# An RFID Distance Bounding Protocol

Gerhard P. Hancke, Markus G. Kuhn

*University of Cambridge, Computer Laboratory*
*15 JJ Thomson Avenue, Cambridge CB3 0FD, UK*
{*gh275,mgk25*}*@cl.cam.ac.uk*

## Abstract

*Radio-frequency identification tokens, such as contactless smartcards, are vulnerable to relay attacks if they are used for proximity authentication. Attackers can circumvent the limited range of the radio channel using transponders that forward exchanged signals over larger distances. Cryptographic distance-bounding protocols that measure accurately the round-trip delay of the radio signal provide a possible countermeasure. They infer an upper bound for the distance between the reader and the token from the fact that no information can propagate faster than at the speed of light. We propose a new distance-bounding protocol based on ultra-wideband pulse communication. Aimed at being implementable using only simple, asynchronous, low-power hardware in the token, it is particularly well suited for use in passive low-cost tokens, noisy environments and high-speed applications.*

## 1. Introduction

Pervasive computing systems aim to provide services specific to the user's context or location. Users who successfully spoof their location could gain access to services to which they are not entitled. Verifying the location of a mobile device, through the use of secure protocols, has therefore become an important function of wireless networks [1]. Secure distance-bounding protocols are intended to enhance traditional authentication mechanisms [2] and can provide additional assurance, such as a metric for secure routing in ad-hoc networks [3].

RFID devices or contactless smartcards are often used to link a user with a location [4] or a context for proximity authentication [5]. Passive RFID devices operate without an internal battery and receive the power they need to operate from an electromagnetic high-frequency field generated by the reader. This offers a long lifetime, but results in short read ranges and requires a high-powered reader. Contactless interfaces have been standardized for "proximity" (ISO 14443 [6]), "vicinity" (ISO 15693 [7]) and "near field" (ISO 18092 [8]) devices, with nominal operating ranges in the order of 10 cm to 1 m [9]. These standards specify the operating frequency, modulation and coding schemes, anti-collision routines and communication protocols.

RFID tokens, especially those implementing cryptographic authentication over an ISO 14443 link, are widely deployed today in ticketing and building access-control applications. However, these are susceptible to relay attacks. An attacker can use two transponders in order to relay over a larger distance the information that a reader and a token exchange during a cryptographic challenge-response protocol. A proxy-token device is placed near the real reader and a proxy-reader device is placed near the real token, possibly unknown to its holder. The proxy reader powers up the token, the proxy token establishes contact with the reader, and then both proxies forward any data received. As a result, the reader will report that it has verified the presence of a token that is actually far away [10].

Relay attacks cannot easily be prevented by cryptographic protocols that operate at the application layer of an RFID protocol stack. At this layer, information about the arrival times of messages has already been blurred substantially by the many synchronization, collision-avoidance, demodulation, symbol-detection, error-detection and retransmission mechanisms that are implemented in the lower layers. The only effective defense are distance-bounding or secure-positioning protocols that are tightly integrated into the physical layer of the communication protocol, so as to obtain high-resolution timing information about the arrival of individual data bits. Only with sub-microsecond timing information can a cryptographic protocol rely directly on the laws of physics that postulate that no information can propagate through space-time faster than light (at 0.3 m/ns).

## 2. Background

A variety of distance measurement and positioning concepts that were originally developed for navigation purposes have more recently been applied to wireless networking and

Proceedings of IEEE/Create-Net SecureComm 2005, 5–9 September 2005, Athens, Greece, pp. 67–73. IEEE Computer Society Press, Los Alamitos, California, USA, ISBN 0-7695-2369-2.

context-aware computing services. Most of these use radio-frequency signals, which propagate well and are already the established medium for mobile communication.

