@qK

Authenticating Pervasive Devices with Human Protocols

Ari Juels[1] and Stephen A. Weis[2]

[1] RSA Laboratories, Bedford, MA, USA
ajuels@rsasecurity.com
[2] Massachusetts Institute of Technology, Cambridge, MA, USA
sweis@mit.edu

Abstract. Forgery and counterfeiting are emerging as serious security risks in low-cost pervasive computing devices. These devices lack the computational, storage, power, and communication resources necessary for most cryptographic authentication schemes. Surprisingly, low-cost pervasive devices like Radio Frequency Identification (RFID) tags share similar capabilities with another weak computing device: people.
These similarities motivate the adoption of techniques from human-computer security to the pervasive computing setting. This paper analyzes a particular human-to-computer authentication protocol designed by Hopper and Blum (HB), and shows it to be practical for low-cost pervasive devices. We offer an improved, concrete proof of security for the HB protocol against passive adversaries.
This paper also offers a new, augmented version of the HB protocol, named HB^+, that is secure against active adversaries. The HB^+ protocol is a novel, symmetric authentication protocol with a simple, low-cost implementation. We prove the security of the HB^+ protocol against active adversaries based on the hardness of the Learning Parity with Noise (LPN) problem.

Keywords: authentication, HumanAut, Learning Parity with Noise (LPN), pervasive computing, RFID

1 Introduction

As low-cost computing devices become more pervasive, counterfeiting may become a more serious security threat. For example, the security of access control or payment systems will rely on the authenticity of low-cost devices. Yet in many settings, low-cost pervasive devices lack the resources to implement standard cryptographic authentication protocols. Low-cost Radio Frequency Identification (RFID) tags exemplify such resource-constrained devices. Viewing them as possible beneficiaries of our work, we use RFID tags as a basis for our discussions of the issues surrounding low-cost authentication.

Low-cost RFID tags in the form of Electronic Product Codes (EPC) are poised to become the most pervasive device in history [10]. Already, there are billions of RFID tags on the market, used for applications like supply-chain management, inventory monitoring, access control, and payment systems. Proposed as a replacement for the Universal Product Code (UPC) (the barcode found on most consumer items), EPC tags are likely one day to be affixed to everyday consumer products.

Today's generation of basic EPC tags lack the computational resources for strong cryptographic authentication. These tags may only devote hundreds of gates to security operations. Typically, EPC tags will passively harvest power from radio signals emitted by tag readers. This means they have no internal clock, nor can perform any operations independent of a reader.

In principle, standard cryptographic algorithms – asymmetric or symmetric – can support authentication protocols. But implementing an asymmetric cryptosystem like RSA in EPC tags is entirely infeasible. RSA implementations require tens of thousands of gate equivalents. Even the storage for RSA keys would dwarf the memory available on most EPC tags.

Standard symmetric encryption algorithms, like DES or AES, are also too costly for EPC tags. While current EPC tags may have at most 2,000 gate equivalents available for security (and generally much less), common DES implementations require tens of thousands of gates. Although recent light-weight AES implementations require approximately 5,000 gates [11], this is still too expensive for today's EPC tags.

It is easy to brush aside consideration of these resource constraints. One might assume that Moore's Law will eventually enable RFID tags and similar devices to implement standard cryptographic primitives like AES. But there is a countervailing force: Many in the RFID industry believe that pricing pressure and the spread of RFID tags into ever more cost-competitive domains will mean little effective change in tag resources for some time to come, and thus a pressing need for new lightweight primitives.

Contribution. This paper's contribution is a novel, lightweight, symmetric-key authentication protocol that we call HB^+. HB^+ may be appropriate for use in today's generation of EPC tags or other low-cost pervasive devices. We prove the security of this protocol against both passive eavesdroppers and adversaries able to adaptively query legitimate tags. We also offer an improved, concrete security reduction of a prior authentication protocol HB that is based on the same underlying hardness problem.

Organization. In Section 2, we describe the basic "human authentication" or "HumanAut" protocol, due to Hopper and Blum (HB), from which we build an authentication protocol appropriate for RFID tags that is secure against passive eavesdroppers. We discuss the underlying hardness assumption, the "Learning Parity with Noise" (LPN) problem, in Section 3. Section 4 offers our new, enhanced variant of the HB protocol, HB^+, that is secure against adversaries able to query legitimate tags *actively*. Section 5 presents an improved, concrete reduction of the LPN problem to the security of the HB protocol, and shows a concrete reduction of security from the HB protocol to the HB^+ protocol. Finally, Section 6 states several open problems related to this work.

Many details – and our proofs in particular – are omitted from this version of the paper due to lack of space. A more complete version of the paper is available at www.ari-juels.com or crypto.csail.mit.edu/∼sweis.

1.1 The Problem of Authentication

It seems inevitable that many applications will come to rely on basic RFID tags or other low-cost devices as authenticators. For example, the United States Food and Drug Administration (FDA) proposed attaching RFID tags to prescription drug containers in an attempt to combat counterfeiting and theft [13].

Other RFID early-adopters include public transit systems and casinos. Several cities around the world use RFID bus and subway fare cards, and casinos are beginning to deploy RFID-tagged gambling chips and integrated gaming tables. Some people have even had basic RFID tags with static identifiers implanted in their bodies as payment devices or medical-record locators [37].

Most RFID devices today promiscuously broadcast a static identifier with no explicit authentication procedure. This allows an attacker to surreptitiously scan identifying data in what is called a *skimming* attack. Besides the implicit threat to privacy, skimmed data may be used to produce cloned tags, exposing several lines of attack.

For example, in a *swapping* attack, a thief skims valid RFID tags attached to products inside a sealed container. The thief then manufactures cloned tags, seals them inside a decoy container (containing, e.g., fraudulent pharmaceuticals), and swaps the decoy container with the original. Thanks to the ability to clone a tag and prepare the decoy in advance, the thief can execute the physical swap very quickly. In the past, corrupt officials have sought to rig elections by conducting this type of attack against sealed ballot boxes [34].

Clones also create denial-of-service issues. If multiple, valid-looking clones appear in a system like a casino, must they be honored as legitimate? Or must they all be rejected as frauds? Cloned tags could be intentionally designed to corrupt supply-chain databases or to interfere with retail shopping systems. Denial of service is especially critical in RFID-based military logistics systems.

Researchers have recently remonstrated practical cloning attacks against real-world RFID devices. Mandel, Roach, and Winstein demonstrated how to read access control proximity card data from a range of several feet and produce low-cost clones [24] (despite the fact that these particular proximity cards only had a legitimate read range of several inches). A team of researchers from Johns Hopkins University and RSA Laboratories recently elaborated attacks against a cryptographically-enabled RFID transponder that is present in millions of payment and automobile immobilization systems [6]. Their attacks involved extraction of secret keys and simulation of target transponders; they demonstrated an existing risk of automobile theft or payment fraud from compromise of RFID systems.

