“Yoking-Proofs” for RFID Tags

Ari Juels
RSA Laboratories
Bedford, MA, USA
ajuels@rsasecurity.com

Abstract

RFID (Radio-Frequency Identification) tags are small, inexpensive microchips capable of transmitting unique identifiers wirelessly over a short distance. Thanks to their utility in automating supply-chain logistics, RFID tags promise eventually to supplant the optical barcode as a means of identifying goods.

We propose the concept of a yoking-proof*, namely a proof that a pair of RFID tags has been scanned simultaneously. Our particular aim is to permit tags to generate a proof that is verifiable off-line by a trusted entity, even when readers are potentially untrusted. We suggest that such proofs are a useful tool for maintaining integrity in supply chains, particularly as RFID data will commonly flow across multiple, loosely affiliated organizations.*

1. Introduction

An RFID (Radio-Frequency Identification) tag is a small microchip, supplemented with an antenna, that is capable of transmitting a unique identifier in response to a query by a reading device. Our focus in this paper is on very basic, passive RFID tags. Such tags do not contain an internal source of power, but instead receive their power from reading devices. Their effective range is on the order of up to several meters.

An industry consortium known as the EPCglobal (formerly the AutoID Center) has promoted a standard tag-data format known as an EPC (Electronic Product Code) [2], along with designs and manufacturing techniques for RFID tags themselves. Thanks to these efforts, estimates suggest that the cost of EPC tags will drop to the vicinity of five cents per unit by 2005 [18]. Ambitious mandates for RFID-tag adoption by large companies such as Wal-Mart and by the United States military [13] suggest that RFID tags – EPC tags in particular – will become commonplace in supply chains in the next few years. The aim of EPCglobal is to see RFID tags supplant barcodes.

The very fact that RFID tags provide rich and commercially beneficial data engenders a number of security and privacy problems [14, 15, 16, 17]. Within supply chains, there is the possibility of corporate espionage through surreptitious scanning of tags or eavesdropping on reading devices (whose transmissions, in contrast to those of tags, are subject to interception at a range of at least hundreds of meters). In the hands of consumers, RFID tags pose a potential privacy threat, as they may permit individuals to be tracked or their possessions to be inventoried without their knowledge, a concern that has received much media attention, as in [11, 19]. A number of papers have sought to address these concerns, by means of enhancements to on-chip protocols [23, 24], through the use of auxiliary privacy-protecting devices [5, 7, 8], or via policy or legislative means [4]. There has also been a great deal of practical deployment of protocols enabling tags to authenticate themselves to reading devices – generally by means of a challenge-response protocol of some kind. See, e.g., [22].

In this paper, we consider a rather different security-oriented problem. Our aim is to enable a pair of RFID tags to generate a proof that they have been scanned simultaneously by a reading device. We refer to this as a *yoking-proof* (applying “yoke” with its meaning “to join together”). To render our proposal practical, we consider yoking-proofs that are verifiable by a trusted party in an offline setting, rather than requiring direct involvement by this party. Here are a couple of brief examples of where such proofs might be useful:

- Pharmaceutical distribution: Suppose that there is a legal requirement for a certain medication to be dispensed together with a leaflet describing its side-effects. One RFID tag might be embedded in the container for the medication, while another is embedded in an accompanying leaflet. (Note that a number of types of RFID tags are designed to be embedded in paper, such as [21].) A yoking-proof would provide evidence that each container of the medication was dispensed with a leaflet. It might be transmitted to, e.g., the U.S. FDA, for verification, or stored by a pharmacist as evidence in case of dispute.

- Manufacturing: Suppose that a manufacturer of aircraft equipment wishes to certify that a certain part always leaves its factories with a safety cap attached. Given RFID tags in both the part and the cap, a yoking-proof can provide third-party-verifiable evidence to support such certification.

It is important to note that our focus is on proofs that tags have been read simultaneously. Our techniques do not in fact provide proof that tags have been read simultaneously by the same reading device, i.e., in physical proximity to one another. They do, however, provide good evidence for this: An adversary would have to create a special linkage between readers (and invest special effort) to effect simultaneous scanning between remote locations.