In broadcast positioning systems, signals flow only in one direction, towards the receivers that determine either their own position or that of the transmitter (GPS being a prominent example). In round-trip systems, signals flow in both directions. The distance between two stations is then calculated as

$$d = c \cdot \frac{t_{\mathrm{m}} - t_{\mathrm{d}}}{2} \quad (1)$$

$$t_{\mathrm{m}} = 2 \cdot t_{\mathrm{p}} + t_{\mathrm{d}} \quad (2)$$

where $c$ is the propagation speed, $t_{\mathrm{p}}$ is the one-way propagation time, $t_{\mathrm{m}}$ is the measured total round-trip time and $t_{\mathrm{d}}$ is the processing delay at the remote device. Time-of-Arrival (TOA) and Difference-in-Time-of-Arrival (DTOA) concepts use the propagation delay to calculate distance, while Angle-of-Arrival (AOA) concepts determine the incoming direction of signals [11, pp 193–219]. All these methods use triangulation, with data from several base stations, to obtain 2D or 3D positions. Such systems have been demonstrated in indoor environments [12] [13], but require hardware capable of high sampling rates and complex DSP operations.

Ultra-wideband (UWB) communication systems require precise synchronization between transmitter and receiver. The resulting shared time base can also be used for distance or position measurements, with resolutions of 30 cm or less [14]. The large bandwidth of such systems makes them more resilient to errors from multi-path effects and provides for finer ranging resolution. Lower-power and smaller components are expected to become available for ranging systems [15].

Many other technologies have been suggested for location applications. Received-Signal-Strength (RSS) systems have been demonstrated that can estimate location with typical errors as small as 1.5 m, by processing signal-strength information from multiple base stations [16] [17]. In [18], an RFID authentication scheme is proposed where the level of trust is related to RSS. However, RSS methods are vulnerable to attackers who can change the broadcast power or directional characteristics of devices to spoof locations.

Ultrasound has been used in positioning applications such as the Active Bat system [19]. Sound travels six orders of magnitude slower than light, so greater spatial resolution can be obtained with simpler hardware. Sound-based systems are not suited for distance-bounding applications, as an attacker can always use faster radio waves to link two transponders. In the Echo system [20], the verifier and prover communicate using both RF and ultrasound. Radio frequency is used for transmitting a challenge nonce which is then sent back using ultrasound. (This particular protocol is vulnerable when a proxy is placed close to the verifier, because echoing a nonce does not require any secret information from the real prover.)

Brands and Chaum [21] described the first distance-bounding protocol based on timing the single-bit round-trip delay in a cryptographic challenge-response exchange. Both the verifier and the prover first generate random bitstrings $C = C_1C_2 \ldots C_n$ and $R = R_1R_2 \ldots R_n$, respectively. The verifier then transmits one challenge bit $C_i$ at a time (for all $i = 1, \ldots, n$), to which the prover responds immediately with $R_i$. The verifier times the round-trip delay between sending each bit $C_i$ and receiving the corresponding response bit $R_i$. After all $n$ bits have been exchanged, the prover completes the protocol by transmitting a message authentication code (or digital signature) for the two bitstrings $C$ and $R$. With this final message, the prover not only provides a value that is cryptographically derived from the challenge nonce $C$, thereby confirming to the verifier that the real prover was indeed involved in the protocol. With it, the prover also confirms to the verifier that it has indeed received each $C_i$ *before* sending out its corresponding nonce bit $R_i$. As a result, a proxy verifier who requests from the prover the $R_i$ replies prematurely, by supplying it with guessed challenges $C'_i$, will succeed in doing so only with probability $2^{-n}$ without being detected (i.e., with $C' = C$).

For applications where the verifier does not trust the prover to wait with the transmission of $R_i$ until it received $C_i$, Brands and Chaum describe a protocol variant in which the prover first commits to a new random bitstring $M = M_1M_2 \ldots M_n$ (e.g., by transmitting a secure hash value $h(M)$). The reply bits $R_i = C_i \oplus M_i$ are then calculated by XOR-ing each challenge bit with the corresponding bit of $M$. Finally, the prover reveals $M$ and signs $C$ and $R$ (or equivalently any two of $C$, $M$, $R$). The commitment on $M$ prevents the prover from sending some random bit $R_i$ early and then setting $M_i = R_i \oplus C_i$ after receiving $C_i$. However, neither variant stops the real prover from colluding with a proxy prover who is located closer to the verifier.