Example EPC Specifications	
Storage:	128-512 bits of read-only storage.
Memory:	32-128 bits of volatile read-write memory.
Gate Count:	1000-10000 gates.
Security Gate Count Budget:	200-2000 gates.
Operating Frequency:	868-956 MHz (UHF).
Scanning Range:	3 meters.
Performance:	100 read operations per second.
Clock Cycles per Read:	10,000 clock cycles.
Tag Power Source:	Passively powered by Reader via RF signal.
Power Consumption:	10 microwatts.
Features:	Anti-Collision Protocol Support Random Number Generator

Fig. 1. Example specification for a 5-10¢ low-cost RFID tag.

1.2 Previous Work on RFID Security

As explained above, securing RFID tags is challenging because of their limited resources and small physical form. Figure 1 offers specifications that might be realistic for a current EPC tag. Such limited power, storage, and circuitry, make it difficult to implement traditional authentication protocols. This problem has been the topic of a growing body of literature.

A number of proposals for authentication protocols in RFID tags rely on the use of symmetric-key primitives. The authors often resort to a hope for enhanced RFID tag functionality in the future, and do not propose use of any particular primitive. We do not survey this literature in any detail here, but refer the reader to, e.g., [15, 29, 32, 33, 36, 39].

Other authors have sought to enforce privacy or authentication in RFID systems while avoiding the need for implementing standard cryptographic primitives on tags, e.g., [12, 18, 19, 21, 20, 29].

Feldhofer, Dominikus, and Wolkerstorfer [11] propose a low-cost AES implementation, potentially useful for higher-cost RFID tags, but still out of reach for basic tags in the foreseeable future.

1.3 Humans vs. RFID Tags

Low-cost RFID tags and other pervasive devices share many limitations with another weak computing device: human beings. The target cost for a EPC-type RFID tag is in the US$0.05-0.10 (5-10¢) range. The limitations imposed at these costs are approximated in Figure 1. We will see that in many ways, the computational capacities of people are similar.

Like people, tags can neither remember long passwords nor keep long calculations in their working memory. Tags may only be able to store a short secret of perhaps 32-128 bits, and be able to persistently store 128-512 bits overall. A working capacity of 32-128 bits of volatile memory is plausible in a low-cost tag, similar to how most human beings can maintain about seven decimal digits in their immediate memory [28].

Neither tags nor humans can efficiently perform lengthy computations. A basic RFID tag may have a total of anywhere from 1000-10000 gates, with only 200-2000 budgeted specifically for security. (Low-cost tags achieve only the lower range of these figures.) As explained above, performing modular arithmetic over large fields or evaluating standardized cryptographic functions like AES is currently not feasible in a low-cost device or for many human beings.

Tags and people each have comparative advantages and disadvantages. Tags are better at performing logical operations like ANDs, ORs and XORs. Tags are also better at picking random values than people – a key property we rely on for the protocols presented here. However, tag secrets can be completely revealed through physical attacks, such as electron microscope probing [1]. In contrast, physically attacking people tends to yield unreliable results.

Because of their similar sets of capabilities, this paper considers adopting human authentication protocols in low-cost pervasive computing devices. The motivating human-computer authentication protocols we consider were designed to allow a person to log onto an untrusted terminal while someone spies over his/her shoulder, without the use of any scratch paper or computational devices. Clearly, a simple password would be immediately revealed to an eavesdropper.

Such protocols are the subject of Carnegie Mellon University's HumanAut project. Earlier work by Matsumoto and Imai [26] and Matsumoto [25] propose human authentication protocols that are good for a small number of authentications [38]. Naor and Pinkas describe a human authentication scheme based on "visual cryptography" [30]. However, this paper will focus primarily on the the human authentication protocols of Hopper and Blum [16, 17].

2 The HB Protocol

We begin by reviewing Hopper and Blum's secure human authentication protocol [16, 17], which we will refer to as the HB protocol. We then place it in the RFID setting. The HB protocol is only secure against passive eavesdroppers – not active attackers. In Section 4, we augment the HB protocol against active adversaries that may initiate their own tag queries.

Suppose Alice and a computing device C share an k-bit secret $\boldsymbol{x}$, and Alice would like to authenticate herself to C. C selects a random challenge $\boldsymbol{a} \in \{0,1\}^k$ and sends it to Alice. Alice computes the binary inner-product $\boldsymbol{a} \cdot \boldsymbol{x}$, then sends the result back to C. C computes $\boldsymbol{a} \cdot \boldsymbol{x}$, and accepts if Alice's parity bit is correct.

In a single round, someone imitating Alice who does not know the secret $\boldsymbol{x}$ will guess the correct value $\boldsymbol{a} \cdot \boldsymbol{x}$ half the time. By repeating for r rounds, Alice can lower the probability of naïvely guessing the correct parity bits for all r rounds to 2^{-r}.

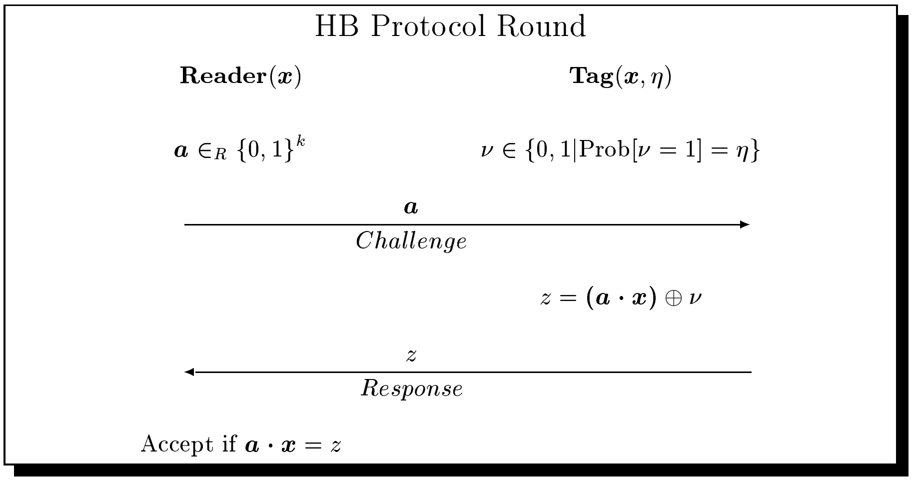

Fig. 2. A single round of the HB authentication protocol.