Resource limitations: The challenge in designing a yoking-proof protocol is that RFID tags have very rudimentary computational abilities. As already explained, they are essentially just small, unpowered microchips. Thus, they can perform only very basic computational operations. Relatively costly (say, fifty-cent) RFID tags are capable of some, limited symmetric-key cryptography. Inexpensive tags – particularly of the common EPC variety, are likely to be unable to execute any kind of traditional cryptographic primitive – even a hash function such as MD5 – for some years to come. RFID tags may contain persistent state, but do not, of course, include clocks. EPC tags of the present generation carry 96-bit identifiers; it is expected that they will contain several hundred bits of storage in the near future [18].

Additionally, basic RFID tags cannot communicate with one another directly. Rather they must rely on the reading device that is querying them as a communications channel. Thus, to be useful, a proof must be verifiably valid even if tags are scanned by an adversarial reader. This accords with the assumption in many cryptographic models for key establishment that the adversary has full control of the communications medium. (See, e.g., [20]. Two-party key agreement is a topic related in a number of ways to our investigation here.)

Assumptions: We assume that an adversary in our proposed system does not reverse-engineer RFID tags. Given their inexpensive nature, RFID tags can offer little in the form of tamper-resistance. It is reasonable to assume, however, that basic tamper-evidence as well as the resources and sophistication required to reverse-engineer tags will act as an obstacle to such attacks in most environments. Of course, where high levels of protection are needed, suitably robust (and expensive) forms of protection are more suitable. (It is interesting to note that a one-time yoking-proof-protocol of the type described in section 3 is by nature resistant to key recovery via conventional side-channel attacks.)

Our proposal relies on a *timeout* assumption, namely that the protocol will always terminate within a certain interval of time t. In many cases, this is a feature of the basic tag-reading protocol itself. In the UHF part of the spectrum, for instance, frequency hopping by the reader – and thus termination of RFID-tag-reading protocols – is required within 400 ms under FCC regulations in the United States [9]. Even in the case of rogue readers that do not comply with these regulations, however, it is a straightforward and inexpensive matter to impose a rough restriction on the interaction time of an RFID-tag based on, e.g., the rate of capacitor discharge. (Indeed, something of the kind is already implemented by RFID-tag manufacturer Alien to prevent guessing of the PINs used to disable RFID tags.)

Key ideas: We offer two key ideas in this paper. Our first is a simple technique for permitting tags to interleave message authentication codes (MACs) [12] using a reading device as a communications medium. We show how, by maintaining state on tags, it is possible to prevent a reader from corrupting the proof by altering or dropping messages or by "rewinding" the protocol. Under our timeout assumption, then, a successful protocol execution will yield a yoking-proof comprising appropriately MACed messages.

Our second idea is a method for enabling inexpensive and therefore severely resource-constrained RFID tags to compute MACs. Such tags cannot execute cryptographic primitives of the traditional kind, such as HMAC [6]. We therefore propose a severely truncated version of the Lamport digital-signature scheme [10] as an alternative. We refer to this as a *minimalist MAC*. Computing a minimalist MAC simply requires that a tag output a collection of pre-stored, secret bits. As we show, given suitable restrictions on the underlying message space, a minimalist MAC may be made highly secure as a one-time operation.

Organization

In section 2, we present a yoking-proof protocol for tags capable of basic cryptographic operations like MACs and keyed hash functions. We describe our minimalist-MAC scheme in section 3, leading to a

