Attacks on Time-of-Flight Distance Bounding Channels

Gerhard P. Hancke
ISG Smart Card Centre
Royal Holloway, University of London
Egham TW20 0EX, UK
ghancke@ieee.org

Markus G. Kuhn
Computer Laboratory, University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
http://www.cl.cam.ac.uk/~mgk25/

ABSTRACT

Cryptographic distance-bounding protocols verify the proximity of two parties by timing a challenge-response exchange. Such protocols rely on the underlying communication channel for accurate and fraud-resistant round-trip-time measurements, therefore the channel's exact timing properties and low-level implementation details become security critical. We practically implement 'late-commit' attacks, against two commercial radio receivers used in RFID and sensor networks, that exploit the latency in the modulation and decoding stages. These allow the attacker to extend the distance to the verifier by several kilometers. We also discuss how 'overclocking' a receiver can make a prover respond early. We practically implement this attack against an ISO 14443A RFID token and manage to get a response 10 µs earlier than normal. We conclude that conventional RF channels can be problematic for secure distance-bounding implementations and discuss the merits and weaknesses of special distance-bounding channels that have been proposed for RFID applications.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*authentication, physical security*; C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based Systems—*real-time and embedded systems*; C.2 [**Computer Systems Organization**]: Computer-communication networks

General Terms

Security

Keywords

Distance-bounding protocols, location-based authentication, data modulation, wireless communication, radio channels, round-trip time measurement, low-latency communication, speed of light, RFID

WiSec'08, March 31–April 2, 2008, Alexandria, Virginia, USA.

1. INTRODUCTION

Physical location can provide a measure of trust in security applications. In some systems, users are granted privileges or services based on their proximity or location. Verifying the location of a device, through authentication protocols, is therefore an important security mechanism. Whereas location services provide absolute location of devices within a network, distance-bounding protocols only aim to prove the proximity of two devices relative to each other. Distance bounding only involves two parties, a prover and a verifier, and allows the verifier to place an upper bound on the physical distance to the prover, without assistance from a third party.

Brands and Chaum first proposed a distance-bounding protocol that could be used to verify a device's proximity cryptographically [3]. Their design relied already on a channel where the prover can reply instantaneously to each single binary digit received from the verifier. Since then, other such protocols have been proposed, to prevent relay attacks in proximity identification tokens [5] and to prevent wormhole attacks in sensor networks [9,10]. These are only a few examples of distance-bounding protocol proposals and numerous more exist, not only in the one-to-one proximity identification context but also as building blocks for secure location systems [8]. All these protocols contain a phase where the round-trip time of a cryptographic challenge-response exchange between the verifier and the prover is measured and used to estimate the distance between the two participating parties.

The security of such time-of-flight distance-bounding protocols depends not only on the cryptographic protocol itself, but also on the practical implementation and the physical attributes of the communication channel. It must ensure that attackers cannot manipulate the transmission and reception time of individual bits at the physical layer. Therefore, it is crucial that the design of such a protocol is carefully integrated with the underlying physical layer of the communication channel. The channel must not introduce any latency that an attacker could bypass with alternate implementations of transmitters and receivers or by using a different medium. For example, ultrasound is not a good medium since the propagation speed is much slower than that of radio waves, leaving the system vulnerable to relay attacks. Such low-level implementation details are easily forgotten in a design, despite their importance.

In this paper we discuss the suitability of conventional radio channels, as used in sensor nodes and RFID systems, for distance-bounding implementations. We discuss the RF re-

ceiver structures often used in these systems and investigate how an attacker can circumvent round-trip-time measurements by exploiting latency in the demodulation and decoding stages. Practical examples of late-commit and overclocking attacks are also presented. Finally, we look at existing proposals for communication channels specifically designed for distance bounding and comment on their effectiveness.

2. ATTACKS AT THE PHYSICAL LAYER

Distance bounding protocols require accurate timing in order to estimate the round-trip time (RTT) of challenge-response pairs. Consider a simple system where the verifier starts a timer when it has sent a challenge bit and stops the timer when it detects the start of a response bit sent by the prover. If all system components have a predictable time delay, the verifier now has a good RTT estimate and can therefore calculate an upper bound on the distance to the prover. Accurate timing alone does not, however, ensure that the protocol is secure, as the actual response value still needs to be determined. If an attacker could start a response within the allowable time period but still change the value at a later stage, once he knows the correct response, the protocol's security would be compromised. The verifier must, therefore, ensure that the prover commits to the response value at a well-defined point of time, which effectively links the time measurement with the cryptographic exchange.

We introduced the idea of 'Deferred Bit Signaling' in [1], an attack on receivers that integrate the signal amplitude over an entire bit period. The attacker could send no energy for the initial $\frac{m-1}{m}$ of the time interval and then send an m-times stronger-than-normal signal during the final $\frac{1}{m}$ of the time interval reserved for the bit. The result of the receiver's integration would be the same, but the attacker can delay committing to a bit's value by $\frac{m-1}{m}$ of the bit period. This notion, that an attacker can change a bit's value after its transmission time has begun, was the starting point for the work presented in this section. We build on this idea and show that late-commit attacks can be implemented in a number of ways, by exploiting different features of the receiver's decoding and demodulation architecture.