Of course, an eavesdropper capturing $O(k)$ valid challenge-response pairs between Alice and C can quickly calculate the value of $\boldsymbol{x}$ through Gaussian elimination. To prevent revealing $\boldsymbol{x}$ to passive eavesdroppers, Alice can inject noise into her response. Alice intentionally sends the wrong response with constant probability $\eta \in (0, \frac{1}{2})$. C then authenticates Alice's identity if fewer than ηr of her responses are incorrect.

Figure 2 illustrates a round of the HB protocol in the RFID setting. Here, the tag plays the role of the prover (Alice) and the reader of the authenticating device C. Each authentication consists of r rounds, where r is a security parameter.

The HB protocol is very simple to implement in hardware. Computing the binary inner product $\boldsymbol{a} \cdot \boldsymbol{x}$ only requires bitwise AND and XOR operations that can be computed on the fly as each bit of $\boldsymbol{a}$ is received. There is no need to buffer the entire value $\boldsymbol{a}$. The noise bit ν can be cheaply generated from physical properties like thermal noise, shot noise, diode breakdown noise, metastability, oscillation jitter, or any of a slew of other methods. Only a single random bit value is needed in each round. This can help avoid localized correlation in the random bit stream, as occurs in chaos-based or diode breakdown random number generators.

Remark: The HB protocol can be also deployed as a *privacy-preserving* identification scheme. A reader may initiate queries to a tag without actually knowing whom that tag belongs to. Based on the responses, a reader can check its database of known tag values and see if there are any likely matches. This preserves the privacy of a tag's identity, since an eavesdropper only captures an instance of the LPN problem, which is discussed in the Section 3.

3 Learning Parity in the Presence of Noise

Suppose that an eavesdropper, i.e., a passive adversary, captures q rounds of the HB protocol over several authentications and wishes to impersonate Alice. Consider each challenge $\boldsymbol{a}$ as a row in a matrix $\boldsymbol{A}$; similarly, let us view Alice's set of responses as a vector $\boldsymbol{z}$. Given the challenge set $\boldsymbol{A}$ sent to Alice, a natural attack for the adversary is to try to find a vector $\boldsymbol{x'}$ that is functionally close to Alice's secret $\boldsymbol{x}$. In other words, the adversary might try to compute a $\boldsymbol{x'}$ which, given challenge set $\boldsymbol{A}$ in the HB protocol, yields a set of responses that is close to $\boldsymbol{z}$. (Ideally, the adversary would like to figure out $\boldsymbol{x}$ itself.)

The goal of the adversary in this case is akin to the core problem on which we base our investigations in this paper. This problem is known as the *Learning Parity in the Presence of Noise*, or LPN problem. The LPN problem involves finding a vector $\boldsymbol{x'}$ such that $|(\boldsymbol{A} \cdot \boldsymbol{x'}) \oplus \boldsymbol{z}| \leq \eta q$, where $|\boldsymbol{v}|$ represents the Hamming weight of vector $\boldsymbol{v}$. Formally, it is as follows:

Definition 1 (LPN Problem). *Let $\boldsymbol{A}$ be a random $q \times k$ binary matrix, let $\boldsymbol{x}$ be a random k-bit vector, let $\eta \in (0, \frac{1}{2})$ be a constant noise parameter, and let $\boldsymbol{\nu}$ be a random q-bit vector such that $|\boldsymbol{\nu}| \leq \eta q$. Given $\boldsymbol{A}$, η, and $\boldsymbol{z} = (\boldsymbol{A} \cdot \boldsymbol{x}) \oplus \boldsymbol{\nu}$, find a k-bit vector $\boldsymbol{x'}$ such that $|(\boldsymbol{A} \cdot \boldsymbol{x'}) \oplus \boldsymbol{z}| \leq \eta q$.*

The LPN problem may also be formulated and referred to as as the *Minimum Disagreement Problem* [9], or the problem of finding the closest vector to a random linear error-correcting code; also known as the syndrome decoding problem [2, 23]. Syndrome decoding is the basis of the McEliece public-key cryptosystem [27] and other cryptosystems, e.g., [8, 31]. Algebraic coding theory is also central to Stern's public-key identification scheme [35]. Chabaud offers attacks that, although infeasible, help to establish practical security parameters for error-correcting-code based cryptosystems [7].

The LPN problem is known to be NP-Hard [2], and is hard even within an approximation ratio of two [14]. A longstanding open question is whether this problem is difficult for random instances. A result by Kearns proves that the LPN is not efficiently solvable in the statistical query model [22]. An earlier result by Blum, Furst, Kearns, and Lipton [3] shows that given a random k-bit vector $\boldsymbol{a}$, an adversary who could weakly predict the value $\boldsymbol{a} \cdot \boldsymbol{x}$ with advantage $\frac{1}{k^c}$ could solve the LPN problem. Hopper and Blum [16, 17] show that the LPN problem is both pseudo-randomizable and log-uniform.

The best known algorithm to solve random LPN instances is due to Blum, Kalai, and Wasserman, and has a subexponential runtime of $2^{O(\frac{k}{\log k})}$ [4]. Based on a concrete analysis of this algorithm, we discuss estimates for lower-bounds on key sizes for the HB and HB$^+$ protocols in the full version of the paper.

As mentioned above, the basic HB protocol is only secure against passive eavesdroppers. It is not secure against an active adversary with the ability to query tags. If the same challenge $\boldsymbol{a}$ is repeated $\Omega((1 - 2\eta)^{-2})$ times, an adversary can learn the error-free value of $\boldsymbol{a} \cdot \boldsymbol{x}$ with very high probability. Given $\Omega(k)$ error-free values, an adversary can quickly compute $\boldsymbol{x}$ through Gaussian elimination.

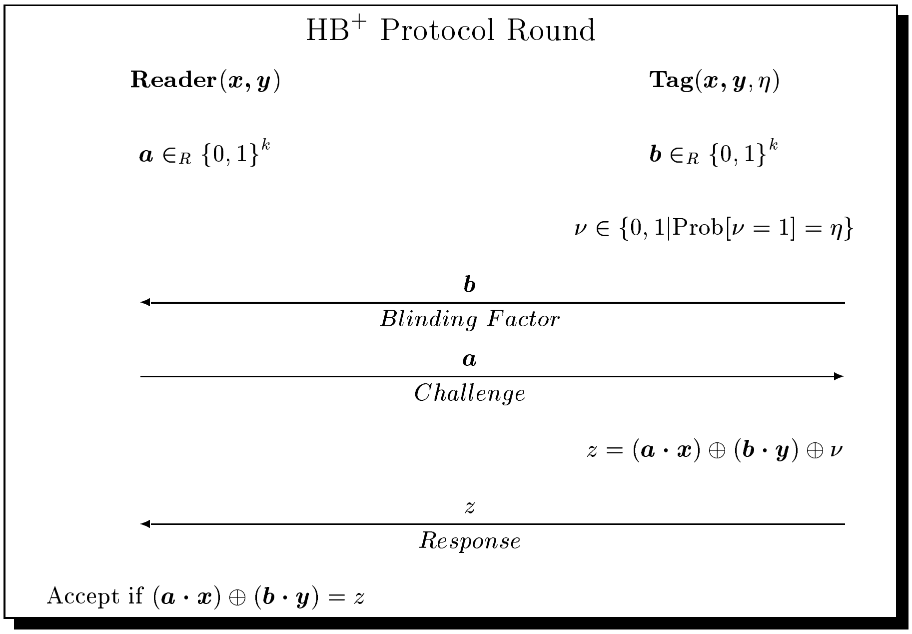

Fig. 3. A single round of the HB$^+$ protocol.

