Lapin: An Efficient Authentication Protocol Based on Ring-LPN

Stefan Heyse[1], Eike Kiltz[1], Vadim Lyubashesvky[2],
Christof Paar[1], and Krzysztof Pietrzak[3]⋆

[1] Ruhr-Universität Bochum
[2] INRIA / ENS, Paris
[3] IST Austria

Abstract. We propose a new authentication protocol that is provably secure based on a *ring* variant of the learning parity with noise (LPN) problem. The protocol follows the design principle of the LPN-based protocol from Eurocrypt'11 (Kiltz et al.), and like it, is a two round protocol secure against *active* attacks. Moreover, our protocol has small communication complexity and a very small footprint which makes it applicable in scenarios that involve low-cost, resource-constrained devices.

Performance-wise, our protocol is more efficient than previous LPN-based schemes, such as the many variants of the Hopper-Blum (HB) protocol and the aforementioned protocol from Eurocrypt'11. Our implementation results show that it is even comparable to the standard challenge-and-response protocols based on the AES block-cipher. Our basic protocol is roughly 20 times slower than AES, but with the advantage of having 10 times smaller code size. Furthermore, if a few hundred bytes of non-volatile memory are available to allow the storage of some off-line pre-computations, then the online phase of our protocols is only twice as slow as AES.

Keywords: HB protocols, RFID authentication, LPN problem, Ring-LPN problem

1 Introduction

Lightweight shared-key authentication protocols, in which a tag authenticates itself to a reader, are extensively used in resource-constrained devices such as radio-frequency identification (RFID) tags or smart cards. The straight-forward approach for constructing secure authentications schemes is to use low-level symmetric primitives such as block-ciphers, e.g. AES [DR02]. In their most basic form, the protocols consist of the reader sending a short challenge c and the tag responding with $\mathsf{AES}_K(c)$, where K is the shared secret key. The protocol is secure if AES fulfils a strong, *interactive* security assumption, namely that it behaves like a strong pseudo-random function.

Authentication schemes based on AES have some very appealing features: they are extremely fast, consist of only 2 rounds, and have very small communication complexities. In certain scenarios, however, such as when low-cost and resource-constrained devices are involved, the relatively large gate-count and code size used to implement AES may pose a problem. One approach to overcome the restrictions presented by low-weight devices is to construct a low-weight block cipher (e.g. PRESENT [BKL$^+$07]), while another approach has been to deviate entirely from block-cipher based constructions and build a *provably-secure* authentication scheme based on the hardness of some mathematical problem. In this work, we concentrate on this second approach.

Ideally, one would like to construct a scheme that incorporates all the beneficial properties of AES-type protocols, while also acquiring the additional provable security and smaller code description characteristics. In the past decade, there have been proposals that achieved some, but not all, of these criteria. Most of these proposals are extensions and variants of the Hopper-Blum (HB) protocol, recently a protocol following a different blueprint has been proposed by Kiltz et al. [KPC$^+$11]. Our proposal can be seen as a continuation of this line of research that contains all the advantages enjoyed by LPN-based protocols, while at the same time, getting even closer to enjoying the benefits of AES-type schemes.

OVERVIEW OF OUR RESULTS. In this work we present a new symmetric authentication protocol which (i) is provably-secure against active attacks (as defined in [JW05]) based on the Ring-LPN assumption,

⋆ Supported by the European Research Council / ERC Starting Grant (259668-PSPC)

a natural variant of the standard LPN (learning parity with noise) assumption; (ii) consists of 2 rounds; (iii) has small communication complexity (approximately 1300 bits); (iv) has efficiency comparable to AES-based challenge-response protocols (depending on the scenario), but with a much smaller code size. To demonstrate the latter we implemented the tag part of our new protocol in a setting of high practical relevance – a low-cost 8-bit microcontroller which is a typical representative of a CPU to be found on lightweight authentication tokens, and compared its performance (code size and running time) with an AES implementation on the same platform.

PREVIOUS WORKS. Hopper and Blum [HB00,HB01] proposed a 2-round authentication protocol that is secure against *passive* adversaries based on the hardness of the LPN problem (we remind the reader of the definition of the LPN problem in Section 1.2). The characteristic feature of this protocol is that it requires very little workload on the part of the tag and the reader. Indeed, both parties only need to compute vector inner products and additions over F_2, which makes this protocol (thereafter named HB) a good candidate for lightweight applications.

Following this initial work, Juels and Weis constructed a protocol called HB^+ [JW05] which they proved to be secure against more realistic, so called *active* attacks. Subsequently, Katz et al. [KS06a], [KS06b,KSS10] provided a simpler security proof for HB^+ as well as showed that it remains secure when executed in parallel. Unlike the HB protocol, however, HB^+ requires three rounds of communication between tag and reader. From a practical aspect, 2 round authentication protocols are often advantageous over 3 round protocols. They often show a lower latency which is especially pronounced on platforms where the establishment of a communication in every directions is accompanied by a fixed initial delay. An additional drawback of both HB and HB^+ is that their communication complexity is on the order of hundreds of thousands of bits, which makes them almost entirely impractical for lightweight authentication tokens because of timing and energy constraints. (The contactless transmission of data on RFIDs or smart cards typically requires considerably more energy than the processing of the same data.)

To remedy the overwhelming communication requirement of HB^+, Gilbert et al. proposed the three-round $\mathsf{HB}^\sharp$ protocol [GRS08a]. A particularly practical instantiation of this protocol requires fewer than two thousand bits of communication, but is no longer based on the hardness of the LPN problem. Rather than using independent randomness, the $\mathsf{HB}^\sharp$ protocol utilized a Toeplitz matrix, and is thus based on a plausible assumption that the LPN problem is still hard in this particular scenario.

A feature that the $\mathsf{HB}, \mathsf{HB}^+$, and $\mathsf{HB}^\sharp$ protocols have in common is that at some point the reader sends a random string r to the tag, which then must reply with $\langle r, s\rangle + e$, the inner product of r with the secret s plus some small noise e. The recent work of Kiltz et al. [KPC$^+$11] broke with this approach, and they were able to construct the first 2-round LPN-based authentication protocol (thereafter named EC11) that is secure against active attacks. In their challenge-response protocol, the reader sends some challenge bit-string c to the tag, who then answers with a noisy inner product of a random r (which the tag chooses itself) and a session-key $K(c)$, where $K(c)$ selects (depending on c) half of the bits from the secret s. Unfortunately, the EC11 protocol still inherits the large communication requirement of HB and HB^+. Furthermore, since the session key $K(c)$ is computed using bit operations, it does not seem to be possible to securely instantiate EC11 over structured (and hence more compact) objects such as Toeplitz matrices (as used in $\mathsf{HB}^\sharp$ [GRS08a]).

