that he can guess the bit period for each symbol without performing clock recovery. The attacker discards

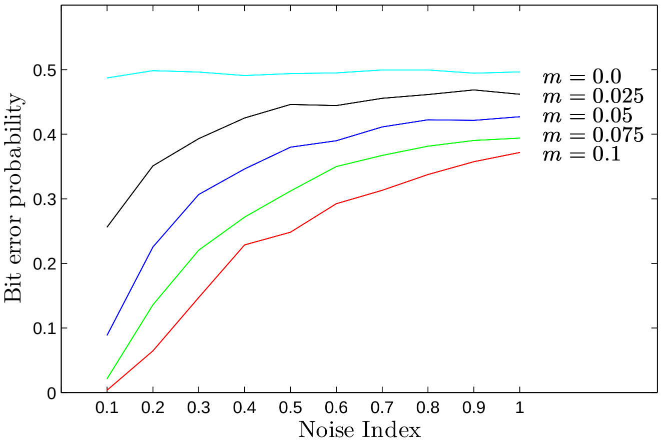


Figure 3 - Simulated results for noisy carrier modulation

symbols S_{11} and S_{00}, and calculates K_{A} based on his knowledge of S_{10} and S_{01}.

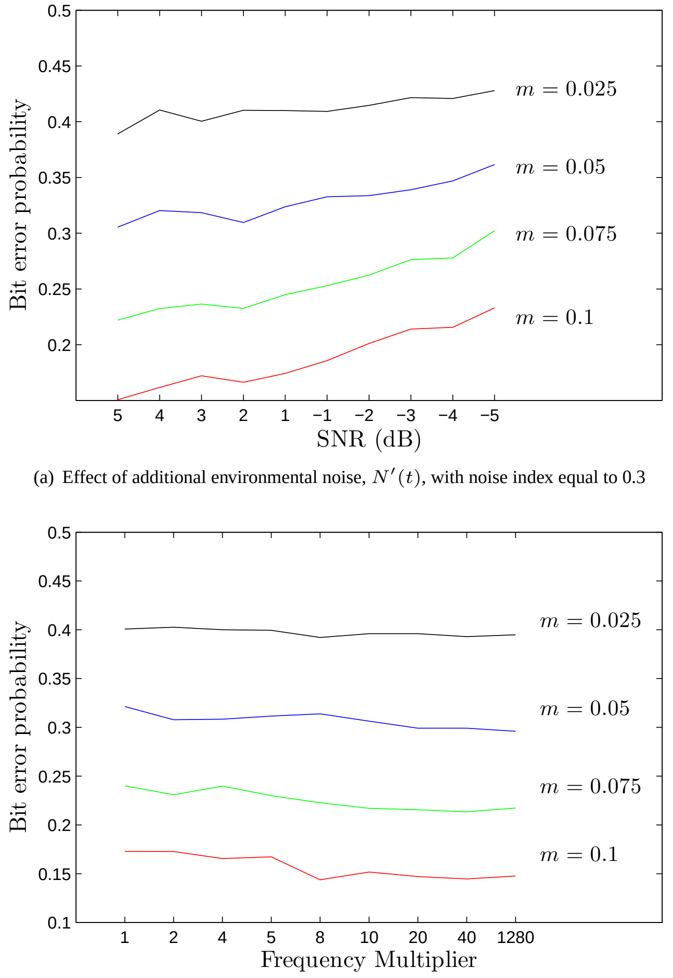
For modeling, the maximum value of the larger symbol is set equivalent to 1 and the value of the smaller symbol is set to $1-m$, e.g., in Figure 3 $S_{10} \approx$ 700 mV and $S_{01} \approx$ 660 mV so $m \approx$ 40 mV ≈ 0.055. The sequence of S_{10} and S_{01} symbols are defined as $S\left(t\right)$. The random noise $N\left(t\right)$ is generated in the range $[-1:1]$ and scaled by a noise index n_{i}. The sequence recovered by the attacker is therefore $S_N\left(t\right) = S\left(t\right) + N\left(t\right) \cdot n_{\mathrm{i}}$

Figure 3 shows some results for my scheme: I calculate the probability of the attacker making a bit error and plot this against the noise index n_i for varying amplitude differences m. For this case I also assumed the best case for the attacker in terms of environmental noise, so there is no additional $N'(t)$. A bit error rate of 50% is equivalent to the attacker randomly guessing the key bits, as statistically he should get half of his guesses correct. The final bit-error probability for each (n_{i}, m) pair is the average bit-error probability of 100 trials, each containing 100 S_{01} and S_{10} symbols.