The late-commit attack can be used in both *distance fraud* (where the prover is fraudulent) and *relay attacks* (where a third party interferes with the channel). A fraudulent prover can commit distance fraud by preemptively guessing a response and then, if required, changing it to the correct value once the challenge is received. In relay attacks the attacker cannot avoid introducing a delay when relaying the challenges and responses, which could cause the round-trip-time to exceed the limit set by the verifier. The attacker can use the late-commit attack to guess the prover's response and then, if needed, change his guess once he receives the actual response from the prover. In this case an attacker would implement a special receiver that determines the response of the prover early in the bit period, which still gives him time to alter his response. This technique can also be used to 'shorten' the time taken to relay the challenge from the verifier to the prover. An example of this type of relay attack is shown in Figure 1.

2.1 The Communication Channel

A typical communication channel consists of a transmitter sending data to a receiver using an RF carrier. Data sent over this channel must first be encoded and then modulated onto the carrier. Coding changes the binary data into a base-band signal suitable for the transmission channel and aids the receiver in recovering the data, e.g. Non-Return-to-Zero (NRZ), Manchester, etc. Modulation is the process by which the amplitude, frequency or phase of an RF carrier is altered in relation to this baseband signal. The receiver must then perform demodulation and decoding to recover the data, as shown in Figure 2.

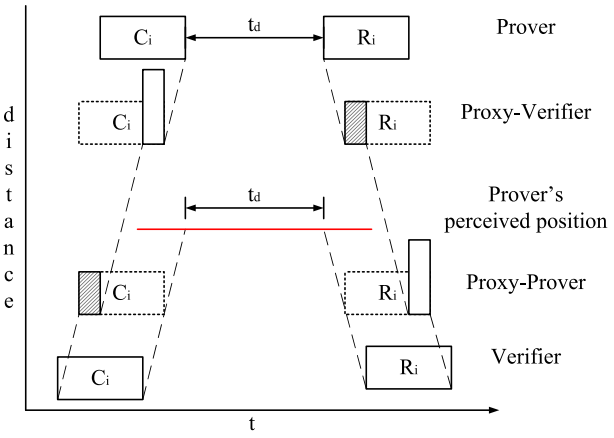


Figure 1: In this variation of the relay attack, the attacker gains time when the proxy prover estimates the value of the challenge bit from the verifier early on in the bit period, and the proxy verifier transmits m times the symbol amplitude to the prover in the final $\frac{1}{m}$-th of the bit period. The process is then repeated for the response bit, albeit with the proxy verifier and prover swapping roles.

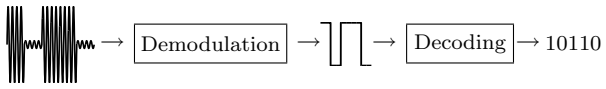


Figure 2: Data recovery at the receiver

2.1.1 Super-Heterodyne Receivers

Sensor nodes generally use RF carriers in the ISM bands (315 MHz, 433 MHz and 2.4 GHz) with simple modulation schemes such as amplitude shift keying (ASK) or frequency shift keying (FSK). For example, the popular Mica2 node by Crossbow Technology [11] uses FSK at 315/433 MHz. Most such RF receivers, including the Chipcon CC1000 [14] on the Mica2 node, use the super-heterodyne architecture shown in Figure 3.

The incoming carrier is mixed down, using a synthesized clock from a phase-locked loop (PLL), to an intermediate frequency band (IF) where it is filtered and amplified. The IF stage often contains a limiter, to prevent saturation in the remaining receiver circuitry. The coded data is demodulated off the IF carrier and quantized by a data slicer. For ASK, the IF carrier is rectified and passed through a low-pass filter (envelope detector), while the carrier is fed into another PLL for FSK. The data slicer is usually a comparator with a dynamic reference. This stage may include additional low-pass filters to remove high frequency glitches. Some receivers also

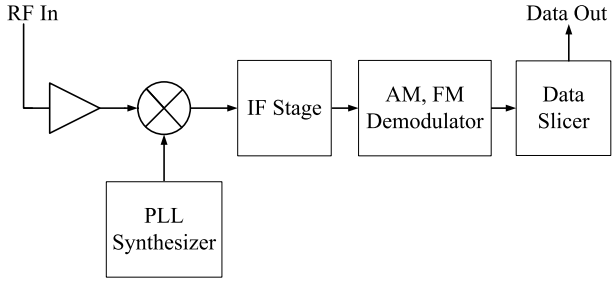


Figure 3: Functional diagram of a generic superheterodyne RF receiver

do decoding, but most of the time this function is performed by another logic device, such as a micro-controller or FPGA connected to the receiver. For the practical work in this paper, we used the MAXIM 1471 433.92 MHz ASK/FSK receiver evaluation board [15] and the RF Solutions RRFQ2 433.92 MHz FM receiver [12].

2.1.2 RFID Receivers

RFID tokens used for proximity authentication use a HF carrier with a two-stage modulation process. The coded data is first modulated onto a low-frequency (847 kHz or 423 kHz) sub-carrier before being amplitude modulated onto the 13.56 MHz carrier. RFID receivers use an architecture similar to the one shown in Figure 4. First the 13.56 MHz carrier is demodulated. This can be done by rectifying the carrier and passing the result through an envelope detector. Alternatively, a zero-IF system could be implemented, where the received signal is mixed with a 13.56 MHz clock. The signal is then low-pass filtered to leave only the modulated sub-carrier, which is then amplified, demodulated and digitized.