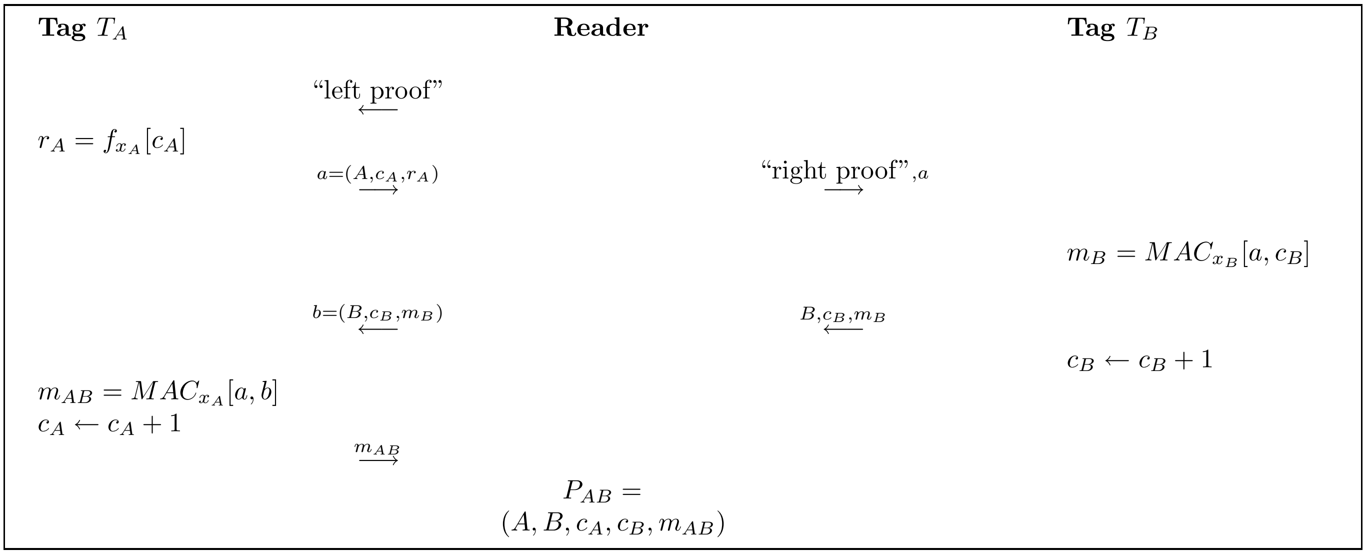
Tag T_A
Reader
Tag T_B
$\xleftarrow{\text{“left proof”}}$
$r_A = f_{x_A}[c_A]$
$\xrightarrow{a=(A,c_A,r_A)}$
$\xrightarrow{\text{“right proof”},a}$
$m_B = MAC_{x_B}[a, c_B]$
$\xleftarrow{b=(B,c_B,m_B)}$
$\xleftarrow{B,c_B,m_B}$
$c_B \leftarrow c_B + 1$
$m_{AB} = MAC_{x_A}[a, b]$
$c_A \leftarrow c_A + 1$
$\xrightarrow{m_{AB}}$
$P_{AB} =$
(A, B, c_A, c_B, m_{AB})

Figure 1. Yoking-proof protocol using standard cryptographic primitives

yoking-proof variant for tags too computationally weak to execute standard cryptographic primitives (e.g., basic EPC tags). We conclude in section 4 with a brief discussion of future research directions.

2. Basic protocol

In this section, we present a yoking-proof protocol that assumes the ability of tags to perform basic cryptographic operations. We denote the full collection of n distinct RFID tags in a system by $T_1, T_2, \ldots, T_n$. We assume that T_i is initialized with a unique, secret, d-bit secret key x_i. The collection of keys $\{x_i\}_{i=1}^n$ is assumed to be known to a trusted verifier, which we denote by $\mathcal{V}$. Additionally, every tag T_i is supplied with a counter c_i, initialized to 0. Let $MAC : \{0,1\}^d \times \{0,1\}^* \rightarrow \{0,1\}^d$ denote a standard cryptographic message authentication code, e.g., HMAC; we let $MAC_x[m]$ denote the MAC computed by applying secret key x to message m. Let $f : \{0,1\}^d \times \{0,1\}^* \rightarrow \{0,1\}^d$ denote a keyed hash function, where for $f_x[c]$, the value x is the key and c is the input value.

Figure 1 specifies our protocol as applied to generate a yoking-proof P_{AB} stating that tags T_A and T_B have been scanned simultaneously. The reader here transmits messages “left proof” and “right proof” to tags to indicate their roles in the protocol. The resulting proof $P_{AB} = (A, B, c_A, c_B, m_{AB})$ may be verified by $\mathcal{V}$ using its knowledge of the secret keys of the tags. In particular, $\mathcal{V}$ computes $a' = (A, c_A, f_{x_A}[c_A])$ and then $b' = (B, c_B, MAC_{x_B}[a', c_B])$, and subsequently checks the equality $P_{AB} = MAC_{x_A}[a', b']$. On timeout or incorrect input, any entity terminates its participation in the protocol.