Location-based authentication services that measure the round-trip time of entire data packets have been proposed [22]. Using a distance-bounding protocol based on single-bit round trips as a building block can improve their security and spatial resolution.

A technique for securing a broadcast positioning system against location spoofing was recently proposed by Kuhn [23]. Unlike deployed military GPS techniques, it does not rely on long-term shared secrets.

## 3. Distance Bounding Protocol

Our proposal is based on the following assumptions about the environment in which the RFID system operates:

- **Security target**: The purpose of our protocol is only to prove to the verifier that the authentication token (prover) is located not more than a specified distance from the verifier. The protocol will not help to prove this fact to any third party, in other words, it does not provide non-repudiation of location for anyone who does not trust the verifier. The protocol also assumes that the prover does not collude with a third party that is located closer to the verifier, in order to pretend to be at a closer distance from the verifier. However, it does not assume that the prover will not violate the protocol on its own (without a colluder) to appear closer than it really is.

- **Cryptographic primitives**: For the purpose of running a distance-bounding protocol, the prover and the verifier share a dedicated secret pseudorandom function (or in practice a dedicated shared secret key and a keyed public pseudorandom function). It is used to calculate the prover's response to a challenge. We assume that the attacker has no access to the shared key or function other than through the radio interface. Our token does not depend on any public-key primitives, but should these be available, then they can be used to set up the shared key mentioned above before the distance-bounding protocol is initiated.

- **Time base**: Our RFID device is computationally weak. It can compute the secret pseudorandom function mentioned above, but the time it takes for this computation (e.g., several milliseconds) is many orders of magnitude longer than the maximum response-delay variance acceptable for our distance-bounding application (tens of nanoseconds). Even worse, the cryptographic calculation progresses according to an externally supplied, and therefore untrusted, clock signal, which a proxy reader might accelerate in the hope of getting a faster response from the token. We assume that the RFID device has no built-in high-precision time base, such as a crystal oscillator. But we do assume that it is reliably able to detect large deviations from its nominal clock frequency, in particular any attempt by an attacker to operate the RFID device at at least *twice* its normal speed (overclocking attack). A simple analog band-pass filter applied to the clock signal can act as a crude trusted time reference, able to prevent a factor-two deviation of the clock frequency. In fact, the carefully tuned magnetic loop antennas used in many existing RFID systems, where the carrier frequency is the clock signal, already are such band-pass filters.

The protocol could be extended into a secure positioning service by running a distance-bounding protocol with multiple verifiers. The resulting confirmed location region would then lie within the intersection of the content of several spheres around these verifiers.

### 3.1. Description

At the start of our protocol, the verifier $V$ (an RFID reader) sends to the prover $P$ (an RFID token) a nonce $N_V$, an unpredictable bitstring that will never again be used for the same purpose:

$$V \rightarrow P: \quad N_V$$

Both the prover and the verifier then use the pseudorandom function $h$ and the secret key $K$ in order to calculate two $n$-bit sequences $R^0$ and $R^1$:

$$R^0_1 R^0_2 R^0_3 \ldots R^0_n \,||\, R^1_1 R^1_2 R^1_3 \ldots R^1_n \;:=\; h(K, N_V)$$

Then, a predefined number of clock cycles after the transmission of $N_V$, begins a sequence of $n$ single-bit challenge-response exchanges. The verifier $V$ generates and sends an unpredictable random challenge bit $C_i$, and the prover $P$ replies instantly with a 1-bit response that is either $R^0_i$ or $R^1_i$, selected by the value of $C_i$. It discards the respective other value securely at the same time. For all $1 \leq i \leq n$:

$$\begin{aligned} V \rightarrow P: &\quad C_i &&\in \{0,1\} \\ P \rightarrow V: &\quad R^{C_i}_i &&\in \{0,1\} \end{aligned}$$

If the correct response $R^{C_i}_i$ is received within a sufficiently short time $t_{\mathrm{m}}$ after $C_i$ had been sent out, for each $1 \leq i \leq n$, then using equation (1) the verifier is satisfied that the prover is not a distance larger than $d$ away.