Apart from the amplitude of the additive noise and the amplitude difference between S_{01} and S_{10}, the frequency of the additive noise and the environmental noise $N'\left(t\right)$ can also influence the scheme's success. The effect of varying the frequency of the generated noise is minimal and, if anything, an increase in frequency decrease bit error probability, as shown in Figure 4(b). As is probably expected, any environmental noise that is added to the signal observed increases the probability that the attacker will make a bit error. An example of the environmental noise's effect is shown in Figure 4(a).

IV. Conclusion

I propose a method for making near-field communication resistant to eavesdropping where the reader transmits a noisy carrier to obfuscate the backward channel. I show that current bit-blocking schemes used to obfuscate RFID data are vulnerable because attackers can distinguish between the blocking sequence and the data based on the difference in the modulation amplitude of the blocker and the device. I improve on these proposals by randomizing the physical communication characteristics with an additional layer of AWGN. This makes it difficult for attackers to distinguish between the blocking sequence and data, even if the devices differ in terms of phase and modulation depth. I show simulated results suggesting that this method significantly increases the probability that an attacker will make a bit-error when attempting to recover the data. Apart from creating an eavesdropping resistant channel the scheme could possible be used for key exchange between devices with limited cryptographic resources. It can also be use by RFID blocker and proxy systems to hide any differences in their communication medium compared to the tokens they



(a) Effect of additional environmental noise, $N'(t)$, with noise index equal to 0.3

(b) Effect of the generated noise's frequency with noise index equal to 0.3. The noise multiplier is applied to the frequency of the data

FIGURE 4 - EFFECT OF NOISE FREQUENCY AND ENVIRONMENTAL NOISE ON THE NOISY CARRIER SCHEME

guard.

In my proposal the reader itself acts as the blocker, which simplifies the system as the user does not need to carry a special blocking device. The reader transmits a noisy carrier onto which the token modulates its data. Implementing the scheme requires little additional hardware in the reader and it is transparent to the token, so it can be extended to any inductively coupled communication, e.g., ISO 14443A/B and ISO 15693. It can also be extended to any system using 'passive' NFC technology and can therefore be applied to ubiquitous computing applications, where pairing and key-exchange often happen between devices that have never interacted before.

REFERENCES

[1] G.P. Hancke. *Practical attacks on proximity identification systems (short paper)*. Proceedings of IEEE Symposium on Security and Privacy, pp 328-333, May 2006.

[2] A. Juels, D. Molnar and D. Wagner. *Security and Privacy Issues in E-passports*, Proceedings IEEE/CreateNet SecureComm, pp 74 – 88, 2005.

[3] A. Juels, R. Rivest and M. Szydlo. *The Blocker Tag: Selective Blocking of RFID tags for consumer privacy*. Proceedings of Conference on Computer and Communications Security (CCS), pp 103–111, October 2003.

[4] M.R. Rieback, G.N. Gaydadjiev, B. Crispo, R.F.H. Hofman and A.S. Tanenbaum. *A Platform for RFID Security and Privacy Administration*. 20th USENIX/SAGE Large Installation System Administration conference (LISA 2006), December 2006.

[5] C. Castelluccia and G. Avoine. *Noisy Tags: Pretty Good Key Exchange Protocol for RFID Tags*. Proceedings of International Conference on Smart Card Research and Advanced Applications (CARDIS), LNCS Vol. 3928, pp 289–299, April 2006.

[6] E. Haselsteiner and K. Breitfuss. *Security in Near Field Communication*. Proceedings of Workshop on RFID Security, pp 3–13, July 2006.

[7] C.E. Shannon. *A Mathematical Theory of Communications*. Bell Systems Technical Journal, Vol. 27, pp 623–656, 1948.

[8] A.D. Wyner. *The Wire-Tap Channel*. Bell Systems Technical Journal, Vol. 54, pp 1355–1387, October 1975.

[9] H. Chabanne and G. Fumaroli. *Noisy Cryptographic Protocols for Low-Cost RFID Tags*. IEEE Transactions on Information Theory, Vol 52, No 8, August 2006

[10] J. Bringer and H. Chabanne. *On the Wiretap Channel Induced by Noisy Tags*. Proceedings of European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS), pp 113–120, 2006.

[11] ISO 14443. *Identification cards – Contactless integrated circuit cards – Proximity cards*.

[12] C. Castellucia and P.Mutaf. *Shake Them Up*, Proceedings of Mobile Systems, Applications and Services (Mobisys), pp 51–64, June 2005.

[13] 'Simple Pairing', Bluetooth White Paper, August, 2006. `http://www.bluetooth.com/Bluetooth/Apply/Technology/Research/Simple_Pairing.htm`

[14] ISO 18092 (ECMA-340). *Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)*.

[15] K. Finkenzeller, *RFID Handbook: Radio-frequency identification fundamentals and applications*, Wiley, 1999.

[16] J.G. Proakis. *Digital Communications*, 3rd Edition, McGraw-Hill, 1995.