4 Authentication Against Active Adversaries

In this section, we show how to strengthen the HB protocol against active adversaries. We refer to the improved protocol as HB$^+$. HB$^+$ prevents corrupt readers from extracting tag secrets through adaptive (non-random) challenges, and thus prevents counterfeit tags from successfully authenticating themselves. Happily, HB$^+$ requires marginally more resources than the "passive" HB protocol in the previous section.

4.1 Defending Against Active Attacks: The HB$^+$ Protocol

The HB$^+$ protocol is quite simple, and shares a familiar "commit, challenge, respond" format with classic protocols like Fiat-Shamir identification. Rather than sharing a single k-bit random secret $\boldsymbol{x}$, the tag and reader now share an additional k-bit random secret $\boldsymbol{y}$.

Unlike the case in the HB protocol, the tag in the HB$^+$ protocol first generates random k-bit "blinding" vector $\boldsymbol{b}$ and sends it to the reader. As before, the reader challenges the tag with an k-bit random vector $\boldsymbol{a}$.

The tag then computes $z = (\boldsymbol{a} \cdot \boldsymbol{x}) \oplus (\boldsymbol{b} \cdot \boldsymbol{y}) \oplus \nu$, and sends the response z to the reader. The reader accepts the round if $z = (\boldsymbol{a} \cdot \boldsymbol{x}) \oplus (\boldsymbol{b} \cdot \boldsymbol{y})$. As before, the reader authenticates a tag after r rounds if the tag's response is incorrect in less than ηr rounds. This protocol is illustrated in Figure 3.

One reason that Hopper and Blum may not have originally proposed this protocol improvement is that it is inappropriate for use by humans. It requires the tag (playing the role of the human), to generate a random k-bit string $\boldsymbol{b}$ on each query. If the tag (or human) does not generate uniformly distributed $\boldsymbol{b}$ values, it may be possible to extract information on $\boldsymbol{x}$ or $\boldsymbol{y}$.

To convert HB^+ into a two-round protocol, an intuitive idea would be to have the tag transmit its $\boldsymbol{b}$ vector along with its response bit z. Being able to choose $\boldsymbol{b}$ *after* receiving $\boldsymbol{a}$, however, may give too much power to an adversarial tag. In particular, our security reduction in Section 5.4 relies on the tag transmitting its $\boldsymbol{b}$ value first. It's an open question whether there exists a secure two-round version of HB^+. Another open question is whether security is preserved if $\boldsymbol{a}$ and $\boldsymbol{b}$ are transmitted simultaneously on a duplex channel.

Beyond the requirements for the HB protocol, HB^+ only requires the generation of k random bits for $\boldsymbol{b}$ and additional storage for an k-bit secret $\boldsymbol{y}$. As before, computations can be performed bitwise; there is no need for the tag to store the entire vectors $\boldsymbol{a}$ or $\boldsymbol{b}$. Overall, this protocol is still quite efficient to implement in hardware, software, or perhaps even by a human being with a decent randomness source.

4.2 Security Intuition

As explained above, an active adversary can defeat the basic HB protocol and extract $\boldsymbol{x}$ by making adaptive, non-random $\boldsymbol{a}$ challenges to the tag. In our augmented protocol HB^+, an adversary can still, of course, select $\boldsymbol{a}$ challenges to mount an active attack.

By selecting its own random blinding factor $\boldsymbol{b}$, however, the tag effectively prevents an adversary from actively extracting $\boldsymbol{x}$ or $\boldsymbol{y}$ with non-random $\boldsymbol{a}$ challenges. Since the secret $\boldsymbol{y}$ is independent of $\boldsymbol{x}$, we may think of the tag as initiating an independent, interleaved HB protocol with the roles of the participants reversed. In other words, an adversary observing $\boldsymbol{b}$ and $(\boldsymbol{b} \cdot \boldsymbol{y}) \oplus \nu$ should not be able to extract significant information on $\boldsymbol{y}$.

Recall that the value $(\boldsymbol{b} \cdot \boldsymbol{y}) \oplus \nu$ is XORed with the the output of the original, reader-initiated HB protocol, $\boldsymbol{a} \cdot \boldsymbol{x}$. This prevents an adversary from extracting information through non-random $\boldsymbol{a}$ challenges. Thus, the value $(\boldsymbol{b} \cdot \boldsymbol{y}) \oplus \nu$ effectively "blinds" the value $\boldsymbol{a} \cdot \boldsymbol{x}$ from both passive and active adversaries.

This observation underlies our proof strategy for the security of HB^+. We argue that an adversary able to efficiently learn $\boldsymbol{y}$ can efficiently solve the LPN problem. In particular, an adversary that does not know $\boldsymbol{y}$ cannot guess $\boldsymbol{b} \cdot \boldsymbol{y}$, and therefore cannot learn information about $\boldsymbol{x}$ from a tag response z.

The blinding therefore protects against leaking the secret $\boldsymbol{x}$ in the face of active attacks. Without knowledge of $\boldsymbol{x}$ or $\boldsymbol{y}$, an adversary cannot create a fake tag that will respond correctly to a challenge $\boldsymbol{a}$. In other words, cloning will be infeasible. In Section 5, we will present a concrete reduction from the LPN problem to the security of the HB^+ protocol. In other words, an adversary with some significant advantage of impersonating a tag in the HB^+ protocol can be used to solve the LPN problem with some significant advantage.

5 Security Proofs

We will first present concrete security notation in Section 5.1. Section 5.2 reviews key aspects of the Blum et al. proof strategy that reduces the LPN problem to the security of the HB protocol [3]. We offer a more thorough and concrete version of the Blum et al. reduction in Section 5.3. In Section 5.4, we present a concrete reduction from the HB protocol to the HB^+ protocol. Finally, in Section 5.5, we combine these results to offer a concrete reduction of the LPN problem to the security of the HB^+ protocol.