The cryptographic function $h$ can be calculated by the prover entirely before the time-critical challenge-response phase begins. The prover will only ever reveal half of all the bits that it derived from the nonce $N_V$ and the key $K$. An attacker could slightly accelerate the clock signal provided to the prover and transmit an anticipated challenge $C'_i$ before the verifier reveals its challenge $C_i$. In half of all cases, the attacker will have guessed the challenge bit correctly, that is $C'_i = C_i$, and therefore will have obtained in advance the correct value $R^{C_i}_i$ that is needed to satisfy the verifier. In the other half of all cases, where $C'_i \neq C_i$, the attacker will have irrevocably destroyed the correct answer $R^{C_i}_i$. In that case, the attacker can reply with a guessed bit, which will be correct in half of all cases. Therefore, for each challenge $C_i$, the attacker has only a $\frac{3}{4}$ probability of replying correctly. Overall, the attacker has only a $(\frac{3}{4})^n$ probability of answering all $n$ challenges correctly.

An attacker could try to retrieve all bits $R^0_i$ and $R^1_i$ in advance by attempting to run the protocol with the prover twice (within the time allowed by the verifier for a single

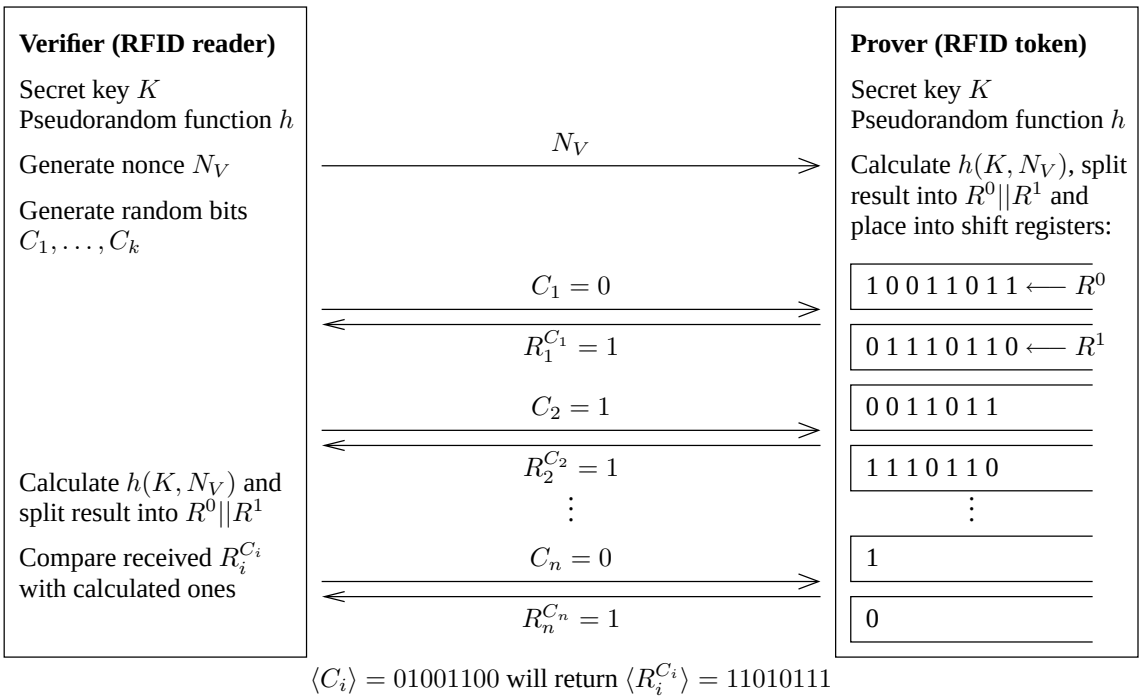


**Figure 1. The challenge-response scheme used in the presented distance-bounding protocol consists of two phases. The first phase is not time-critical and calculates (typically in software) a response $R$ to a challenge $N_V$, using a pseudo-random function $h$ and a shared secret key $S$ known to both parties. The $2n$ bits of $R$ are not returned directly. Instead, they are split up and loaded into two $n$-bit shift registers. A preagreed fixed number of clock cycles after the transmission of $N_V$, the time-critical second phase begins, in which additional single-bit challenges $C_i$ are transmitted. Each selects one of the two shift registers, which returns its first bit directly, using fast asynchronous logic that does not wait on any clock cycle. The first bit in the respective other shift register is discarded at the same time. This way, only half of all response bits $R$ that were generated for an $N_V$ are revealed.**

run). In both protocol runs, the attacker would forward the same nonce $N_V$, but the values $C_i'$ in the second run would be complementary to those in the first run. We assumed that this is not possible because the prover has access to a crude trusted time reference that keeps it from running at twice the normal clock frequency. Where this assumption is not practical, the protocol can be modified by adding a prover-generated nonce. The protocol then starts with both sides transmitting to each other their nonce

$$\begin{aligned} V \to P: &\quad N_V \\ P \to V: &\quad N_P \end{aligned}$$