1.1 Our contributions

PROTOCOL. In this paper we propose a variant of the EC11 protocol from [KPC$^+$11] which uses an "algebraic" derivation of the session key $K(c)$, thereby allowing to be instantiated over a carefully chosen ring $\mathsf{R} = \mathsf{F}_2[X]/(f)$. Our scheme is no longer based on the hardness of LPN, but rather on the hardness of a natural generalization of the problem to rings, which we call Ring-LPN(see Section 3 for the definition of the problem.) The general overview of our protocol is quite simple. Given a challenge c from the reader, the tag answers with $(r, z = r \cdot K(c) + e) \in \mathsf{R} \times \mathsf{R}$, where r is a random ring element, e is a

Table 1. Summary of implementation results

Protocol	Time (cycles)		Code size
	online	offline	(bytes)
Ours: reducible f (§5.1)	$30,000$	$82,500$	$1,356$
Ours: irreducible f (§5.2)	$21,000$	$174,000$	459
AES-based [LLS09,Tik]	$10,121$	0	$4,644$

low-weight ring element, and $K(c) = sc + s'$ is the session key that depends on the shared secret key $K = (s, s') \in \mathsf{R}^2$ and the challenge c. The reader accepts if $e' = r \cdot K(c) - z$ is a polynomial of low weight, cf. Figure 1 in Section 4. Compared to the HB and HB^+ protocols, ours has one less round and a dramatically lower communication complexity. Our protocol has essentially the same communication complexity as $\mathsf{HB}^\sharp$, but still retains the advantage of one fewer round. And compared to the two-round EC11 protocol, ours again has the large savings in the communication complexity. Furthermore, it inherits from EC11 the simple and tight security proof that, unlike three-round protocols, does not use rewinding.

We remark that while our protocol is provably secure against active attacks, we do not have a proof of security against man-in-the-middle ones. Still, as argued in [KSS10], security against active attacks is sufficient for many use scenarios (see also [JW05,KW05,KW06]). We would like to mention that despite man-in-the-middle attacks being outside our "security model", we think that it is still worthwhile investigating whether such attacks do in fact exist, because it presently seems that all previous man-in-the middle attacks against HB-type schemes along the lines of Gilbert et al. [GRS05] and of Ouafi et al. [OOV08] do not apply to our scheme. In Appendix A, however, we do present a man-in-the-middle attack that works in time approximately $n^{1.5} \cdot 2^{\lambda/2}$ (where n is the dimension of the secret and λ is the security parameter) when the adversary can influence on the order of $n^{1.5} \cdot 2^{\lambda/2}$ interactions between the reader and the tag. To resist this attack, one could simply double the security parameter, but we believe that even for $\lambda = 80$ (and $n > 512$, as it is currently set in our scheme) this attack is already impractical because of the extremely large number of interactions that the adversary will have to observe and modify.

IMPLEMENTATION. We demonstrate that our protocol is indeed practical by providing a lightweight implementation of the tag part of the protocol. (The reader is typically not run on a constrained device and therefore we do not consider its performance.) The target platform was an AVR ATmega163 [Atm] based smart card. The ATmega163 is a small 8-bit microcontroller which is a typical representative of a CPU to be found on lightweight authentication tokens. The main metrics we consider are run time and code size. We compare our results with a challenge-response protocol using an AES implementation optimized for the target platform. A major advantage of our protocol is its very small code size. The most compact implementation requires only about 460 bytes of code, which is an improvement by factor of about 10 over AES-based authentication. Given that EEPROM or FLASH memory is often one of the most precious resources on constrained devices, our protocol can be attractive in certain situations. The drawback of our protocol over AES on the target platform is an increase in clock cycles for one round of authentication. However, if we have access to a few hundred bytes of non-volatile data memory, our protocol allows precomputations which make the on-line phase only a factor two or three slower than AES. But even without precomputations, the protocol can still be executed in a few 100 msec, which will be sufficient for many real-world applications, e.g. remote keyless entry systems or authentication for financial transactions. Table 1 gives a summary of the results, see Section 5 for details.

We would like to stress at this point that our protocol is targeting lightweight tags that are equipped with (small) CPUs. For ultra constrained tokens (such as RFIDs in the price range of a few cents targeting the EPC market) which consist nowadays of a small integrated circuit, even compact AES implementations are often considered too costly. (We note that virtually all current commercially available low-end RFIDs do not have any crypto implemented.) However, tokens which use small microcontrollers are far more common, e.g., low-cost smart cards, and they do often require strong authentication. Also, it can be speculated that computational RFIDs such as the WISP [Wik] will become more common in the future, and hence software-friendly authentication methods that are highly efficient such as the protocol provided here will be needed.

1.2 LPN, Ring-LPN, and Related Problems

The security of our protocols relies on the new Ring Learning Parity with Noise (Ring-LPN) problem which is a natural extension of the standard Learning Parity with Noise (LPN) problem to rings. It can also be seen as a particular instantiation of the Ring-LWE (Learning with Errors over Rings) problem that was recently shown to have a strong connection to lattices [LPR10]. We will now briefly describe and compare these hardness assumptions, and we direct the reader to Section 3 for a formal definition of the Ring-LPN problem.

The decision versions of these problems require us to distinguish between two possible oracles to which we have black-box access. The first oracle has a randomly generated secret vector $s \in \mathsf{F}_2^n$ which it uses to produce its responses. In the LPN problem, each query to the oracle produces a uniformly random matrix[4] $A \in \mathsf{F}_2^{n \times n}$ and a vector $As + e = t \in \mathsf{F}_2^n$ where e is a vector in F_2^n each of whose entries is an independently generated Bernoulli random variable with probability of 1 being some public parameter τ between 0 and $1/2$. The second oracle in the LPN problem outputs a uniformly-random matrix $A \in \mathsf{F}_2^{n \times n}$ and a uniformly random vector $t \in \mathsf{F}_2^n$.

The only difference between LPN and Ring-LPN is in the way the matrix A is generated (both by the first and second oracle). While in the LPN problem, all its entries are uniform and independent, in the Ring-LPN problem, only its first column is generated uniformly at random in F_2^n. The remaining n columns of A depend on the first column and the underlying ring $\mathsf{R} = \mathsf{F}_2[X]/(f(X))$. If we view the first column of A as a polynomial $r \in \mathsf{R}$, then the i^{th} column (for $0 \leq i \leq n-1$) of A is just the vector representation of rX^i in the ring R. Thus when the oracle returns $As + e$, this corresponds to it returning the polynomial $r \cdot s + e$ where the multiplication of polynomials r and s (and the addition of e) is done in the ring R. The $\mathsf{Ring\text{-}LPN}^{\mathsf{R}}$ assumption states that it is hard to distinguish between the outputs of the first and the second oracle described above. In Section 3, we discuss how the choice of the ring R affects the security of the problem.

While the standard Learning Parity with Noise (LPN) problem has found extensive use as a cryptographic hardness assumption (e.g., [HB01,JW05,GRS08b,GRS08a,ACPS09,KSS10]), we are not aware of any constructions that employed the Ring-LPN problem. There have been some previous works that considered some relatively similar "structured" versions of LPN. The $\mathsf{HB}^\sharp$ authentication protocol of Gilbert et al. [GRS08a] made the assumption that for a random Toeplitz matrix $S \in \mathsf{F}_2^{m \times n}$, a uniformly random vector $a \in \mathsf{F}_2^n$, and a vector $e \in \mathsf{F}_2^m$ whose coefficients are distributed as Ber_τ, the output $(a, Sa + e)$ is computationally indistinguishable from (a, t) where t is uniform over F_2^m.