5.1 Notation and Definitions

We define a tag-authentication system in terms of a pair of probabilistic functions $(\mathcal{R}, \mathcal{T})$, namely a reader function $\mathcal{R}$ and a tag function $\mathcal{T}$.

The tag function $\mathcal{T}$ is defined in terms of a noise parameter η, a k-bit secret $\boldsymbol{x}$, and a set of q random k-bit vectors $\{\boldsymbol{a}^{(i)}\}_{i=1}^{q}$ that we view for convenience as a matrix $\boldsymbol{A}$. Additionally, $\mathcal{T}$ includes a k-bit secret y for protocol HB^+. We let q be the maximum number of protocol invocations on $\mathcal{T}$ in our experiments.

For protocol HB, we denote the fully parameterized tag function by $\mathcal{T}_{x,\boldsymbol{A},\eta}$. On the ith invocation of this protocol, $\mathcal{T}$ is presumed to output $(\boldsymbol{a}^{(i)}, (\boldsymbol{a^{(i)}} \cdot \boldsymbol{x}) \oplus \nu)$. Here ν is a bit of noise parameterized by η. This models a passive eavesdropper observing a round of the HB protocol. Note that the oracle $\mathcal{T}_{x,\boldsymbol{A},\eta}$ takes no input and essentially acts as an interface to a flat transcript. For this protocol, the reader $\mathcal{R}_x$ takes as input a pair $(\boldsymbol{a}, z)$. It outputs either "accept" or "reject".

For protocol HB^+, we denote a fully parameterized tag function as $\mathcal{T}_{x,y,\eta}$. This oracle internally generates random blinding vectors $\boldsymbol{b}$. On the ith invocation of $\mathcal{T}$ for this protocol, the tag outputs some random $\boldsymbol{b}^{(i)}$, takes a challenge vector $\boldsymbol{a}^{(i)}$ (that could depend on $\boldsymbol{b}^{(i)}$) as input, and outputs $z = (\boldsymbol{a^{(i)}} \cdot \boldsymbol{x}) \oplus (\boldsymbol{b^{(i)}} \cdot \boldsymbol{y}) \oplus \nu$. This models an active adversary querying a tag in a round of the HB^+ protocol. For this protocol, the reader $\mathcal{R}_{x,y}$ takes as input a triple $(\boldsymbol{a}, \boldsymbol{b}, z)$ and outputs either "accept" or "reject".

For both protocols HB and HB^+, we consider a two-phase attack model involving an adversary comprising a pair of functions $\mathcal{A} = (\mathcal{A}_{query}, \mathcal{A}_{clone})$, a reader $\mathcal{R}$, and a tag $\mathcal{T}$. In the first, "query" phase, the adversarial function $\mathcal{A}_{query}$ has oracle access to $\mathcal{T}$ and outputs some state σ.

The second, "cloning" phase involves the adversarial function $\mathcal{A}_{clone}$. The function $\mathcal{A}_{clone}$ takes as input a state value σ. In HB^+, it outputs a blinding factor $\boldsymbol{b'}$ (when given the input command "initiate"). In both HB and HB^+, when given the input command "guess", $\mathcal{A}_{clone}$ takes the full experimental state as input, and outputs a response bit z'.

We presume that a protocol invocation takes some fixed amount of time (as would be the case, for example, in an RFID system). We characterize the total protocol time by three parameters: the number of queries to a $\mathcal{T}$ oracle, q; the computational runtime t_1 of $\mathcal{A}_{query}$; and the computational runtime t_2 of $\mathcal{A}_{clone}$. Let D be some distribution of $q \times k$ matrices. We let $\stackrel{R}{\leftarrow}$ denote uniform random assignment. Other notation should be clear from context.

Experiment $\mathbf{Exp}_{\mathcal{A},D}^{HB-attack}[k,\eta,q]$

$x \stackrel{R}{\leftarrow} \{0,1\}^k$;

$\boldsymbol{A} \stackrel{R}{\leftarrow} D$

$\sigma \leftarrow \mathcal{A}_{query}^{\mathcal{T}x,\boldsymbol{A},\eta}$;

$\boldsymbol{a}' \stackrel{R}{\leftarrow} \{0,1\}^k$;

$z' \leftarrow \mathcal{A}_{clone}(\sigma, \boldsymbol{a}', \text{"guess"})$;

Output $\mathcal{R}_x(\boldsymbol{a}', z')$.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{HB^+-attack}[k,\eta,q]$

$x, y \stackrel{R}{\leftarrow} \{0,1\}^k$;

$\sigma \leftarrow \mathcal{A}_{query}^{\mathcal{T}_{x,y,\eta}}$;

$\boldsymbol{b}' \leftarrow \mathcal{A}_{clone}(\sigma, \text{"initiate"})$;

$\boldsymbol{a}' \stackrel{R}{\leftarrow} \{0,1\}^k$;

$z' \leftarrow \mathcal{A}_{clone}(\sigma, \boldsymbol{a}', \boldsymbol{b}', \text{"guess"})$;

Output $\mathcal{R}_{x,y}(\boldsymbol{a}', \boldsymbol{b}', z')$.

Consider $\mathcal{A}$'s advantage for key-length k, noise parameter η, over q rounds. In the case of the HB-attack experiment, this advantage will be over matrices $\boldsymbol{A}$ drawn from the distribution D:

$$\mathsf{Adv}_{\mathcal{A},D}^{HB-attack}(k,\eta,q) = \left| Pr\left[\mathbf{Exp}_{\mathcal{A},D}^{HB-attack}[k,\eta,q] = \text{"accept"}\right] - \frac{1}{2}\right|$$

Let $Time(t_1, t_2)$ represent the set of all adversaries $\mathcal{A}$ with runtimes t_1 and t_2, respectively. Denote the maximum advantage over $Time(t_1, t_2)$:

$$\mathsf{Adv}_{D}^{HB-attack}(k,\eta,q,t_1,t_2) = \max_{\mathcal{A}\in Time(t_1,t_2)} \{\mathsf{Adv}_{\mathcal{A},D}^{HB-attack}(k,\eta,q)\}$$

The definitions for Adv are exactly analogous for HB$^+$-attack, except that there is no input distribution D, as adversarial queries are active.

5.2 Blum et al. Proof Strategy Outline

Given an adversary $\mathcal{A}$ that achieves the advantage $\mathsf{Adv}_{\mathcal{A},U}^{HB-attack}(k,q,\eta,t_1,t_2) = \epsilon$, Blum et al. [3] offer a proof strategy to extract bits of x, and thus solve the LPN problem. If ϵ is a non-negligible function of k, then x can be extracted by their reduction in polynomial time.