(in any order) and then continues to generate $R^0$ and $R^1$ from $h(K, N_V, N_P)$. Neither $N_V$ nor $N_P$ need to be unpredictable now to the attacker. They merely must be bitstrings that are guaranteed to never repeat during the lifetime of the verifier or prover. In practice, such nonces can either be sufficiently long random-bit sequences or they can be strictly monotonic counter values or timestamps. An implementor can chose between the need for including into the prover one of either a clock-frequency limiter, some non-volatile memory, a hardware random-bit generator, or a continuously running clock.

### 3.2. Practical Implementation

Formulating the response $R_i^{C_i}$ based on the received $C_i$ is a simple single-bit lookup in a 2-bit memory, which can be implemented in an entirely asynchronous fashion, requiring only a small number of gate delays, without any clock signals that the attacker could accelerate to obtain $R_i^{C_i}$ prematurely.

The function $h$ may in a typical implementation be some form of secure (i.e., one-way and collision-resistant) hash function. The lengths of $K$, $N_V$, $N_P$, as well as $n$, are all security parameters. Typical values should be comparable to the lengths acceptable for symmetric-cryptography keys (80 to 256 bits). If much higher values of $n$ are needed,

for example to cope with transmission errors, $R^0||R^1$ can be generated using a secure pseudo random-bit generator seeded from the output of $h$.

In a practical hardware implementation, $R^0$ and $R^1$ could be loaded into two shift registers, which are both clocked well before another $C_i$ is received. Instead of using shift registers, the next pair $(R_i^0, R_i^1)$ can also be computed by iterating a pseudo random-bit generator before $C_i$ is received. As the signal propagation time $t_\mathrm{p}$ is very small, it is important that the processing delay $t_\mathrm{d}$ of the token is short and predictable. From equation 1, we can see that variations in $t_\mathrm{d}$ greatly effect $d$, especially if $t_\mathrm{p} < t_\mathrm{d}$. The verifier allows the token a preagreed number of clock cycles to calculate $h(K, N_V)$ and store the result. This effectively separates the processing delay from the distance-bounding process, as $t_\mathrm{d}$ is reduced to the time it takes an asynchronous digital circuit to lookup and transmit one bit.

### 3.3. Radio Channel Considerations

For this protocol to be implemented, we need a radio frequency communication link with high bandwidth. The distance resolution of a communication channel with bandwidth $B$ is roughly

$$r = \frac{c}{B}. \tag{3}$$

The carrier frequency and communication bandwidth in typical existing RFID systems is not adequate for localization. For example, in the ISO 14443 A standard, the carrier has a frequency of 13.56 MHz and its data bandwidth of only ≈300 kHz corresponds to a distance resolution in the order of a kilometer. Therefore, we require a different communication technology to obtain a useful distance bound. UWB communication concepts seem to be an attractive option, with bandwidths large enough to provide resolution down to centimeters.