Another related work, as mentioned above, is the recent result of Lyubashevsky et al. [LPR10], where it is shown that solving the decisional Ring-LWE (Learning with Errors over Rings) problem is as hard as quantumly solving the worst case instances of the shortest vector problem in *ideal* lattices. The Ring-LWE problem is quite similar to Ring-LPN, with the main difference being that the ring R is defined as $\mathsf{F}_q[X]/(f(X))$ where $f(X)$ is a cyclotomic polynomial and q is a prime such that $f(X)$ splits completely into $deg(f(X))$ distinct factors over F_q.

Unfortunately, the security proof of our authentication scheme does not allow us to use a polynomial $f(X)$ that splits into low-degree factors, and so we cannot base our scheme on lattice problems. For a similar reason (see the proof of our scheme in Section 4 for more details), we cannot use samples that come from a Toeplitz matrix as in [GRS08a]. Nevertheless, we believe that the Ring-LPN assumption is very natural and will find further cryptographic applications, especially for constructions of schemes for low-cost devices.

[4] In the more common description of the LPN problem, each query to the oracle produces one random sample in F_2^n. For comparing LPN to Ring-LPN, however, it is helpful to consider the oracle as returning a matrix of n random independent samples on each query.

2 Definitions

2.1 Rings and Polynomials

For a polynomial $f(X)$ over F_2, we will often omit the indeterminate X and simply write f. The degree of f is denoted by $deg(f)$. For two polynomials a, f in $\mathsf{F}_2[X]$, $a \bmod f$ is defined to be the unique polynomial r of degree less than $deg(f)$ such that $a = fg + r$ for some polynomial $g \in \mathsf{F}_2[X]$. The elements of the ring $\mathsf{F}_2[X]/(f)$ will be represented by polynomials in $\mathsf{F}_2[X]$ of maximum degree $deg(f) - 1$. In this paper, we will only be considering rings $\mathsf{R} = \mathsf{F}_2[X]/(f)$ where the polynomial f factors into *distinct* irreducible factors over F_2. For an element a in the ring $\mathsf{F}_2[X]/(f)$, we will denote by $\widehat{a}$, the CRT (Chinese Remainder Theorem) representation of a with respect to the factors of f. In other words, if $f = f_1 \ldots f_m$ where all f_i are irreducible, then

$$\widehat{a} \doteq (a \bmod f_1, \ldots, a \bmod f_m).$$

If f is itself an irreducible polynomial, then $\widehat{a} = a$. Note that an element $\widehat{a} \in \mathsf{R}$ has a multiplicative inverse iff, for all $1 \leq i \leq m$, $a \neq 0 \bmod f_i$. We denote by R^* the set of elements in R that have a multiplicative inverse.

2.2 Distributions

For a distribution D over some domain, we write $r \overset{\$}{\leftarrow} D$ to denote that r is chosen according to the distribution D. For a domain Y, we write $U(Y)$ to denote the uniform distribution over Y. Let Ber_τ be the Bernoulli distribution over F_2 with parameter (bias) $\tau \in \,]0, 1/2[$ (i.e., $\Pr[x = 1] = \tau$ if $x \leftarrow \mathsf{Ber}_\tau$). For a polynomial ring $\mathsf{R} = \mathsf{F}_2[X]/(f)$, the distribution $\mathsf{Ber}_\tau^{\mathsf{R}}$ denotes the distribution over the polynomials of R, where each of the coefficients of the polynomial is drawn independently from Ber_τ. For a ring R and a polynomial $s \in \mathsf{R}$, we write $\Lambda_\tau^{\mathsf{R},s}$ to be the distribution over $\mathsf{R} \times \mathsf{R}$ whose samples are obtained by choosing a polynomial $r \overset{\$}{\leftarrow} U(\mathsf{R})$ and another polynomial $e \overset{\$}{\leftarrow} \mathsf{Ber}_\tau^{\mathsf{R}}$, and outputting $(r, rs + e)$.

2.3 Authentication Protocols

An authentication protocol Π is an interactive protocol executed between a Tag $\mathcal{T}$ and a reader $\mathcal{R}$, both PPT algorithms. Both hold a secret x (generated using a key-generation algorithm KG executed on the security parameter λ in unary) that has been shared in an initial phase. After the execution of the authentication protocol, $\mathcal{R}$ outputs either accept or reject. We say that the protocol has completeness error ε_c if for all $\lambda \in \mathbb{N}$, all secret keys x generated by $\mathsf{KG}(1^\lambda)$, the honestly executed protocol returns reject with probability at most ε_c. We now define different security notions of an authentication protocol.

PASSIVE ATTACKS. An authentication protocol is secure against *passive* attacks, if there exists no PPT adversary $\mathcal{A}$ that can make the reader $\mathcal{R}$ return accept with non-negligible probability after (passively) observing any number of interactions between reader and tag.

ACTIVE ATTACKS. A stronger notion for authentication protocols is security against *active* attacks. Here the adversary $\mathcal{A}$ runs in two stages. First, she can interact with the honest tag a polynomial number of times (with concurrent executions allowed). In the second phase $\mathcal{A}$ interacts with the reader only, and wins if the reader returns accept. Here we only give the adversary one shot to convince the verifier.[5] An authentication protocol is (t, q, ε)-*secure against active adversaries* if every PPT $\mathcal{A}$, running in time at most t and making q queries to the honest reader, has probability at most ε to win the above game.

[5] By using a hybrid argument one can show that this implies security even if the adversary can interact in $k \geq 1$ independent instances concurrently (and wins if the verifier accepts in at least one instance). The use of the hybrid argument looses a factor of k in the security reduction.

3 Ring-LPN and its Hardness

The decisional $\mathsf{Ring\text{-}LPN}^{\mathsf{R}}$ (Ring Learning Parity with Noise in ring R) assumption, formally defined below, states that it is hard to distinguish uniformly random samples in $\mathsf{R} \times \mathsf{R}$ from those sampled from $\Lambda_\tau^{\mathsf{R},s}$ for a uniformly chosen $s \in \mathsf{R}$.

Definition 1 ($\mathsf{Ring\text{-}LPN}^{\mathsf{R}}$)**.** *The (decisional)* $\mathsf{Ring\text{-}LPN}_\tau^{\mathsf{R}}$ *problem is* (t, q, ε)*-hard if for every distinguisher* $\mathcal{D}$ *running in time* t *and making* q *queries,*

$$\left|\Pr\left[s \xleftarrow{\$} \mathsf{R} \,:\, \mathcal{D}^{\Lambda_\tau^{\mathsf{R},s}} = 1\right] - \Pr\left[\mathcal{D}^{U(\mathsf{R}\times\mathsf{R})} = 1\right]\right| \leq \varepsilon.$$

3.1 Hardness of LPN and Ring-LPN

One can attempt to solve $\mathsf{Ring\text{-}LPN}$ using standard algorithms for LPN, or by specialized algorithms that possibly take advantage of $\mathsf{Ring\text{-}LPN}$'s additional structure. Some work towards constructing the latter type of algorithm has recently been done by Hanrot et al. [HLPS11], who show that in certain cases, the algebraic structure of the $\mathsf{Ring\text{-}LPN}$ and $\mathsf{Ring\text{-}LWE}$ problems makes them vulnerable to certain attacks. These attacks essentially utilize a particular relationship between the factorization of the polynomial $f(X)$ and the distribution of the noise.