Security analysis: Space limitations permit only an informal security analysis of the protocol. Assume that any tag completes the full protocol in time at most t, i.e., that once it receives a “left proof” or “right proof” input, it subsequently terminates the protocol once time t has elapsed. We define the security of our protocol in terms of an game as follows. All tags are initialized by $\mathcal{V}$ with their secret values. The adversary $\mathcal{A}$ is then permitted to interact with all tags $\{T_i\}$ for an arbitrarily long period of time and with arbitrary interleaving of queries to tags. (A refinement of this definition might parameterize the total number of inputs that $\mathcal{A}$ makes to tags, but this makes no difference for our proposed construction.) $\mathcal{A}$ then submits a yoking-proof P_{AB} for some pair of tags T_A and T_B to $\mathcal{V}$. We say that $\mathcal{A}$ has read these tags simultaneously if it has provided inputs to each of the two tags within some interval of time of duration t. $\mathcal{A}$ is deemed to have won the game if P_{AB} is accepted as valid by $\mathcal{V}$, but $\mathcal{A}$ has not read T_A and T_B simultaneously. We define the *success probability* δ of $\mathcal{A}$ to be the probability that it is able to win the game. We claim the following, applying the random-oracle model [1] to the underlying cryptographic primitives:

Claim 1: Given random-oracle assumptions on f and MAC, the success probability δ of $\mathcal{A}$ for our proposed

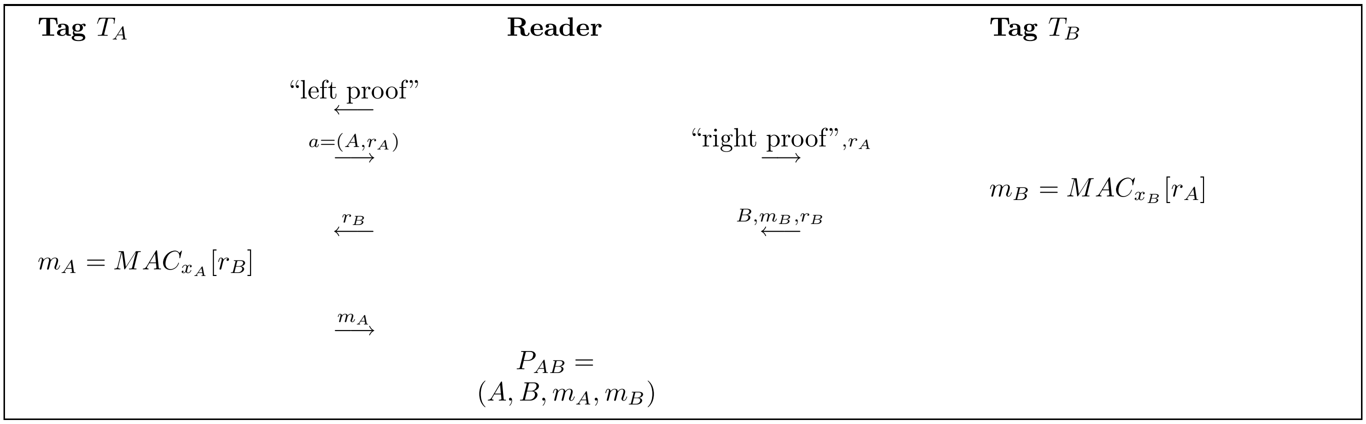

Figure 2. One-time yoking-proof protocol using minimalist MACs

scheme is bounded above by 2^{-d}.

Proof (sketch): Let $P_{AB} = (A, B, c_A, c_B, m_{AB})$. Case 1: Suppose that $\mathcal{A}$ did not submit input "left proof" to T_A on counter value c_A at any time. Then under the random-oracle assumption on f, the adversary $\mathcal{A}$ could only guess $r_A = f_{x_A}[c_A]$ with probability at most 2^{-d}. The random-oracle assumption on MAC then implies that $\mathcal{A}$ could determine the correct value m_B and subsequently the correct value m_{AB} with probability at most 2^{-d}. A similar argument applies for the following cases: Case 2: $\mathcal{A}$ did input "left proof" to T_A on counter value c_A, but did not input $r_A = f_{x_A}[c_A]$ to T_B on counter value c_B and Case 3: $\mathcal{A}$ did not input $b = (B, c_B, m_B)$ to T_A on counter value c_A. If none of these three cases holds, then $\mathcal{A}$ must have scanned T_A and T_B simultaneously. ▮

Note that our protocol also imposes a temporal order on yoking-proofs. In particular, suppose yoking-proofs P_{AB} and P'_{AB} have respective associated counter value pairs (c_A, c_B) and (c'_A, c'_B), where $c'_A > c_A$ and $c'_B > c_B$. Valid proofs demonstrate that P'_{AB} was generated later than P_{AB}. This may have a number of uses. In our aircraft-part-manufacture example above, for instance, it may be desirable for aircraft maintenance crews to demonstrate that a safety cap is in place for every flight: Temporal ordering would prevent reuse of old yoking-proofs.