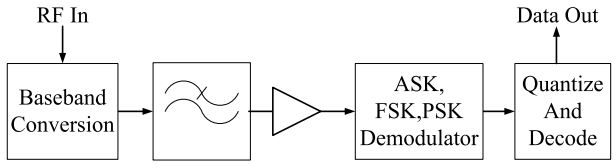


Figure 4: Functional diagram of a generic 13.56 MHz RFID receiver

For our experiments, we used the NXP MF RC531 contactless reader IC [20]. This receiver uses an IQ demodulator to recover the coded data from the sub-carrier and also performs decoding. During the demodulation process, the received signal is correlated with an expected base function, which produces a peak in the output whenever the signal corresponds closely to the base function. A general correlation receiver with N correlators projects the received signal $r(t)$ onto N basis functions $f_k(t)$ [21, pp 233–244]:

$$y_k = \int_0^T r(t) f_k(t)\,\mathrm{d}t,\ k = 1, 2, \ldots, N$$

Here, $[0, T]$ is the time interval of one symbol, each of which can represent several bits. In the simple case of rectangular-shaped pulses representing individual bits, as used by the shift-keying receiver examined here, we have only a single basis function, which is rectangular, and the correlator just integrates the time interval $[0, T_\mathrm{B}]$ assigned to each bit:

$$y = \int_0^{T_\mathrm{B}} r(t)\,\mathrm{d}t$$

So in our test case, the correlator is just an integrator, and this receiver architecture is ideal for testing the 'Deferred Bit Signaling' attack. For comparison we also studied the Melexis MLX90121 13.56 MHz RFID transceiver [16]. This receiver's data sheet does not describe how it performs the required sub-carrier demodulation, but it does contain an additional 'majority voting' step to help filter noise and correct distorted signals during digitization. An example of a majority voting scheme is shown in Figure 5. Multiple samples are taken over the bit period, and at the end, a decision is made as to whether the signal should be high or low. This is, in effect, similar to integrating over the bit period, or averaging, and comparing to a threshold, except that here binary threshold decisions are made both before and after adding up all samples taken for one bit.

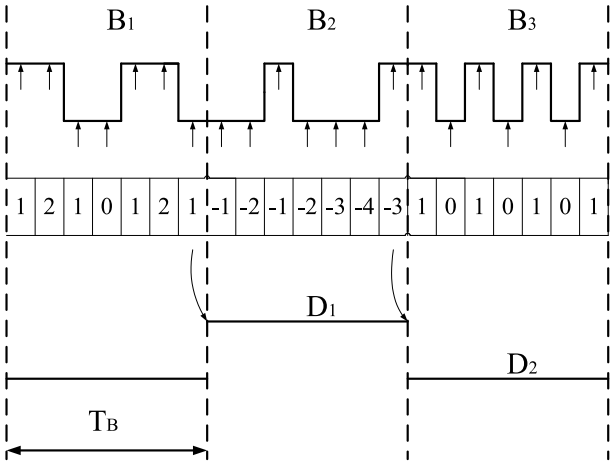


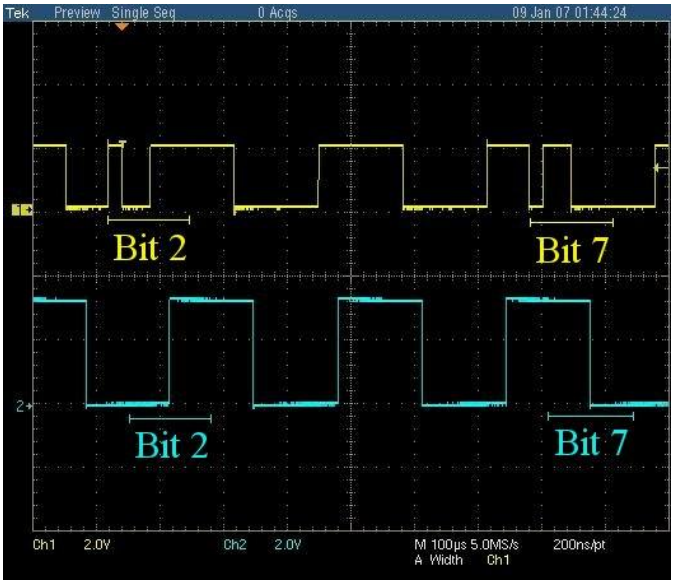
Figure 5: Example of majority voting over a bit period. Here the receiver samples the incoming signal $B_{1,2,3}$ seven times during each bit period T_B. It then decides, based on the vote, whether to make the corresponding output $D_{1,2}$ high or low.

2.2 Late-Commit Attacks

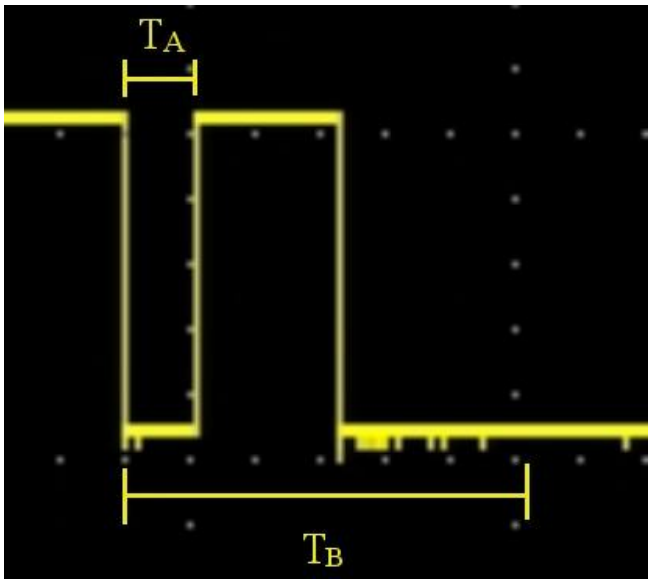
We implemented 'late-commit' attacks against two RF chips that use the above receiver and decoder structures. In both cases, it was possible to start transmitting a response and then change the value of this response later during T_B. We used two different experimental setups to implement the attack. We then discuss optimized attack strategies against these and several other commonly used decoding techniques, as well as counter measures and clock attacks.