$\mathsf{Ring\text{-}LPN}$ with an irreducible $f(X)$ When $f(X)$ is irreducible over F_2, the ring $\mathsf{F}_2[X]/(f)$ is a field. For such rings, the algorithm of Hanrot et al. does not apply, and we do not know of any other algorithm that takes advantage of the added algebraic structure of this particular $\mathsf{Ring\text{-}LPN}$ instance. Thus to the best of our knowledge, the most efficient algorithms for solving this problem are the same ones that are used to solve LPN, which we will now very briefly recount.

The computational complexity of the LPN problem depends on the length of the secret n and the noise distribution Ber_τ. Intuitively, the larger the n and the closer τ is to $1/2$, the harder the problem becomes. Usually the LPN problem is considered for constant values of τ somewhere between 0.05 and 0.25. For such constant τ, the fastest asymptotic algorithm for the LPN problem, due to Blum et al. [BKW03], takes time $2^{\Omega(n/\log n)}$ and requires approximately $2^{\Omega(n/\log n)}$ samples from the LPN oracle. If one has access to fewer samples, then the algorithm will perform somewhat worse. For example, if one limits the number of samples to only polynomially-many, then the algorithm has an asymptotic complexity of $2^{\Omega(n/\log\log n)}$ [Lyu05]. In our scenario, the number of samples available to the adversary is limited to n times the number of executions of the authentication protocol, and so it is reasonable to assume that the adversary will be somewhat limited in the number of samples he is able to obtain (perhaps at most 2^{40} samples), which should make our protocols harder to break than solving the $\mathsf{Ring\text{-}LPN}$ problem. Levieil and Fouque [LF06] made some optimizations to the algorithm of Blum et al. and analyzed its precise complexity. To the best of our knowledge, their algorithm is currently the most efficient one and we will refer to their results when analyzing the security of our instantiations.

In Section 5, we base our scheme on the hardness of the $\mathsf{Ring\text{-}LPN}^{\mathsf{R}}$ problem where the ring is $\mathsf{R} = \mathsf{F}_2[X]/(X^{532} + X + 1)$ and $\tau = 1/8$. According to the analysis of [LF06], an LPN problem of dimension 512 with $\tau = 1/8$ would require 2^{77} memory (and thus at least that much time) to solve when given access to approximately as many samples (see [LF06, Section 5.1]). Since our dimension is somewhat larger and the number of samples will be limited in practice, it is reasonable to assume that this instantiation has 80-bit security.

$\mathsf{Ring\text{-}LPN}$ with a reducible $f(X)$ For efficiency purposes, it is sometimes useful to consider using a polynomial $f(X)$ that is not irreducible over F_2. This will allow us to use the CRT representation of the elements of $\mathsf{F}_2[X]/(f)$ to perform multiplications, which in practice turns out to be more efficient. Ideally, we would like the polynomial f to split into as many small-degree polynomials f_i as possible, but there are some constraints that are placed on the factorization of f both by the security proof, and the possible weaknesses that a splittable polynomial introduces into the Ring-LPN problem.

If the polynomial f splits into $f = \prod_{i=1}^{m} f_i$, then it may be possible to try and solve the Ring-LPN problem modulo some f_i rather than modulo f. Since the degree of f_i is smaller than the degree of f, the resulting Ring-LPN problem may end up being easier. In particular, when we receive a sample $(r, rs+e)$ from the distribution $\Lambda_\tau^{\mathsf{R},s}$, we can rewrite it in CRT form as

$$(\widehat{r}, \widehat{rs+e}) = ((r \bmod f_1, rs + e \bmod f_1), \ldots, \\ (r \bmod f_m, rs + e \bmod f_m)),$$

and thus for every f_i, we have a sample

$$(r \bmod f_i, (r \bmod f_i)(s \bmod f_i) + e \bmod f_i),$$

where all the operations are in the ring (or field) $\mathsf{F}_2[X]/(f_i)$. Thus solving the (decision) Ring-LPN problem in $\mathsf{F}_2[X]/(f)$ reduces to solving the problem in $\mathsf{F}_2[X]/(f_i)$. The latter problem is in a smaller dimension, since $deg(s) > deg(s \bmod f_i)$, but the error distribution of $(e \bmod f_i)$ is quite different than that of e. While each coefficient of e is distributed independently as Ber_τ, each coefficient of $(e \bmod f_i)$ is distributed as the distribution of a sum of certain coefficients of e, and therefore the new error is larger.[6] Exactly which coefficients of e, and more importantly, how many of them, combine to form every particular coefficient of e' depends on the polynomial f_i. For example, if

$$f(X) = (X^3 + X + 1)(X^3 + X^2 + 1)$$

and $e = \sum\limits_{i=0}^{5} e_i X^i$, then,

$$e' = e \bmod (X^3 + X + 1) = (e_0 + e_3 + e_5) + (e_1 + e_3 + e_4 + e_5)X + (e_2 + e_4 + e_5)X^2,$$

and thus every coefficient of the error e' is comprised of at least 3 coefficients of the error vector e, and thus $\tau' > \frac{1}{2} - \frac{(1-2\tau)^3}{2}$.

In our instantiation of the scheme with a reducible $f(X)$ in Section 5, we used the $f(X)$ such that it factors into f_i's that make the operations in CRT form relatively fast, while making sure that the resulting Ring-LPN problem modulo each f_i is still around 2^{80}-hard.

4 Authentication Protocol

In this section we describe our new 2-round authentication protocol and prove its active security under the hardness of the Ring-LPN problem. Detailed implementation details will be given in Section 5.

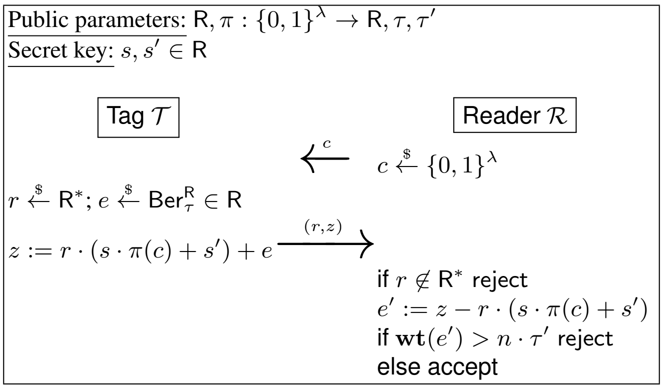
4.1 The Protocol

Our authentication protocol is defined over the ring $\mathsf{R} = \mathsf{F}_2[X]/(f)$ and involves a "suitable" mapping $\pi : \{0,1\}^\lambda \to \mathsf{R}$. We call π *suitable* for ring R if for all $c, c' \in \{0,1\}^\lambda$, $\pi(c) - \pi(c') \in \mathsf{R} \setminus \mathsf{R}^*$ iff $c = c'$. We will discuss the necessity and existence of such mappings after the proof of Theorem 1

- Public parameters. The authentication protocol has the following public parameters, where τ, τ' are constants and n depend on the security parameter λ.