The basic idea is that both $C_i$ and $R_i^{C_i}$ are transmitted on a wideband channel as short pulses generated by a single signal edge applied to an antenna. The token requires a time base to be able to predict when the $C_i$ pulse will arrive, and then it simply samples with a fast sample-and-hold stage the value received. The time-base can be defined by the same narrowband carrier wave that is used to power the token and carry the non-distance-bounding parts of the supported protocols, including the transmission of the nonce $N_V$. Our protocol only requires a short sequence of bit pulses to be sampled. Of the established UWB puls modulation schemes, Bi-Phase Modulation (BPM), where bits are represented by pulses of opposite polarity, is far better suited for a distance-bounding application than Pulse Position Modulation (PPM), where bits are represented by pulses transmitted with or without delay. With BPM, all energy related to a single bit is released at the same time, whereas the potentially delayed pulse in PPM would only add additional round-trip timing uncertainty.

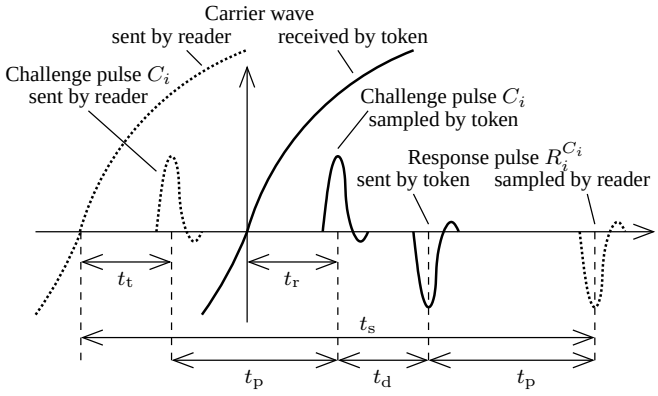


**Figure 2. The power-supply carrier wave emitted by the reader establishes a common time base for synchronizing the pulse communication of both parties. The token samples its wideband input at time $t_\mathrm{r}$ after a zero crossing of the carrier wave, to read a challenge bit $C_i$, and the reader must adjust its transmission delay $t_\mathrm{t} \approx t_\mathrm{r}$ such that its pulse arrives exactly at that time. The token responds with $R_i^{C_i}$ after a short, nearly constant switching delay $t_\mathrm{d}$. The reader must adjust delay $t_\mathrm{s}$ until it receives the correct response, and can then deduce the distance $d = c \cdot (t_\mathrm{s} - t_\mathrm{t} - t_\mathrm{d})/2$.**

We need to synchronize between the transmitter and receiver to recover each impulse reliably. Although the carrier of a 13.56 MHz system does not provide the bandwidth needed for localization, it does provide us with a time-base for synchronizing the communication. One possibility, as shown in Figure 2, is to trigger the token's response circuit through the zero crossing of the carrier. This initiates the readout of the 2-bit memory and leads, after a fixed latency, to a response impulse from the token.

In a very simple implementation, the verifier is equipped with two adjustable delay circuits. The first one (with delay time $t_\mathrm{t}$) is used to time the pulse representing $C_i$ far enough from the zero crossing, such that a token sampling it at around the time of the zero crossing will sample exactly the peak of the pulse. A second delay element (with delay time $t_\mathrm{s}$) in the reader is used to time the moment after the zero-crossing when the reader has the best chance to sample the tip of the incoming $R_i^{C_i}$ pulse. It is then up to the reader to repeat the protocol and try different values for $t_\mathrm{t}$ and $t_\mathrm{s}$ until the answer pulse matches the expected result well. From the settings of the delay circuits, the round-trip time can then be inferred.