2.2.1 Example 1: UHF Data Transceiver

In the case of the Maxim 1471 ASK/FSK receiver, we connected an RF signal generator (HP E4421B) directly to the antenna input and provided a 433.92 MHz carrier suitably modulated with a data stream generated by an FPGA board. Our aim was to exploit the low-pass data filters in the AM demodulator and data slicer to implement a 'late-commit' attack. These filters are designed to both remove unwanted demodulation products, as well as glitches with



(a) Bit 2: '1'→'0', Bit 7: '0'→'1'



(b) Bit 7: $T_A \approx 22\ \mu s \triangleq 6.6$ km (at speed of light)

Figure 6: Late-commit attack exploiting the receiver's data filter.

a shorter duration than the bit period T_B. This, however, means that if the attacker changes his response after a short time period (T_A), his initial incorrect response is filtered out and the receiver outputs the expected data stream.

Figure 6 shows an example implementation of this attack against the Maxim 1471 receiver. The top waveform, in (a), shows the irregular data stream and the bottom waveform shows the output of the receiver. In the example, a Manchester encoded data stream (10101010) is transmitted and bits 2 and 7 are changed to illustrate the attack. In (b), a magnified trace of bit 7 showing T_A and T_B is shown. The attacker assumes that the next bit will be the inverse of the current bit and starts to transmit the relevant data. He then finds that either he is correct, in which case he keeps with his current value, or that he is wrong and he changes the value accordingly.

In our case an attacker can still commit to the right value T_A = 20–22 µs after he started the response, and his incorrect attempt will be filtered out by the receiver. The demodulated data will be the same as if he guessed the response correctly from the start and the receiver output will be as expected by the verifier.

2.2.2 Example 2: ISO 14443 Reader

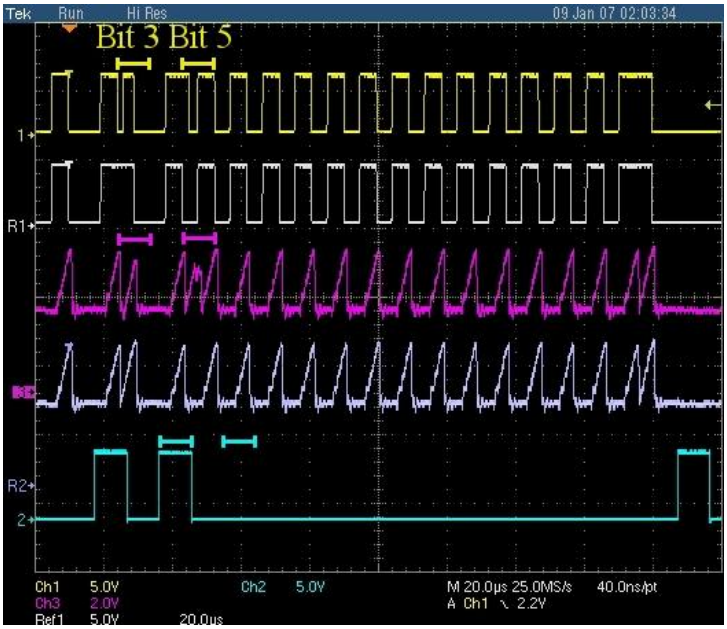
In the case of the NXP MF RC531 contactless reader IC, we used an ISO 14443A compliant test PICC (proximity integrated circuit card), described in [19], to load modulate the 13.56 MHz carrier with a data stream from the FPGA board. The data steam was formatted according to ISO 14443A, i.e. 106 kbit/s Manchester coded data. The NXP MF RC531 has several debugging outputs that allowed us to observe the signal waveforms at different stages in the receiver.

This time, our aim was to exploit the integrator to not having to commit to a bit value at the start of the bit period. We start to respond with some arbitrary value and then change it once the attacker knows the correct one. This only must happen soon enough to ensure that the correlation result ends up on the correct side of the decision threshold. Figure 7 shows an example implementation of this approach where the PICC answers with an ATQA (Answer to Request: Type A) in response to a REQA (Request: Type A) command from the reader. The top trace, in (a), shows the irregular data stream, the second trace shows the correct response, the third trace shows the output of the correlation stage for the irregular input, the fourth trace shows correlation stage output for the correct input and the bottom trace shows the output of the receiver measured for both inputs. In (b), a magnified trace of bit 3 and 5 showing T_A, T_B, the corresponding correlation output and the threshold is shown. During bit 3 the attacker initially guesses low but then changes to high, still ensuring that the correlation peak is large enough. In bit 5 the attacker guesses high but then changes his answer to low before the correlation peak reaches the threshold. In this case, an attacker can gain almost a quarter of the bit period, approximately 2–2.5 µs.