R, n	ring $\mathsf{R} = \mathsf{F}_2[X]/(f)$, $\deg(f) = n$
$\pi : \{0,1\}^\lambda \to \mathsf{R}$	mapping
$\tau \in (0, 1/2)$	parameter of Bernoulli distribution
$\tau' \in (\tau, 1/2)$	acceptance threshold

[6] If we have k elements $e_1, \ldots, e_k \xleftarrow{\$} \mathsf{Ber}_\tau$, then a simple calculation shows that the element $e' = e_1 + \ldots + e_k$ is distributed as $\mathsf{Ber}_{\tau'}$ where $\tau' = \frac{1}{2} - \frac{(1-2\tau)^k}{2}$.



Public parameters: $\mathsf{R}, \pi : \{0,1\}^\lambda \to \mathsf{R}, \tau, \tau'$

Secret key: $s, s' \in \mathsf{R}$

Tag $\mathcal{T}$ | Reader $\mathcal{R}$

$\xleftarrow{c}$ $c \overset{\$}{\leftarrow} \{0,1\}^\lambda$

$r \overset{\$}{\leftarrow} \mathsf{R}^*; e \overset{\$}{\leftarrow} \mathsf{Ber}_\tau^{\mathsf{R}} \in \mathsf{R}$

$z := r \cdot (s \cdot \pi(c) + s') + e \xrightarrow{(r,z)}$

if $r \notin \mathsf{R}^*$ reject

$e' := z - r \cdot (s \cdot \pi(c) + s')$

if $\mathbf{wt}(e') > n \cdot \tau'$ reject

else accept

Fig. 1. Two-round authentication protocol with active security from the $\mathsf{Ring\text{-}LPN}^{\mathsf{R}}$ assumption.

- Key Generation. Algorithm $\mathsf{KG}(1^\lambda)$ samples $s, s' \overset{\$}{\leftarrow} \mathsf{R}$ and returns s, s' as the secret key.
- Authentication Protocol. The Reader $\mathcal{R}$ and the Tag $\mathcal{T}$ share secret value $s, s' \in \mathsf{R}$. To be authenticated by a Reader, the Tag and the Reader execute the authentication protocol from Figure 1.

4.2 Analysis

For our analysis we define for $x, y \in]0,1[$ the following constant:

$$c(x,y) := \left(\frac{x}{y}\right)^x \left(\frac{1-x}{1-y}\right)^{1-x} .$$

We now state that our protocol is secure against active adversaries. Recall that active adversaries can arbitrarily interact with a Tag oracle in the first phase and tries to impersonate the Reader in the 2nd phase.

Theorem 1. *If ring mapping π is suitable for ring R and the $\mathsf{Ring\text{-}LPN}_{\mathsf{R}}$ problem is (t, q, ε)-hard then the authentication protocol from Figure 1 is (t', q, ε')-secure against active adversaries, where*

$$t' = t - q \cdot \exp(R) \qquad \varepsilon' = \varepsilon + q \cdot 2^{-\lambda} + c(\tau', 1/2)^{-n} \tag{4.1}$$

and $\exp(R)$ is the time to perform $O(1)$ exponentiations in R. Furthermore, the protocol has completeness error $\varepsilon_{\mathrm{c}}(\tau, \tau', n) \approx c(\tau', \tau)^{-n}$.

Proof. The completeness error $\varepsilon_{\mathrm{c}}(\tau, \tau', n)$ is (an upper bound on) the probability that an honestly generated Tag gets rejected. In our protocol this is exactly the case when the error e' has weight $\geq n \cdot \tau'$, i.e.

$$\varepsilon_{\mathrm{c}}(\tau, \tau', n) = \Pr[\mathbf{wt}(e') > n \cdot \tau' \ : \ e \overset{\$}{\leftarrow} \mathsf{Ber}_\tau^{\mathsf{R}}]$$

Levieil and Fouque [LF06] show that one can approximate this probability as $\varepsilon_{\mathrm{c}} \approx c(\tau', \tau)^{-n}$.

To prove the security of the protocol against active attacks we proceed in sequences of games. Game_0 is the security experiment describing an active attack on our scheme by an adversary $\mathcal{A}$ making q queries and running in time t', i.e.

- Sample the secret key $s, s' \overset{\$}{\leftarrow} \mathsf{R}$.
- (1st phase of active attack) $\mathcal{A}$ queries the tag $\mathcal{T}$ on $c \in \{0,1\}^\lambda$ and receives (r, z) computed as illustrated in Figure 1.
- (2nd phase of active attack) $\mathcal{A}$ gets a random challenge $c^* \overset{\$}{\leftarrow} \{0,1\}^\lambda$ and outputs (r, z). $\mathcal{A}$ wins if the reader $\mathcal{R}$ accepts, i.e. $\mathbf{wt}(z - r \cdot (s \cdot \pi(c^*) + s')) \leq n \cdot \tau'$.

By definition we have $\Pr[\mathcal{A} \text{ wins in } \mathsf{Game}_0] \leq \varepsilon'$.

Game_1 is as Game_0, except that all the values (r, z) returned by the Tag oracle in the first phase (in return to a query $c \in \{0,1\}^\lambda$) are uniform random elements $(r, z) \in \mathsf{R}^2$. We now show that if $\mathcal{A}$ is successful against Game_0, then it will also be successful against Game_1.

Claim. $|\Pr[\mathcal{A} \text{ wins in } \mathsf{Game}_1] - \Pr[\mathcal{A} \text{ wins in } \mathsf{Game}_0]| \leq \varepsilon + q \cdot 2^{-\lambda}$

To prove this claim, we construct an adversary $\mathcal{D}$ (distinguisher) against the Ring-LPN problem which runs in time $t = t' + \exp(\mathsf{R})$ and has advantage

$$\varepsilon \geq |\Pr[\mathcal{A} \text{ wins in } \mathsf{Game}_1] - \Pr[\mathcal{A} \text{ wins in } \mathsf{Game}_0]| - q \cdot 2^{-\lambda}$$

$\mathcal{D}$ has access to a Ring-LPN oracle $\mathcal{O}$ and has to distinguish between $\mathcal{O} = \Lambda_\tau^{\mathsf{R},s}$ for some secret $s \in \mathsf{R}$ and $\mathcal{O} = U(\mathsf{R} \times \mathsf{R})$.