In a more sophisticated implementation, the verifier has a

fast single-shot sampling unit, samples for a single response the antenna signals for all delays $t_\mathrm{s}$ that are of interest, and then searches in the recorded results for the lowest value $t_\mathrm{s}$ with an acceptable response. This implementation is faster, as the protocol needs to be repeated only to search for the right delay $t_\mathrm{t}$, at the expense of a faster acquisition logic in the verifier.

By varying in the verifier the delay $t_\mathrm{t}$ during the first few values of $i$ until a $t_\mathrm{t}$ has been found that results in correct response bits, the verifier can adjust itself automatically to any component tolerances and instabilities that may affect the exact sample time in the token. If parameter $n$ is chosen large enough such that after such a delay-element adjustment phase enough bits remain available to satisfy the security of the challenge-response phase, then no repetitions of the protocol will be necessary.

In neither case are variable delay circuits, high clock-frequency circuits, or precise reference frequencies needed in the RFID token. The only RF-powered passive token remains simple; as much circuit complexity as possible is moved into the verifier.

### 3.4. Dealing with Noise

The wide input bandwidth of an RF pulse receiver makes it very sensitive to background noise. In practice, many of the sampled $C_i$ or $R_i^{C_i}$ bits may be corrupted. Therefore, a verifier will have to accept a prover as valid, even if out of $n$ received $R_i^{C_i}$ bits only at least $k$ were correct.

An attacker can guess at least $k$ out of the $n$ response bits $R_i^{C_i}$ right with the false-accept probability

$$p_\mathrm{FA} = \sum_{i=k}^{n} \binom{n}{i} \cdot \left(\frac{3}{4}\right)^i \cdot \left(\frac{1}{4}\right)^{n-i}.$$

On the other hand, if the probability that a received $R_i^{C_i}$ is corrupted by noise is $\epsilon$, then the false-reject probability for a correct token is

$$p_\mathrm{FR} = \sum_{i=0}^{k-1} \binom{n}{i} \cdot (1-\epsilon)^i \cdot \epsilon^{n-i}.$$

The number of transmitted pulses $n$ and the threshold $k$ are security parameters that must be chosen suitably to keep both $p_\mathrm{FA}$ and $p_\mathrm{FR}$ within acceptable margins. This choice is also influenced by $\epsilon$.

## 4. Conclusion

Our protocol provides secure distance bounding for RFID authentication systems. It protects against relay attacks and ensures that a valid token is within an acceptable distance from the verifier. We foresee that this protocol can also be extended to applications other than RFID tokens. Our protocol requires little in terms of power and processing resources from the token, with all time-sensitive adjustments and measurements being done entirely by the verifier. This protocol could therefore be implemented on RFID devices ranging from simple tags to contactless smart cards. The protocol is best used for radio-frequency or optical ranging, which are more suited for security applications than ultrasonic or RSS concepts. We handle communication errors simply by tolerating some bit errors during the single-bit challenge-response exchanges.

Compared to the Brands-Chaum distance bounding protocols [21], our protocol can be implemented with faster authentication time. In a noise-free environment, our protocol requires about twice as many single-bit challenge-response round-trips for the same level of security, because in our protocol an attacker can guess a correct response with probability $\frac{3}{4}$, compared to probability $\frac{1}{2}$ with Brands-Chaum. However, on a noisy channel, typical for UWB communication, this no longer holds. In this environment, the Brands-Chaum protocols would have to be extended to transmit at the end all the actually received $n$ bits of $C$ and the actually transmitted $n$ bits of $R$ to the verifier, such that the message authentication code (MAC) can be verified in spite of bit errors. This additional transmission not only doubles the number of bits needed by the Brands-Chaum protocol, but, along with the MAC, this additional data has to go over a reliable and therefore much slower channel. This substantially increases the time required to complete the full protocol. Furthermore, in contrast to the Brands-Chaum protocol, ours does not require a commitment step to prevent a non-colluding prover from cheating, which reduces even more the number of bits on the reliable channel. In fact, unless the $N_P$ option is chosen, our protocol does not require any noise-free communication channel from the prover to the verifier. This not only simplifies its implementation, but also makes it suitable for applications where a rapid completion of the protocol is required.