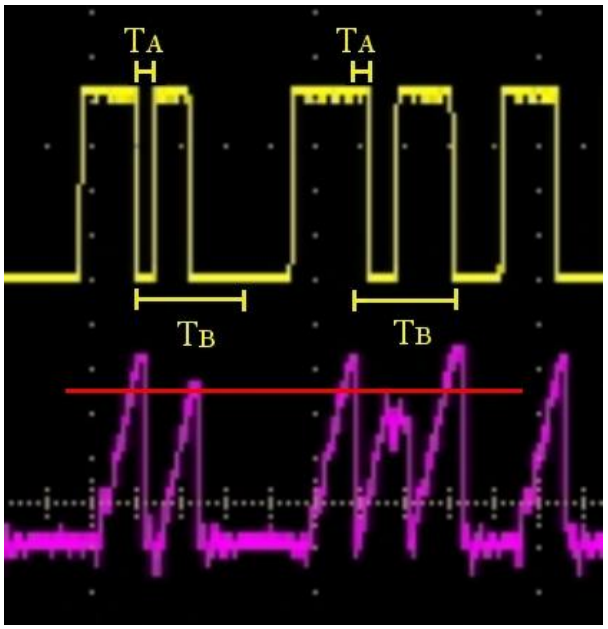
2.2.3 Strategies

Which signal value an attacker should best transmit initially, before deciding on a bit value, depends on which range of signal values the attacker can achieve at the input of the integrator. If the attacker were able to achieve arbitrarily large positive and negative input voltages there, then the initial voltage would not matter much, as the integration result could be changed in any direction by large values at the last moment.

At the output of an ideal linear AM demodulator, an attacker could achieve arbitrarily large positive voltages, but no negative voltage (due to the rectification in an envelope detector). In this case, it would be prudent for the attacker to start out with zero voltage, to keep the accumulating value in the integrator low for as long as possible, because it will be easy to increase the output of the integrator later,



(a) Bit 3: '0'→'1', Bit 5: '1'→'0'



(b) $T_A \approx 2.5$ µs $\triangleq$ 750 m (at speed of light). The red line shows the level of the decision threshold.

Figure 7: Late-commit attack exploiting the correlator in an RFID receiver

but there will be no way to reduce it. This is the scenario that motivated the initially described attack, in which a '1'-bit is represented by the lowest possible base-band voltage for $\frac{m-1}{m}$ of the bit period and by a voltage m-times the one normally used for a '1'-bit during the final $\frac{1}{m}$-th part of the period T_B.

In practice, the output of a linear or logarithmic demodulator might be limited to voltages in the range 0 to V_m, with the binary threshold for the integration result set at $T_\mathrm{B} \cdot V_\mathrm{m}/2$. Such limits may either have been introduced intentionally, to reduce the ability of brief large voltage spikes to interfere with the decoding process, or they may just be an unintended side effect of amplifier and supply-voltage limitations. In either case, the optimal late-commit strategy for an attacker facing two voltage limits 0 and V_m is to initially aim at a demodulator output voltage of $V_\mathrm{m}/2$, and then to switch to 0 or V_m when the desired bit value is known. This keeps the integrator heading for exactly the threshold level, ensuring that even a very brief voltage-limited deviation at the last moment can still steer it to either side, thereby maximising the attacker's timing advantage T_A.

If a constant integrator input voltage that would lead to an integration result identical to the threshold value is not achievable, e.g. because of non-linearities such as threshold elements between the demodulator output and integrator input, then alternating between the voltages corresponding to '0' and '1' during the undecided period might be used to achieve the same effect.

2.2.4 Other Decoder Algorithms

During the decoding stage, a device needs to decide if a '1' or a '0' was transmitted during a particular bit period, T_B. Using an integrator to determine the average input voltage during the bit duration, followed by applying a threshold, is only one of several commonly implemented decoding techniques. Others involve simply sampling at carefully chosen points in time the output of a simpler low-pass filter that crudely approximates the function of the integrator. To prevent bit errors due to clock jitter, these sampling times usually incorporate a safety margin to ensure that the receiver samples not too close to the boundaries between bit periods.

Consider the following popular methods to decode NRZ and Manchester signals. In each case, the device requires a locally generated clock to periodically sample the incoming signal. In the case of NRZ, a common method is to sample once, preferably at $\frac{1}{2}T_B$ after the start of the bit period, during each bit period and assign a '1' to a high, and '0' to a low input state. For Manchester coding, the device might take a sample S_1 at $\frac{1}{4}T_B$ and a sample S_2 at $\frac{3}{4}T_B$. The result S_1 = high and S_2 = low would decode to '1', S_1 = low and S_2 = high would decode to '0', and any other combination would be invalid.

In some cases, devices take more than one sample per bit period and determine the value by a majority voting scheme similar to the one in Figure 5. For example, the USART module of a PIC16F876 micro-controller will sample three times during each bit period and make a decision on the number of those samples corresponding to high or low [17].

An attacker could try to late-commit by exploiting the conservative sampling times during all these decoding processes. The exact attack method and time gain, T_A, would depend on the specific decoding method and filter time-constants. For example, when the channel uses NRZ encoding and the verifier samples each bit once, the attacker needs to apply the correct bit value only one or two filter time-constants before $\frac{1}{2}T_\mathrm{B}$. If the device samples earlier and more often during the bit period and uses a majority voting scheme, this does not make the situation any worse for the

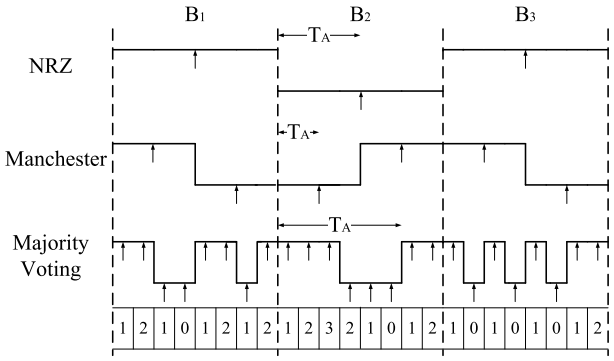


Figure 8: Examples of 'late-commit' attacks on the decoding stage.

attacker. He can then send balanced data and use the last few samples to move the counter to the desired side of the threshold. Figure 8 shows example attacks on NRZ, Manchester and a majority voting scheme, and indicates in each case the time T_A gained by the attacker before he needs to commit to a value.