- $\mathcal{D}$ picks a random challenge $c^* \xleftarrow{\$} \{0,1\}^\lambda$ and $a \xleftarrow{\$} \mathsf{R}$. Next, it runs $\mathcal{A}$ and simulates its view with the unknown secret s, s', where $s \in \mathsf{R}$ comes from the oracle $\mathcal{O}$ and s' is implicitly defined as $s' := -\pi(c^*) \cdot s + a \in \mathsf{R}$.
- In the 1st phase, $\mathcal{A}$ can make q (polynomial many) queries to the Tag oracle. On query $c \in \{0,1\}^\lambda$ to the Tag oracle, $\mathcal{D}$ proceeds as follows. If $\pi(c) - \pi(c^*) \notin \mathsf{R}^*$, then abort. Otherwise, $\mathcal{D}$ queries its oracle $\mathcal{O}()$ to obtain $(r', z') \in \mathsf{R}^2$. Finally, $\mathcal{D}$ returns (r, z) to $\mathcal{A}$, where
$$r := r' \cdot (\pi(c) - \pi(c^*))^{-1}, \quad z := z' + ra. \tag{4.2}$$
- In the 2nd phase, $\mathcal{D}$ uses $c^* \in \{0,1\}^\lambda$ to challenge $\mathcal{A}$. On answer (r, z), $\mathcal{D}$ returns 0 to the Ring-LPN game if $\mathbf{wt}(z - r \cdot a) > n \cdot \tau'$ or $r \notin \mathsf{R}^*$, and 1 otherwise. Note that $s\pi(c^*) + s' = (\pi(c^*) - \pi(c^*))s + a = a$ and hence the above check correctly simulates the output of a reader with the simulated secret s, s'.

Note that the running time of $\mathcal{D}$ is that of $\mathcal{A}$ plus $O(q)$ exponentiations in R.

Let bad be the event that for at least one query c made by $\mathcal{A}$ to the Tag oracle, we have that $\pi(c) - \pi(c^*) \notin \mathsf{R}^*$. Since c^* is uniform random in R and hidden from $\mathcal{A}$'s view in the first phase we have by the union bound over the q queries

$$\begin{aligned} \Pr[\mathsf{bad}] &\leq q \cdot \Pr_{c^* \in \{0,1\}^\lambda}[\pi(c) - \pi(c^*) \in \mathsf{R} \setminus \mathsf{R}^*] \\ &= q \cdot 2^{-\lambda}. \end{aligned} \tag{4.3}$$

The latter inequality holds because π is suitable for R.

Let us now assume bad does not happen. If $\mathcal{O} = \Lambda_\tau^{\mathsf{R},s}$ is the real oracle (i.e., it returns (r', z') with $z' = r's + e$) then by the definition of (r, z) from (4.2),

$$z = (r's + e) + ra = r(\pi(c)s - \pi(c^*)s + a) + e = r(s\pi(c) + s') + e.$$

Hence the simulation perfectly simulates $\mathcal{A}$'s view in Game_0. If $\mathcal{O} = U(\mathsf{R} \times \mathsf{R})$ is the random oracle then (r, z) are uniformly distributed, as in Game_1. That concludes the proof of Claim 4.2.

We next upper bound the probability that $\mathcal{A}$ can be successful in Game_1. This bound will be information theoretic and even holds if $\mathcal{A}$ is computationally unbounded and can make an unbounded number of queries in the 1st phase. To this end we introduce the minimal soundness error, $\varepsilon_{\mathrm{ms}}$, which is an upper bound on the probability that a tag (r, z) chosen independently of the secert key is valid, i.e.

$$\varepsilon_{\mathrm{ms}}(\tau', n) := \max_{(z,r) \in \mathsf{R} \times \mathsf{R}^*} \Pr_{s,s' \xleftarrow{\$} \mathsf{R}} [\mathbf{wt}(\underbrace{z - r \cdot (s \cdot \pi(c^*) + s')}_{e'}) \leq n\tau']$$

As $r \in \mathsf{R}^*$ and $s' \in \mathsf{R}$ is uniform, also $e' = z - r \cdot (s \cdot \pi(c^*) + s'$ is uniform, thus $\varepsilon_{\mathrm{ms}}$ is simply

$$\varepsilon_{\mathrm{ms}}(\tau', n) := \Pr_{e' \xleftarrow{\$} \mathsf{R}} [\mathbf{wt}(e') \leq n\tau']$$

Again, it was shown in [LF06] that this probability can be approximated as

$$\varepsilon_{\mathrm{ms}}(\tau', n) \approx c(\tau', 1/2)^{-n}. \tag{4.4}$$

Clearly, $\varepsilon_{\mathrm{ms}}$ is a trivial lower bound on the advantage of $\mathcal{A}$ in forging a valid tag, by the following claim in Game_1 one cannot do any better than this.

Claim. $\Pr[\mathcal{A} \text{ wins in } \mathsf{Game}_1] = \varepsilon_{\mathrm{ms}}(\tau', n)$

To see that this claim holds one must just observe that the answers $\mathcal{A}$ gets in the first phase of the active attack in Game_1 are independent of the secret s, s'. Hence $\mathcal{A}$'s advantage is $\varepsilon_{\mathrm{ms}}(\tau', n)$ by definition.

Claims 4.2 and 4.2 imply (4.1) and conclude the proof of Theorem 1.

We require the mapping $\pi : \{0,1\}^\lambda \to \mathsf{R}$ used in the protocol to be *suitable* for R, i.e. for all $c, c' \in \{0,1\}^\lambda$, $\pi(c) - \pi(c') \in \mathsf{R} \setminus \mathsf{R}^*$ iff $c = c'$. In Section 5 we describe efficient suitable maps for any $\mathsf{R} = \mathsf{F}_2[X]/(f)$ where f has no factor of degree $\leq \lambda$. This condition is necessary, as no suitable mapping exists if f has a factor f_i of degree $\leq \lambda$: in this case, by the pigeonhole principle, there exist distinct $c, c' \in \{0,1\}^\lambda$ such that $\pi(c) = \pi(c') \mod f_i$, and thus $\pi(c) - \pi(c') \in \mathsf{R} \setminus \mathsf{R}^*$.

We stress that for our security proof we need π to be suitable for R, since otherwise (4.3) is no longer guaranteed to hold. It is an interesting question if this is inherent, or if the security of our protocol can be reduced to the $\mathsf{Ring\text{-}LPN}^{\mathsf{R}}$ problem for arbitrary rings $\mathsf{R} = \mathsf{F}_2[X]/(f)$, or even $\mathsf{R} = \mathsf{F}_q[X]/(f)$ (This is interesting since, if f has factors of degree $\ll \lambda$, the protocol could be implemented more efficiently and even become based on the worst-case hardness of lattice problems). Similarly, it is unclear how to prove security of our protocol instantiated with Toeplitz matrices.

5 Implementation

There are two objectives that we pursue with the implementation of our protocol. First, we will show that the protocol is in fact practical with concrete parameters, even on extremely constrained CPUs. Second, we investigate possible application scenarios where the protocol might have additional advantages. From a practical point of view, we are particularly interested in comparing our protocol to classical symmetric challenge-response schemes employing AES. Possible advantages of the protocol at hand are (i) the security properties and (ii) improved implementation properties. With respect to the former aspect, our protocol has the obvious advantage of being provably secure under a reasonable and static hardness assumption. Even though AES is arguably the most trusted symmetric cipher, it is “merely” computationally secure with respect to known attacks.