Our future work will focus on a practical demonstration of this proposal. We wish to thank the anonymous reviewers and Gildas Avoine for valuable comments.

## References

[1] S. Čapkun and J. Hubaux. *Secure positioning of wireless devices with application to sensor networks*, IEEE INFOCOM 2005. http://lcawww.epfl.ch/capkun/secpos.pdf

[2] L. Bussard and Y. Roudier. *Embedding distance-bounding protocols within intuitive interactions*. Security in Pervasive Computing: First International Conference, Springer-Verlag LNCS 2802, pp 143–156, March 2004.

[3] 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.

[4] I. Satoh. *Location-based services in ubiquitous computing environments*, Service-Oriented Computing – ICSOC 2003, Springer-Verlag LNCS 2910, pp 527–542, November 2003.

[5] J.E. Bardram, R.E. Kjær and M.Ø. Pedersen. *Context-aware user authentication – Supporting proximity-based login in pervasive computing*, UbiComp 2003, LNCS 2864, pp 107–123, Spring-Verlag 2003.

[6] ISO 14443. *Identification cards — Contactless integrated circuit cards — Proximity cards*. International Organization for Standardization, Geneva.

[7] ISO 15693. *Identification cards – Contactless integrated circuit cards — Vicinity cards*. International Organization for Standardization, Geneva.

[8] ISO 18092 (ECMA-340). *Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1)*. Int. Organization for Standardization, Geneva, 2004.

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

[10] G.P. Hancke. *A practical relay attack on ISO 14443 proximity cards*. http://www.cl.cam.ac.uk/˜gh275/relay.pdf

[11] M. Ghavami, L.B. Michael and R. Kohno. *Ultra wideband signals and systems in communication engineering*, Wiley, 2004.

[12] J. Werb and C. Lanzl. *Designing a positioning system for finding things and people indoors*. IEEE Spectrum, Vol. 35, Issue 9, pp 71–78, September 1998.

[13] P. Bahl and V.N. Padmanabhan. *RADAR: an in-building RF-based user location and tracking system*. IEEE Proceedings Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, pp 775–784, March 2000.

[14] R.J. Fontana, E. Richley and J. Barney. *Commercialization of an ultra wideband precision asset location system*. IEEE Conference on Ultra Wideband Systems and Technologies, pp 369–373, November 2003.

[15] R. Zetik, J. Sachs and R. Thome. *UWB localization – active and passive approach*. Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference, Vol. 2, pp 1005–1009, May 2004.

[16] S. Čapkun, M. Srivastava, M. Čagalj and J. Hubaux. *Securing positioning with covert base stations*. NESL, UCLA Technical Report TR-UCLA-NESL-200503-01, 2005. http://lcawww.epfl.ch/capkun/spot/

[17] J. Krumm and E. Horvitz. *LOCADIO: Inferring motion and location from Wi-Fi signal strengths*. First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Mobiquitous, 2004. http://research.microsoft.com/˜horvitz/locadio.pdf

[18] K.P. Fishkin and S. Roy. *Enhancing RFID privacy via antenna energy analysis*, RFID Privacy Workshop, 2003. http://seattleweb.intel-research.net/people/fishkin/pubs.html

[19] A. Harter, A. Hopper, P. Steggles, A. Ward and Paul Webster. *The anatomy of a context-aware application*. Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MOBICOM'99, pp 59–68, August 1999.

[20] 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.

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

[22] B. Walters and E. Felten. *Proving the location of tamper resistant devices*. February 2003. http://www.cs.princeton.edu/˜bwaters/research/

[23] M.G. Kuhn. *An asymmetric security mechanism for navigation signals*. 6th Information Hiding Workshop, May 2004, Springer-Verlag LNCS 3200, pp 239–252.