To extract the ith bit of the secret x, the Blum et al. reduction works as follows. The reduction takes a given LPN instance $(\boldsymbol{A}, \boldsymbol{z})$ and randomly modifies it to produce a new instance $(\boldsymbol{A}', \boldsymbol{z}')$.

The modification involves two steps. First, a vector $\boldsymbol{x}'$ is chosen uniformly at random and $\boldsymbol{z}' = (\boldsymbol{z} \oplus \boldsymbol{A}) \cdot \boldsymbol{x}' = (\boldsymbol{A} \cdot (\boldsymbol{x} \oplus \boldsymbol{x}')) \oplus \boldsymbol{\nu}$ is computed. Note that thanks to the random selection of x', the vector $(\boldsymbol{x} \oplus \boldsymbol{x}')$ is uniformly distributed. Second, the ith column of $\boldsymbol{A}$ is replaced with random bits. To view this another way, denote the subspace of matrices obtained by uniformly randomizing the ith column of $\boldsymbol{A}$ as $R_i^{\boldsymbol{A}}$. The second step of the modification involves setting $\boldsymbol{A}' \stackrel{R}{\leftarrow} R_i^{\boldsymbol{A}}$. Once computed as described,the modified problem instance $(\boldsymbol{A}', \boldsymbol{z}')$ is fed to an HB adversary $\mathcal{A}_{query}$.

Suppose that the ith bit of $(\boldsymbol{x} \oplus \boldsymbol{x}')$, which we denote $(x \oplus x')_i$, is a binary '1'. In this case, since $\boldsymbol{A}$ is a randomly distributed matrix (because HB challenges are random), and the secret $\boldsymbol{x}$ is also randomly distributed, the bits of $\boldsymbol{z}'$ are random. In other words, thanks to the '1' bit, the randomized ith row of $\boldsymbol{A'}$ "counts" in the computation of $\boldsymbol{z'}$, which therefore comes out random. Hence $\boldsymbol{z'}$ contains no information about the correct value of $\boldsymbol{A} \cdot (\boldsymbol{x} \oplus \boldsymbol{x'})$ or about the secret x. Since $\mathcal{A}_{query}$ cannot pass any meaningful information in σ to $\mathcal{A}_{clone}$ in this case, $\mathcal{A}_{clone}$ can do no better than random guessing of parity bits, and enjoys no advantage.

In contrast, suppose that $(x \oplus x')_i$, is a binary '0'. In this case, the ith row of $\boldsymbol{A'}$ does not "count" in the computation of $\boldsymbol{z'}$, and does not have a randomizing effect. Hence $\boldsymbol{z'}$ may contain meaningful information about the secret $\boldsymbol{x}$ in this case. As a result, when $\mathcal{A}_{clone}$ shows an advantage over modified problem instances $(\boldsymbol{A'}, \boldsymbol{z'})$ for a particular fixed choice of $\boldsymbol{x'}$, it is clear for those instances that $(x \oplus x')_i = 0$, i.e. $x_i = x'_i$.

In summary then, the Blum et al. reduction involves presentation of suitably modified problem instances $(\boldsymbol{A'}, \boldsymbol{z'})$ to HB adversary $\mathcal{A}$. By noting choices of $\boldsymbol{x}'$ for which $\mathcal{A}$ demonstrates an advantage, it is possible in principle to learn individual bits of the secret $\boldsymbol{x}$. With presentation of enough modified problem instances to $\mathcal{A}$, it is possible to learn $\boldsymbol{x}$ completely with high probability.

5.3 Reduction from LPN to HB-attack

We will show a concrete reduction from the LPN problem to the HB-attack experiment. This is essentially a concrete version of Blum et al.'s asymptotic reduction strategy from [3] and is an important step in proving Theorem 1.

Unfortunately, the original Blum et al. proof strategy does not account for the fact that while $\mathcal{A}$'s advantage may be non-negligible over random matrices, it may actually be negligible over modified $(\boldsymbol{A'}, \boldsymbol{z'})$ values, i.e., over the distribution $R_i^{\boldsymbol{A}}$. Matrices are not independent over this distribution: Any two sample matrices are identical in all but one column. Thus, it is possible in principle that $\mathcal{A}$ loses its advantage over this distribution of matrices and that the reduction fails to work. This is a problem that we must remedy here.

We address the problem by modifying a given sample matrix only once. A modified matrix $\boldsymbol{A'}$ in our reduction is uniformly distributed. This is because it is chosen uniformly from a random $R_i^{\boldsymbol{A}}$ subspace associated with a random matrix $\boldsymbol{A}$. Additionally, since we use a fresh sample for each trial, our modified matrices are necessarily independent of each other. The trade-off is that kL times as many sample matrices are needed for our reduction, where L is the number of trials per bit.

This is an inefficient solution in terms of samples. It is entirely possible that the adversary's advantage is preserved when, for each column j, samples are drawn from the $R_i^{\boldsymbol{A}_j}$ subspace for a matrix $\boldsymbol{A}_j$. It might even be possible to devise a rigorous reduction that uses a single matrix $\boldsymbol{A}$ for all columns. We leave these as open questions.

Lemma 1. *Let* $\mathsf{Adv}_U^{HB-Attack}(k, \eta, q, t_1, t_2) = \epsilon$, *where* U *is a uniform distribution over binary matrices* $\mathbb{Z}_2^{q \times k}$, *and let* $\mathcal{A}$ *be an adversary that achieves this* ϵ*-advantage. Then there is an algorithm* $\mathcal{A}'$ *with running time* $t'_1 \leq kLt_1$ *and*

$t_2' \leq kLt_2$, *where* $L = \frac{8(\ln k - \ln\ln k)}{(1-2\eta)^2\epsilon^2}$, *that makes* $q' \leq kLq + 1$ *queries that can correctly extract all* k *bits of* x *with probability* $\epsilon' \geq \frac{1}{k}$.

5.4 Reduction from HB-attack to HB$^+$-attack

We show that an HB$^+$-attack adversary with ζ-advantage can be used to build an HB-attack adversary with advantage $\frac{\zeta^3}{4} - \frac{\zeta^3+1}{2k}$. We provide concrete costs of this reduction that will be used for Theorem 1.

Lemma 3. *If* $\mathsf{Adv}_U^{HB^+-Attack}(k, \eta, q, t_1, t_2) = \zeta$, *then*

$$\mathsf{Adv}_U^{HB-Attack}(k, \eta, q', t_1', t_2') \geq \frac{\zeta^3}{4} - \frac{\zeta^3+1}{2k},$$

where $q' \leq q(2 + \log_2 q)$, $t_1' \leq kq't_1$, $t_2' \leq 2kt_2$, *and* $k \geq 9$.