In order to investigate implementation properties, constrained microprocessors are particularly relevant. We chose an 8-bit AVR ATmega163 [Atm] based smartcard, which is widely used in myriads of embedded applications. It can be viewed as a typical representative of a CPU used in tokens that are in need for an authentication protocol, e.g., computational RFID tags or (contactless) smart cards. The main metrics we consider for the implementation are run-time and code size. We note at this point that in many lightweight crypto applications, code size is the most precious resource once the run-time constraints are fulfilled. This is due to the fact that EEPROM or flash memory is often heavily constrained. For instance, the WISP, a computational RFID tag, has only 8 kBytes of program memory [Wik,MSP].

We implemented two variants of the protocol described in Section 4. The first variant uses a ring $\mathsf{R} = \mathsf{F}_2[X]/(f)$, where f splits into five irreducible polynomials; the second variant uses a field, i.e., f is irreducible. For both implementations, we chose parameters which provide a security level of $\lambda = 80$ bits, i.e., the parameters are chosen such that ε' in (4.1) is bounded by 2^{-80} and the completeness ε_{c} is bounded by 2^{-40}. This security level is appropriate for the lightweight applications which we are targeting.

5.1 Implementation with a Reducible Polynomial

From an implementation standpoint, the case of reducible polynomial is interesting since one can take advantage of arithmetic based on the Chinese Remainder Theorem.

PARAMETERS. To define the ring $\mathsf{R} = \mathsf{F}_2[X]/(f)$, we chose the reducible polynomial f to be the product of the $m = 5$ irreducible pentanomials specified by the following powers with non-zero coefficients: $(127, 8, 7, 3, 0)$, $(126, 9, 6, 5, 0)$, $(125, 9, 7, 4, 0)$, $(122, 7, 4, 3, 0)$, $(121, 8, 5, 1, 0)$[7]. Hence f is a polynomial of degree $n = 621$. We chose $\tau = 1/6$ and $\tau' = .29$ to obtain minimal soundness error $\varepsilon_{\mathrm{ms}} \approx c(\tau', 1/2)^{-n} \leq 2^{-82}$ and completeness error $\varepsilon_{\mathrm{c}} \leq 2^{-42}$. From the discussion of Section 3 the best known attack on $\mathsf{Ring\text{-}LPN}_\tau^{\mathsf{R}}$ with the above parameters has complexity $> 2^{80}$. The mapping $\pi : \{0,1\}^{80} \to \mathsf{R}$ is defined as follows. On input $c \in \{0,1\}^{80}$, for each $1 \leq i \leq 5$, pad $c \in \{0,1\}^{80}$ with $\deg(f_i) - 80$ zeros and view the result as coefficients of an element $v_i \in \mathsf{F}_2[X]/(f_i)$. This defines $\pi(c) = (v_1, \ldots, v_5)$ in CRT representation. Note that, for fixed $c, c^* \in \{0,1\}^{80}$, we have that $\pi(c) - \pi(c^*) \in \mathsf{R} \setminus \mathsf{R}^*$ iff $c = c^*$ and hence π is *suitable* for R.

IMPLEMENTATION DETAILS. The main operations are multiplications and additions of polynomials that are represented by 16 bytes. We view the CRT-based multiplication in three stages. In the first stage, the operands are reduced modulo each of the five irreducible polynomials. This part has a low computational complexity. Note that only the error e has to be chosen in the ring and afterwards transformed to CRT representation. It is possible to save the secret key (s, s') and to generate r directly in the CRT representation. This is not possible for e because e has to come from $\mathsf{Ber}_\tau^{\mathsf{R}}$. In the second stage, one multiplication in each of the finite fields defined by the five pentanomials has to be performed. We used the right-to-left comb multiplication algorithm from [HMV03]. For the multiplication with $\pi(c)$ we exploit the fact that only the first 80 coefficients can be non-zero. Hence we wrote one function for *normal* multiplication and one for *sparse* multiplication. The latter is more than twice as fast as the former. The subsequent reduction takes care of the special properties of the pentanomials, thus code reuse is not possible for the different fields. The third stage, constructing the product polynomial in the ring, is shifted to the prover (RFID reader) which normally has more computational power than the tag $\mathcal{T}$. Hence the response (r, z) is sent in CRT form to the reader. If non-volatile storage — in our case we need $2 \cdot 5 \cdot 16 = 160$ bytes — is available we can heavily reduce the response time of the tag. At an arbitrary point in time, choose e and r according to their distribution and precompute $\mathit{tmp}_1 = r \cdot s$ and $\mathit{tmp}_2 = r \cdot s' + e$. When a challenge c is received afterwards, tag $\mathcal{T}$ only has to compute $z = \mathit{tmp}_1 \cdot \pi(c) + \mathit{tmp}_2$. Because $\pi(c)$ is sparse, the tag can use the *sparse* multiplication and response very quickly. The results of the implementation are shown in Table 2 in Section 5.3. Note that all multiplication timings given already include the necessary reductions and addition of a value according to Figure 1.

5.2 Implementation with an Irreducible Polynomial

PARAMETERS. To define the field $\mathsf{F} = \mathsf{F}_2[X]/(f)$, we chose the irreducible trinomial $f(X) = X^{532} + X + 1$ of degree $n = 532$. We chose $\tau = 1/8$ and $\tau' = .27$ to obtain minimal soundness error $\varepsilon_{\mathrm{ms}} \approx c(\tau', 1/2)^{-n} \leq 2^{-80}$ and completeness error $\varepsilon_{\mathrm{c}} \approx 2^{-55}$. From the discussion in Section 3 the best known attack on $\mathsf{Ring\text{-}LPN}_\tau^{\mathsf{F}}$ with the above parameters has complexity $> 2^{80}$. The mapping $\pi : \{0,1\}^{80} \to \mathsf{F}$ is defined as follows. View $c \in \{0,1\}^{80}$ as $c = (c_1, \ldots, c_{16})$ where c_i is a number between 1 and 32. Define the coefficients of the polynomial $v = \pi(c) \in \mathsf{F}$ as zero except all positions i of the form $i = 16 \cdot (j - 1) + c_j$, for some $j = 1, \ldots, 16$. Hence $\pi(c)$ is sparse, i.e., it has exactly 16 non-zero coefficients. Since π is injective and F is a field, the mapping π is suitable for F.

IMPLEMENTATION DETAILS. The main operation for the protocol is now a 67-byte multiplication. Again we used the right-to-left comb multiplication algorithm from [HMV03] and an optimized reduction algorithm. Like in the reducible case, the tag can do similar precomputations if $2 \cdot 67 = 134$ bytes non-volatile

[7] $(127, 8, 7, 3, 0)$ refers to the polynomial $X^{127} + X^8 + X^7 + X^3 + 1$.

storage are available. Because of the special type of the mapping $v = \pi(c)$, the gain of the *sparse* multiplication is even larger than in the reducible case. Here we are a factor of 7 faster, making the response time with precomputations faster, although the field is larger. The results are shown in Table 3 in Section 5.3.