3. Minimalist MAC

We now consider how to adapt our basic yoking-proof protocol to tags that cannot execute standard cryptographic primitives. Let us first briefly describe our proposed *minimalist* technique for computing one-time MACs (and thus one-time yoking-proofs). This is essentially a stripped-down, symmetric-key version of the Lamport digital-signature construction as described in [10]. In our scheme we assume that the message m to be MACed is exactly d bits in length. A secret key SK is determined as collection of random, l-bit secret values $\{(s_i^{(0)}, s_i^{(1)})\}_{i=1}^d$. This secret key is shared between the signer and verifier. A MAC on message $m = b_1 b_2 \ldots b_d$ simply consists of the collection of secret values $\{s_i^{(b_i)}\}_{i=1}^d$. To forge a MAC in this scheme, an adversary must successfully guess at least one unrevealed value $s_i^{(1-b_i)}$. Given sufficiently a large value l (e.g., $l = 80$), this is infeasible.

To render our MAC scheme space-efficient, however, we propose setting $l = 1$, i.e., making each $s_i^{(b)}$ value consist of a single, random bit. The problem with this approach, of course, is that given a $MAC_x[m]$, an adversary can forge $MAC_x[m']$ on a new message m' quite easily: If m' differs from m in a single bit position, then it suffices for the adversary to guess a single bit to perform the forgery successfully.

What we observe, however, is that if the message space is sufficiently sparse and pairs of messages tend to have relatively large Hamming distances, then forgery is more difficult. By choosing sufficiently large d, we can ensure that the Hamming distance between randomly selected bitstrings is large.

It is possible to do somewhat better, however, by crafting the message space more carefully. Given the space limitations of RFID tags, this is important. In particular, we may select a message space with a good lower bound on the Hamming distance between *any* two messages. This is most easily achieved by defining the message space as the codebook for an error-correcting code. As an example, suppose that we set $d = 120$, and select a message space size of 2^{32} (enough for billions of tags). It is possible, then, to choose a code such that the minimum Hamming distance between any two messages is at least 32 (and probably higher, as this bound is not tight) [3]. In this case, the success probability of an attacker may be bounded above by 2^{-32}, which is probably enough for most practical purposes, given that the attacker has no means of off-line verification. This is particularly the case as our protocol Let M denote a message space computed this way.[1]

In order to restrict the message space in this way, however, we must modify our basic protocol slightly. In the protocol in Figure 1, tag T_A computes a MAC on m_B, itself the output of a MAC, rather than a codeword, as we desire here.

We therefore modify our scheme as follows. We initalize the tag T_A with a one-time random value r_A, and likewise T_B with a one-time random value r_B. The proof protocol is then modified such that T_B releases r_B along with m_B, and T_A computes a MAC on r_B, rather than m_B. Given $d = 120$, then, tags could perform a one-time execution of the protocol given storage of a 120-bit random string and a 240-bit secret key for the MAC, for a total of 360 bits. We give this protocol in Figure 2 above.

We do not offer a security analysis for this protocol, but simply claim that if M is a codebook for a $(d, k, w) - binary$ code, then the probability of successful attack is bounded above by $min(2^{-k}, 2^{-w})$.

4. Conclusions

We have presented techniques that enable a pair of RFID tags to furnish a yoking-proof, i.e., an offline-verifiable proof that they have been read simultaneously.

As EPC tags are expected to contain several hundred bits of storage in the near future, the requirement of 360 bits of storage for our minimalist-MAC protocol is within reason. Nonetheless, it would certainly be desirable to reduce this requirement. An additional important direction of research is the extension of our yoking-proof protocol so as to enable efficient proof that *groups* of tags have been scanned simultaneously.

Acknowledgments

The author would like to thank Burt Kaliski for his very helpful comments on this paper.

References

[1] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[2] D.L. Brock. The electronic product code (EPC): A naming scheme for objects. Technical Report MIT-AUTOID-WH-002, Auto-ID Labs, 2001. Referenced 23 Dec. 2003 at http://www.autoidlabs.com.