(Lemma 2, a technical lemma, is omitted here. For consistency, however, we retain the numbering of lemmas from the full version of this paper.)

Lemma 3 is the main technical core of the paper, but its proof must be omitted here due to lack of space. It is worth briefly explaining the proof intuition. The proof naturally involves a simulation where the HB-attack adversary $\mathcal{A}$ makes calls to the furnished HB$^+$-attack adversary, which we call $\mathcal{A}^+$. In other words, $\mathcal{A}$ simulates the environment for $\mathbf{Exp}_{\mathcal{A}^+}^{HB^+-attack}$. The goal of $\mathcal{A}$ is to use $\mathcal{A}^+$ to compute a correct target response w to an HB challenge vector $\boldsymbol{a}$ that $\mathcal{A}$ itself receives in an experiment $\mathbf{Exp}_{\mathcal{A}}^{HB-attack}$.

$\mathcal{A}$ makes its calls to $\mathcal{A}^+$ in a special way: It "cooks" transcripts obtained from its own HB oracle before passing them to $\mathcal{A}^+$ during its simulation of the query phase of $\mathbf{Exp}^{HB^+-attack}$. The "cooked" transcripts are such that the target value w is embedded implicitly in a secret bit of the simulated HB$^+$ oracle.

In its simulation of the cloning phase of $\mathbf{Exp}^{HB^+-attack}$, the adversary $\mathcal{A}$ extracts the embedded secret bit using a standard cryptographic trick. After $\mathcal{A}^+$ has committed a blinding value $\boldsymbol{b}$, $\mathcal{A}$ rewinds $\mathcal{A}^+$ to as to make two different challenges $\boldsymbol{a}^{(0)}$ and $\boldsymbol{a}^{(1)}$ relative to $\boldsymbol{b}$. By looking at the difference in the responses, $\mathcal{A}$ can extract the embedded secret bit and compute its own target response w.

There are two main technical challenges in the proof. The first is finding the right embedding of w in a secret bit of the simulated HB$^+$-oracle. Indeed, our approach is somewhat surprising. One might intuitively expect $\mathcal{A}$ instead to cause $\mathcal{A}^+$ to emit a *response* equal to w during the simulation; after all, w itself is intended to be a tag response furnished by $\mathcal{A}$, rather than a secret bit. (We could not determine a good way to have w returned as a response.) The second challenge comes in the rewinding and extraction. There is the possibility of a non-uniformity in the responses of $\mathcal{A}^+$. An important technical lemma (Lemma 2) is necessary to bound this non-uniformity. The statement and proof of Lemma 2 are omitted here, but provided in the full version of the paper.

5.5 Reduction of LPN to HB$^+$-attack

By combining Lemmas 1 and 3, we obtain a concrete reduction of the LPN problem to the HB$^+$-attack experiment. Given an adversary that has an ϵ-advantage against the HB$^+$-attack experiment within a specific amount of time and queries, we can construct an adversary that solves the LPN problem within a concrete upper bound of time and queries. The following theorem follows directly from Lemmas 1 and 3.

Theorem 1. *Let* $\mathsf{Adv}^{HB^+-Attack}(k, \eta, q, t_1, t_2) = \zeta$*, where* U *is a uniform distribution over binary matrices* $\mathbb{Z}_2^{q \times k}$*, and let* $\mathcal{A}$ *be an adversary that achieves this* ζ*-advantage. Then there is an algorithm that can solve a random* $q' \times k$ *instance of the LPN problem in time* (t'_1, t'_2) *with probability* $\frac{1}{k}$*, where* $t'_1 \leq k^2 Lq(2+\log_2 q)t_1$*,* $t'_2 \leq 2k^2 Lt_2$*,* $q' \leq kLq(2 + \log_2 q)$*, and* $L = \frac{128k^4(\ln k - \ln\ln k)}{(1-2\eta)^2(\zeta^3(k-2)+2)^2}$.

To put this in asymptotic terms, the LPN problem may be solved by an adversary where $\mathsf{Adv}^{HB^+-Attack}(k, \eta, q, t_1, t_2) = \zeta$ in time $O(\frac{(k^5 \log k)(q \log q)\ t}{(1-2\eta^2)\zeta^6})$, where $t = t_1 + t_2$.

6 Conclusion and Open Questions

In summary, this paper presents a new authentication protocol named HB$^+$ that is appropriate for low-cost pervasive computing devices. The HB$^+$ protocol is secure in the presence of both passive and active adversaries and should be implementable within the tight resource constraints of today's EPC-type RFID tags. The security of the HB$^+$ protocol is based on the LPN problem, whose hardness over random instances remains an open question.

It is also open question whether the two-round variant of HB$^+$ that was briefly mentioned in Section 4.1 is secure. The security of concurrent executions of the HB$^+$ protocol is also unknown. Our security proof uses a rewinding technique that would be take time exponential in the number of concurrent rounds.

Our results here do not offer direct practical guidance for parameterization in real RFID tags. It would be desirable to see a much tighter concrete reduction than we give here. One avenue might be improvement to the Blum et al. reduction. As mentioned in Section 5.3, the efficiency of the modified concrete version of Blum et al. reduction [3] may be improved. Our version uses sample values only once. It may be possible to use a single sample to generate several trials per column, or perhaps to generate trials for every column. This lowers the concrete query costs. It is unclear, however, whether the reduction holds over non-uniform input distributions.

Finally, there is second human authentication protocol by Hopper and Blum, based on the "Sum of k Mins" problem and error-correcting challenges [5, 17]. Unlike the HB protocol, this protocol is already supposed to be secure against active adversaries. However, the hardness of the "Sum of k Mins" has not been studied as much as the LPN problem, nor is it clear whether this protocol can efficiently be adapted for low-cost devices. These remain open avenues of research.

References

[1] Anderson, R., and Kuhn, M. Low Cost Attacks on Tamper Resistant Devices. In *International Workshop on Security Protocols* (1997), vol. 1361 of *Lecture Notes in Computer Science*, pp. 125–136.

[2] Berlekamp, E. R., McEliece, R. J., and Tilborg, V. On the Inherent Intractability of Certain Coding Problems. *IEEE Transactions on Information Theory 24* (1978), 384–386.

[3] Blum, A., Furst, M., Kearns, M., and Lipton, R. J. Cryptographic Primitives Based on Hard Learning Problems. In *Advances in Cryptology – CRYPTO'93* (1994), vol. 773 of *Lecture Notes in Computer Science*, pp. 278–291.

[4] Blum, A., Kalai, A., and Wasserman, H. Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model. *Journal of the ACM 50*, 4 (July 2003), 506–519.