To exploit such decoding steps, an attacker needs to send pulses shorter than T_B. In the NRZ and Manchester examples, the demodulator and any subsequent smoothing filter need to allow through pulse of somewhat less than half a bit period length, i.e. for the attack to work, the receiver must allow frequency components higher than the intended data frequency to pass. In some cases, nodes communicate with each other using a receiver that can easily transmit higher frequency data, e.g. using a 100 kbit/s receiver to receive 9.6 kbit/s serial data. To test this, we kept the received signal strength constant while increasing the data frequency. We found that the receivers tested reliably demodulate data above the data filter cut-off frequency. At -100 dBm the Maxim and RF Solutions FSK receivers managed to demodulate 15 kbit/s NRZ data even though their low-pass data filters have cut-off frequencies of 9.6 and 4.8 kHz respectively.

2.3 Countermeasures

The underlying vulnerability of all attacks presented so far is present as long as the bit duration T_B is substantially longer than the time that light needs to travel twice the distance that marks an acceptable accuracy for a distance-bounding scheme. In the case of a 9.6 kbit/s channel used on a sensor node, this half-bit length is more than 15 km, in the case of a 100 kbit/s RFID channel it is still 1.5 km.

So the obvious countermeasure is to adhere to the four principles for secure time-of-flight distance-bounding in [1], in particular Principle 3, which states that T_B should be as short as possible. Nevertheless, it may not always be practical for reasons of cost and compatibility to make these modifications to the transmitter and receiver hardware.

It may, however, be feasible in some circumstances to modify a decoding process only during the execution of a distance-bounding protocol. This particularly applies to the case, where a software routine decides on the exact time of sampling the output of a demodulation filter, or even applies a majority vote to several such samples. In this case, the software only has to be modified to sample the values considered in the bit value decision, as early as possible in the bit period, to the extent allowed by the filter's time constant. This, however, means that only a fraction of the energy normally transmitted for each bit will be utilized to distinguish it from background noise, and as a result, this approach will lead to higher bit error rates. This approach also requires accurate timing and synchronization between the transmitter and receiver, since the idea behind voting or sampling in the middle of the period was to allow for differences and drift in their respective clocks. While the resulting increase in bit error rate may not be tolerable for regular data transmission purposes, it may still be more than sufficient for use with a distance-bounding protocol that was especially designed for use on highly unreliable channels, such as [5, 22].

Other approaches involve using tighter decision thresholds, in particular the use of separate thresholds for '0' and '1' bits. These would equally increase the bit-error rate, and might also be less practical in situations where an existing RF chip with a built-in single-level comparator has to be used.

2.4 Clocking Attacks

In addition to the late-commit attacks we also consider the possibility that the attacker could speed up the reply from the prover, as previously mentioned in [1]. This attack is especially relevant for protocols that expect the prover to first receive an entire multi-bit challenge before replying. Getting the correct response earlier than expected could allow the attacker enough time to relay the response back to the verifier.

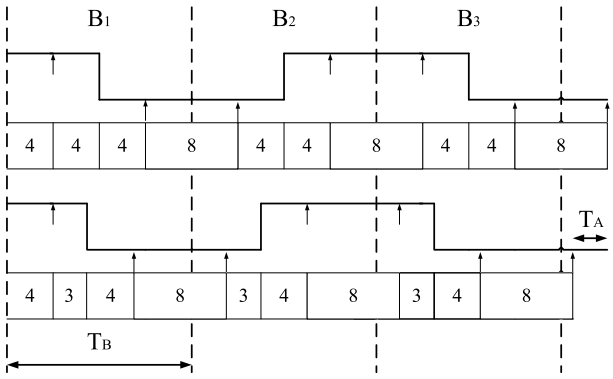


Figure 9: Inducing an early response by influencing the receiver's sampling clock. The numbered blocks indicate the value of the counter when it was reset.

This attack assumes that the sampling, or data, clock is not generated independently by the receiver but recovered from the encoded data. Since we could not obtain a suitable IC implementing such an operation, we devised our proof-of-concept attacks against a Manchester decoder, using a PLL or counter clock recovery method, as described in [18]. Since the attacker controls the transmission of the encoded data to the prover, he can alter the sequence in such a way as to increase the frequency of the data clock, which would cause the prover to decode the data in a shorter time and reply early. Exploiting the PLL is theoretically easier since the attacker just needs to transmit the data at a higher data rate, taking care to stay within the limits of the PLL. The PLL will now synthesize a faster data clock, which is the desired outcome. Speeding up the data clock when the

receiver uses the counter method requires slightly more effort but it is still feasible. The counter method described requires that the receiver generates a 16× sampling clock, which is then synchronized with the Manchester encoded data. After receiving the first transition, the counter limit is set to 4. Once the counter reaches 4, the receiver samples and sets the counter to 0 and the counter limit to 8. Once the counter reaches 8, the receiver samples and again sets the counter to 0 and the counter limit to 8. It continues in this state until an edge transition occurs, at which time the entire process starts over, i.e. counter limit set to 4 and counter set to 0. By shifting the edge transitions forward the attacker resets the counter early, causing the receiver to sample earlier. As a result the attacker speeds up the sampling process and gains time T_{A} over the entire data sequence, as shown in Figure 9.

3. DISTANCE-BOUNDING CHANNELS

Conventional communication channels are designed for reliable data transfer. Channels feature redundancy and timing tolerances to prevent bit errors, but this also introduces latency for an attacker to exploit. Systems planning to use distance-bounding protocols should, therefore, implement special low-latency channels. Published proposals for the implementation of distance-bounding channels over radio channels are currently confined to the HF RFID environment.