[3] A.E. Brower. The linear programming bound for binary linear codes. *IEEE Trans. Inform. Th.*, 39:677–680, 1993.

[4] S. Garfinkel. An RFID Bill of Rights. *Technology Review*, page 35, October 2002.

[5] P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal re-encryption for mixnets. In T. Okamoto, editor, *CT-RSA '04*. Springer-Verlag, 2004. To appear.

[6] H.Krawczyk, M.Bellare, and R.Canetti. HMAC: Keyed-hashing for message authentication, February 1997. Internet Engineering Task Force RFC 2104.

[7] A. Juels and R. Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In R. Wright, editor, *Financial Cryptography '03*, pages 103–121. Springer-Verlag, 2003. LNCS no. 2742.

[8] A. Juels, R.L. Rivest, and M. Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. In V. Atluri, editor, *8th ACM Conference on Computer and Communications Security*, pages 103–111. ACM Press, 2003.

[9] AutoID Labs. 860 MHz-960 Mhz class 1 radio frequency identification tag radio frequency and logical communication interface specification recommended standard, version 1.0.0. Technical Report MIT-AUTOID-WH-007, Auto-ID Labs, 2002. Referenced 23 Dec. 2003 at http://www.autoidlabs.com.

[10] L. Lamport. Constructing digital signatures from a one way function. Technical Report Technical Report CSL-98, SRI International, October 1979.

[11] D. McCullagh. RFID tags: Big Brother in small packages. *CNet*, 13 January 2003. Available at http://news.com.com/2010-1069-980325.html.

[12] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.

[13] U.S. military to issue RFID mandate. 15 Sept. 2003. Referenced 15 Oct. 2003 at http://www.rfidjournal.com/article/articleview/576/1/1/.

[1] By contrast, suppose that M consisted of 2^{32} random 120-bit strings. Then we would expect to see, e.g., a pair of messages in M with Hamming distance less than 32 with very high probability. We conclude this by observing that the standard deviation on a 120-degree binomial random variable is about 5.5. Thus, the probability that a random pair of bitstrings differs in fewer than 32 positions is roughly the cumulative density in the tail of a standard normal distribution at distance 5.1 S.D. from the mean. This is about $1.7 \times 10^{-7} \approx 2^{-22}$. Since the number of distinct pairs among n tags is $\binom{n}{2}$, we would expect to see this low a Hamming distance among just several thousand tags!

[14] S. Sarma, S. A. Weis, and D. Engels. Radio-frequency identification: Risks and challenges. *RSA CryptoBytes*, 6(1), Winter/Spring 2003.

[15] S. E. Sarma, S. A. Weis, and D.W. Engels. Radio-frequency identification systems. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES '02*, pages 454–469. Springer-Verlag, 2002. LNCS no. 2523.

[16] S. E. Sarma, S. A. Weis, and D.W. Engels. RFID systems, security and privacy implications. Technical Report MIT-AUTOID-WH-014, Auto-ID Labs, 2002. Referenced 23 Dec. 2003 at http://www.autoidlabs.com.

[17] S. E. Sarma, S. A. Weis, and D.W. Engels. Radio-frequency-identification security risks and challenges. *CryptoBytes*, 6(1), 2003.

[18] S.E. Sarma. Towards the five-cent tag. Technical Report MIT-AUTOID-WH-006, Auto-ID Labs, 2001. Referenced 23 Dec. 2003 at http://www.autoidlabs.com.

[19] R. Shim. Benetton to track clothing with ID chips. *CNET*, 11 March 2003. URL: http://news.com.com/2100-1019-992131.html.

[20] V. Shoup. On formal models for secure key exchange (version 4), 15 November 1999. Revision of IBM Research Report RZ 3120 (April 1999).

[21] K. Takaragi, M. Usami, R. Imura, R. Itsuki, and T. Satoh. An ultra small individual recognition security chip. *IEEE Micro*, 21(6):43–49, 2001.

[22] Texas Instruments digital signature transponder data sheet. Referenced 15 Oct. 2003 at www.ti.com/tiris/docs/products/transponders/dst.shtml.

[23] S. A. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *First International Conference on Security in Pervasive Computing*, 2003.

[24] S.A. Weis. Radio-frequency identification security and privacy. Master's thesis, M.I.T., June 2003.