[5] Blum, M., Luby, M., and Rubinfeld, R. Self-Testing/Correcting with Applications to Numerical Problems. In *Symposium on Theory of Computation* (1990), pp. 73–83.

[6] Bono, S., Green, M., Stubblefield, A., Juels, A., Rubin, A., and Szydlo, M. Security Analysis of a Cryptographically-Enabled RFID Device. In *USENIX Security* (2005). To appear. Available at http://rfidanalysis.org/.

[7] Chabaud, F. On the Security of Some Cryptosystems Based on Error-Correcting Codes. In *Advances in Cryptology - EUROCRYPT* (1995), vol. 950 of *Lecture Notes in Computer Science*, pp. 131–139.

[8] Courtois, N., Finiasz, M., and Sendrier, N. How to Achieve a McEliece-based Digital Signature Scheme. In *Advances in Cryptology - ASIACRYPT* (2001), vol. 2248 of *Lecture Notes in Computer Science*, pp. 157–174.

[9] Crawford, J. M., Kearns, M. J., and Shapire, R. E. The Minimal Disagreement Parity Problem as a Hard Satisfiability Problem. Tech. rep., Computational Intelligence Research Laboratory and AT&T Bell Labs, February 1994.

[10] EPCglobal. Website. http://www.epcglobalinc.org/, 2005.

[11] Feldhofer, M., Dominikus, S., and Wolkerstorfer, J. Strong Authentication for RFID Systems using the AES Algorithm. In *Cryptographic Hardware in Embedded Systems (CHES)* (2004).

[12] Floerkemeier, C., and Lampe, M. Issues with RFID Usage in Ubiquitous Computing Applications. In *Pervasive Computing (PERVASIVE)* (2004), vol. 3001 of *Lecture Notes in Computer Science*, pp. 188–193.

[13] Food and Drug Administration. Combating counterfeit drugs. Tech. rep., US Department of Health and Human Services, Rockville, Maryland, Februrary 2004.

[14] Håstad, J. Some Optimal Inapproximability Results. In *Symposium on Theory of Computing* (1997), pp. 1–10.

[15] Henrici, D., and Müller, P. Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers. In *Pervasive Computing and Communications (PerCom)* (2004), IEEE Computer Society, pp. 149–153.

[16] Hopper, N., and Blum, M. A Secure Human-Computer Authentication Scheme. Tech. Rep. CMU-CS-00-139, Carnegie Mellon University, 2000.

[17] Hopper, N. J., and Blum, M. Secure Human Identification Protocols. In *Advances in Cryptology - ASIACRYPT* (2001), vol. 2248 of *Lecture Notes in Computer Science*, pp. 52–66.

[18] Juels, A. Minimalist Cryptography for RFID Tags. In *Security in Communication Networks* (2004), C. Blundo and S. Cimato, Eds., vol. 3352 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 149–164.

[19] Juels, A. "Yoking Proofs" for RFID Tags. In *Pervasive Computing and Communications Workshop* (2004), IEEE Press.

[20] Juels, A., and Pappu, R. Squealing Euros: Privacy Protection in RFID-Enabled Banknotes. In *Financial Cryptography* (2003), vol. 2742 of *Lecture Notes in Computer Science*, pp. 103–121.

[21] Juels, A., Rivest, R. L., and Szydlo, M. The blocker tag: selective blocking of RFID tags for consumer privacy. In *Proceedings of the 10th ACM conference on Computer and communication security* (2003), ACM Press, pp. 103–111.

[22] Kearns, M. Efficient Noise-Tolerant Learning from Statistical Queries. *Journal of the ACM 45*, 6 (November 1998), 983–1006.

[23] MacWilliams, F., and Sloane, N. *The Theory of Error-Correcting Codes.* North-Holland, Amsterdam, 1977.

[24] Mandel, J., Roach, A., and Winstein, K. MIT Proximity Card Vulnerabilities. Tech. rep., Massachusetts Institute of Technology, March 2004.

[25] Matsumoto, T. Human-computer Cryptography: An Attempt. In *Computer and Communications Security* (1996), ACM Press, pp. 68–75.

[26] Matsumoto, T., and Imai, H. Human Identification through Insecure Channel. In *Advances in Cryptology - EUROCRYPT* (1991), vol. 547 of *Lecture Notes in Computer Science*, pp. 409–421.

[27] McEliece, R. J. DSN Progress Report. Tech. Rep. 42–44, JPL-Caltech, 1978.

[28] Miller, G. A. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *Psychological Review 63* (1956), 81–97.

[29] Molnar, D., and Wagner, D. Privacy and Security in Library RFID : Issues, Practices, and Architectures. In *Computer and Communications Security* (2004), B. Pfitzmann and P. McDaniel, Eds., ACM, pp. 210 – 219.

[30] Naor, M., and Pinkas, B. Visual Authentication and Identification. In *Advances in Cryptology - CRYPTO* (1997), vol. 1294 of *Lecture Notes in Computer Science*, pp. 322–336.

[31] Niederreiter, H. Knapsack-Type Cryptosystems and Algebraic Coding Theory. *Problems of Control and Information Theory 15*, 2 (1986), 159–166.

[32] Ohkubo, M., Suzuki, K., and Kinoshita, S. Efficient Hash-Chain Based RFID Privacy Protection Scheme. In *Ubiquitous Computing (UBICOMP)* (September 2004).

[33] Sarma, S. E., Weis, S. A., and Engels, D. W. RFID Systems and Security and Privacy Implications. In *Workshop on Cryptographic Hardware and Embedded Systems* (2002), vol. 2523, Lecture Notes in Computer Science, pp. 454–470.

[34] Shamos, M. I. Paper v. Electronic Voting Records - An Assessment, 2004. Paper written to accompany panel presentation at Computers, Freedom, and Privacy Conference '04. Available at http://euro.ecom.cmu.edu/people/faculty/mshamos/paper.htm.

[35] Stern, J. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory 42*, 6 (1996), 1757–1768.

[36] Vajda, I., and Buttyan, L. Lightweight Authentication Protocols for Low-Cost RFID Tags. In *Ubiquitious Computing (UBICOMP)* (2003).
[37] Verichip. Website. http://www.4verichip.com/, 2005.
[38] Wang, C.-H., Hwang, T., and Tsai, J.-J. On the Matsumoto and Imai's Human Identification Scheme. In *EuroCrypt '95* (1995), vol. 921 of *Lecture Notes in Computer Science*, pp. 382–392.
[39] Weis, S. A., Sarma, S. E., Rivest, R. L., and Engels, D. W. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In *Security in Pervasive Computing* (2004), vol. 2802 of *Lecture Notes in Computer Science*, pp. 201–212.