There are two proposals for a distance-bounding channel where the verifier directly samples the modulated carrier. This means that the verifier could determine the prover's response without performing traditional demodulation and decoding, thus reducing communication channel latency. Both proposals are tailored to the HF RFID environment and depend on the load modulation process, which allows the token to amplitude modulate the carrier transmitted by the reader by changing its impedance.

In the proposal described by Munilla, et al. [4] the reader transmits a periodic sequence of pulses that are 100% ASK modulated onto the carrier. The pulses act as synchronization bits, with the periods in between, when there is no carrier present, being called slots. In some slots the reader will switch on the carrier for a short period of time to indicate that it wants a response. The token knows when to expect these requests and preemptively switches its impedance to indicate the answer. When the reader then switches on the carrier, the envelope of the signal rises immediately to a level that indicates the token's answer state. As soon as the envelope finishes rising and is stable, the reader checks whether load modulation is on or off. The time it takes until two levels can be distinguished, and the difference between the envelope amplitude for the two states, depends on the distance between the token and the reader. The authors state that the timing resolution of the channel is less than 1 µs.

Since the token knows when the reader will issue a challenge, and is in fact expected to respond preemptively, this implementation does not prevent distance fraud. Another problem arising from the token preemptively setting its reply state is that a proxy-reader could generate a weak carrier, that will not be interpreted as 'on' by the token, to read out the answer early. An additional practical drawback is that the carrier is switched off regularly, which means that the token has no source of power for long periods of time.

The proposal by Reid, et al. [2] assumes that the token will reply after a fixed time. Practically the token waits for a pre-determined number of cycles of the 13.56 MHz carrier, which in theory would synchronize its response to an accuracy in the order of $\frac{1}{13.56\ \mathrm{MHz}} \approx 75$ ns. The reader times from the end of its command to the moment that the response is detected. The time at which the response is received is measured using a special detector that tries to determine the exact moment that the amplitude of the carrier is first modulated. This involves sampling the peaks of the HF carrier and comparing the latest sample to a threshold calculated from the eight previous samples. The resolution of the system is once again dependent on the distance between the token and the reader, with a 300 ns resolution obtained when the token and the reader were 4–5 cm apart.

The authors state their assumption that the token is protected against overclocking and that the RF carrier operates within the ±7 kHz tolerance specified by the relevant standard. However, this requirement does not seem to be enforced by tokens currently available.

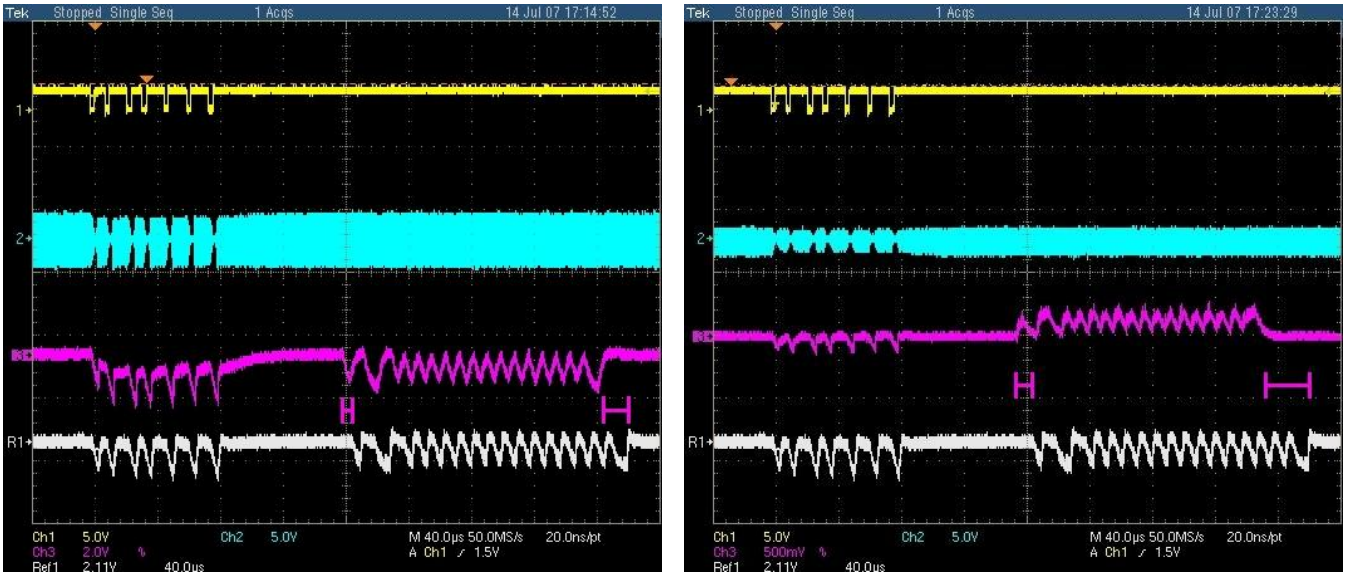
Figure 10 shows the effect on the token's response if the frequency of the HF carrier is increased by 1 MHz and 2 MHz, respectively. The first trace shows the *REQA* request sent by the reader, with the second trace showing the corresponding carrier modulation. The third trace shows the token's *ATQA* response, the sequence on the right of the picture, when the carrier frequency is increased. The fourth trace shows a reference token's response when the carrier is 13.56 MHz. In each case the response was recovered using a tuned pick-up coil held close to the token. In Figure 10(b) the recovered data is slightly distorted as the operating frequency moves away from the coil's tuned frequency. The accelerated carrier increases the transmitted bitrate resulting in a knock-on effect on the subsequent bits, so the final bit is time shifted much more compared to the time shift of the first bit.

In Section 2.3, we mentioned that T_{A} can be decreased by reducing T_{B}, even if that compromises the reliability of the channel. In [5], we followed this principle when suggesting the use of a crude ultra-wideband channel for distance-bounding RFID tokens. The verifier and the prover use the 13.56 MHz carrier for loose synchronization. At a pre-agreed time, the verifier transmits a single challenge bit, to which the prover replies with a single response bit generated by an asynchronous circuit, which eliminates overclocking attacks. The short duration of the bits reduces latency and makes it difficult for the attack to shorten the bit even more using one of the attacks in the previous section. This channel is not meant to be reliable in terms of data transfer and therefore it can only implement protocols that allow for a substantial bit-error rate during the fast exchange phase.

A similar method was recently used to implement distance bounding for contact smart cards [6]. This technique has yet to be practically implemented on resource-limited devices in an RFID environment.

4. CONCLUSION

We showed that it is possible to implement 'late-commit' attacks against popular receiver architectures. We describe practical attack implementations on two receiver architectures and also discuss further attack strategies that an attacker could implement to exploit timing latency in the demodulation and decoding stages of receivers. We also show how an attacker can speed up the reply of the prover by in-



(a) +1 MHz: First edge ≈ 5 µs, last edge ≈ 15 µs

(b) +2 MHz: First edge ≈ 10 µs, last edge ≈ 30 µs

Figure 10: Time gained from 'overclocking' a 13.56 MHz contactless token

fluencing the receiver's recovered data clock or, in the case of RFID tokens, the system clock of the prover.

The communication channel vulnerabilities presented in this paper undermine the security of distance-bounding protocols, so we must conclude that conventional receiver architectures are not well suited for distance bounding implementations. Distance-bounding should rather be implemented using a specially designed channel. Current proposals for such channels in the RFID environment show some promise but are not yet perfect, with each of the two 'rise-time' measuring schemes [2, 4] exhibiting weaknesses. The other proposal requires an additional wideband pulse channel to be implemented, which appears to be feasible, but it needs specially designed tokens. Further research on implementing communication links that will support distance bounding protocols is needed.

5. REFERENCES

[1] J. Clulow, G.P. Hancke, M.G. Kuhn, T. Moore. *So Near and Yet So Far: Distance-Bounding Attacks in Wireless Networks.* Proceedings of European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS), pp 83–97, 2006.

[2] J. Reid, J.M.G. Nieto, T. Tang and B. Senadji. *Detecting Relay Attacks with Timing-Based Protocols.* Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, pp 204–213, March 2007.

[3] S. Brands and D. Chaum. *Distance-bounding protocols.* Advances in Cryptology EUROCRYPT '93, Springer-Verlag LNCS 765, pp 344–359, May 1993.

[4] J. Munilla, A. Ortiz and A. Peinado. *Distance Bounding Protocols with Void Challenges for RFID.* Proceedings of Workshop on RFID Security (RFIDSec), pp 15–26, July, 2006.

[5] G.P. Hancke and M.G. Kuhn. *An RFID distance bounding protocol.* Proceedings of IEEE SecureComm, pp 67–73, 2005.

[6] S. Drimer and S.J. Murdoch. *Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks.* In Proceedings of USENIX Security, September 2007.

[7] K.B. Rasmussen and S. Čapkun. *Implications of Radio Fingerprinting on the Security of Sensor Networks.* In Proceedings of IEEE SecureComm, 2007.

[8] N. Sastry, U. Shankar and D. Wagner. *Secure verification of location claims.* Proceedings of the 2003 ACM Workshop on Wireless Security, pp 1–10, September 2003.

[9] Y.C. Hu, A. Perrig and D.B. Johnson. *Packet leashes: A defense against wormhole attacks in wireless networks.* Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 3, pp 1976–1986, April 2003.

[10] S. Čapkun, L. Buttyán and J. Hubaux. *SECTOR: secure tracking of node encounter in multi-hop wireless networks*, Proceedings ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN), ACM Press, 2003.

[11] Mica2 node, 2006. Crossbow Technology, `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf`.

[12] RF Solutions FM Transmitter and Receiver Modules `http://www.rfsolutions.co.uk/acatalog/DS069-7.pdf`

[13] NXP MF RC531 Contactless Reader IC `http://www.nxp.com/products/identification/mifare/index.html`

[14] ChipCon CC1000 Single Chip Very Low Power RF Transceiver `www.chipcon.com/files/CC1000_Data_Sheet_2_2.pdf`

[15] MAXIM-IC 1471 315MHz/434MHz Low-Power, 3V/5V ASK/FSK Superheterodyne Receiver
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/4304

[16] MELEXIS MLX90121 13.56MHz RFID Transceiver
http://www.melexis.com/ProdMain.aspx?nID=78

[17] Microchip 16F87X Datasheet
ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf

[18] Xilinx, Inc. *Manchester Encoder-Decoder for Xilinx CPLDs*. Application Note XAPP339 (v1.3), October, 2002.

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

[20] NXP Semiconductors. *Contactless Reader Components – Data Sheets and Application Notes*.
www.nxp.com/products/identification/readers/contactless/

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

[22] D. Singelée, B. Preneel. *Distance Bounding in Noisy Environments*. European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS), Springer-Verlag LNCS 4572, pp 101–115, 2007.