5.3 Implementation Results

All results presented in this section consider only the clock cycles of the actual arithmetic functions. The communication overhead and the generation of random bytes is excluded because they occur in every authentication scheme, independent of the underlying cryptographic functions. The time for building e from $\mathsf{Ber}_\tau^{\mathsf{R}}$ out of the random bytes and converting it to CRT form is included in *Overhead*. Table 2 and Table 3 shows the results for the ring based and field based variant, respectively.

Table 2. Results for the ring based variant w/o precomputation

Aspect	**time** **in cycles**	**code size** **in bytes**
Overhead	$17,500$	264
Mul	$5 \times 13,000$	164
sparse Mul	$5 \times 6,000$	170
total	$112,500$	1356

The overall code size is not the sum of the other values because, as mentioned before, the same multiplication code is used for all *normal* and *sparse* multiplications, respectively, while the reduction code is different for every field ($\approx$ 134 byte each). The same code for reduction is used independently of the type of the multiplication for the same field. If precomputation is acceptable, the tag can answer the challenge after approximately $30,000$ clock cycles, which corresponds to a 15 msec if the CPU is clocked at 2 MHz.

Table 3. Results for the field based variant w/o precomputation

Aspect	**time** **in cycles**	**code size** **in bytes**
Overhead	$3,000$	150
Mul	$150,000$	161
sparse Mul	$21,000$	148
total	$174,000$	459

For the field-based protocol, the overall performance is slower due to the large operands used in the multiplication routine. But due to the special mapping $v = \pi(c)$, here the tag can do a sparse multiplications in only $21,000$ clocks cycles. This allows the tag to respond in 10.5 msec at 2 MHz clock rate if non-volatile storage is available.

As mentioned in the introduction, we want to compare our scheme with a conventional challenge-response authentication protocol based on AES. The tag's main operation in this case is one AES encryption. The implementation in [LLS09] states $8,980$ clock cycles for one encryption on a similar platform, but unfortunately no code size is given; [Tik] reports 10121 cycles per encryption and a code size of 4644 bytes.[8] In comparison with these highly optimized AES implementations, our scheme is around eleven times slower when using the ring based variant without precomputations. If non-volatile storage allows

[8] An internet source [Poe] claims to encrypt in 3126 cycles with code size of 3098 bytes but since this is unpublished material we do not consider it in our comparison.

precomputations, the ring based variant is only three times slower than AES. But the code size is by a factor of two to three smaller, making it attractive for Flash constrained devices. The field based variant without precomputations is 17 to 19 times slower than AES, but with precompuations it is only twice as slow as AES, while only consuming one tenths of the code size. From a practical point of view, it is important to note that even our slowest implementation is executed in less than 100 msec if the CPU is clocked at 2 MHz. This response time is sufficient in many application scenarios. (For authentications involving humans, a delay of 1 sec is often considered acceptable.)

The performance drawback compared to AES is not surprising, but it is considerably less dramatic compared to asymmetric schemes like RSA or ECC [GPW$^+$04]. But exploiting the special structure of the multiplications in our scheme and using only a small amount of non-volatile data memory provides a response time in the same order of magnitude as AES, while keeping the code size much smaller.

6 Conclusions and open Problems

We proposed a new [KPC$^+$11]-like authentication protocol with provable security against active attacks based on the Ring-LPN assumption, consisting of only two rounds, and having small communication complexity. Furthermore, our implementations on an 8-bit AVR ATmega163 based smartcard demonstrated that it has very small code size and its efficiency can be of the same order as traditional AES-based authentication protocols. Overall, we think that its features make it very applicable in scenarios that involve low-cost, resource-constrained devices.

Our protocol cannot be proved secure against man-in-the-middle (MIM) attacks, but using a recent transformation from [DKPW12] we can get a MIM secure scheme with small extra cost (one application of a universal hash function.) Still, finding a more direct construction which achieves MIM security (or proving that the current protocol already has this property) but doesn't require any hashing remains an interesting open problem.

We believe that the Ring-LPN assumption is very natural and will find further cryptographic applications, especially for constructions of schemes for low-cost devices. In particular, we think that if the LPN-based line of research is to lead to a practical protocol in the future, then the security of this protocol will be based on a hardness assumption with some "extra algebraic structure", such as Ring-LPN in this work, or LPN with Toeplitz matrices in the work of Gilbert et al. [GRS08a]. More research, however, needs to be done on understanding these problems and their computational complexity. In terms of Ring-LPN, it would be particularly interesting to find out whether there exists an equivalence between the decision and the search versions of the problem similar to the reductions that exist for LPN [BFKL93,Reg09,KS06a] and Ring-LWE [LPR10].

7 Acknowledgements.

We would like to thank the anonymous referees of this conference and those of the *ECRYPT Workshop on Lightweight Cryptography* for very useful comments, and in particular for the suggestion that the scheme is somewhat vulnerable to a man-in-the-middle attack whenever an adversary observes two reader challenges that are the same. We hope that the attack we described in Appendix A corresponds to what the reviewer had in mind. This work was partially supported by the European Research Council.

References

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai, *Fast cryptographic primitives and circular-secure encryption based on hard learning problems*, CRYPTO 2009 (Shai Halevi, ed.), LNCS, vol. 5677, Springer, August 2009, pp. 595–618.

[Atm] Atmel, *ATmega163 datasheet*, "`www.atmel.com/atmel/acrobat/doc1142.pdf`".

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton, *Cryptographic primitives based on hard learning problems*, CRYPTO, 1993, pp. 278–291.

[BKL$^+$07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe, *PRESENT: An ultra-lightweight block cipher*, CHES 2007 (Pascal Paillier and Ingrid Verbauwhede, eds.), LNCS, vol. 4727, Springer, September 2007, pp. 450–466.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman, *Noise-tolerant learning, the parity problem, and the statistical query model*, J. ACM **50** (2003), no. 4, 506–519.

[DKPW12] Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs, *Message authentication, revisited*, EUROCRYPT, 2012.

[DR02] Joan Daemen and Vincent Rijmen, *The design of rijndael: AES - the advanced encryption standard*, Springer, 2002.

[GPW$^+$04] Nils Gura, Arun Patel, Arvinderpal W, Hans Eberle, and Sheueling Chang Shantz, *Comparing elliptic curve cryptography and RSA on 8-bit CPUs*, Cryptographic Hardware and Embedded Systems - CHES 2004, 2004, pp. 119–132.

[GRS05] Henri Gilbert, Matt Robshaw, and Herve Sibert, *An active attack against HB+ – a provably secure lightweight authentication protocol*, Cryptology ePrint Archive, Report 2005/237, 2005, `http://eprint.iacr.org/`.

[GRS08a] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin, *HB$^\sharp$: Increasing the security and efficiency of HB$^+$*, EUROCRYPT 2008 (Nigel P. Smart, ed.), LNCS, vol. 4965, Springer, April 2008, pp. 361–378.

[GRS08b] ———, *How to encrypt with the LPN problem*, ICALP 2008, Part II (Luca Aceto, Ivan Damgard, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, eds.), LNCS, vol. 5126, Springer, July 2008, pp. 679–690.

[HB00] N. Hopper and M. Blum, *A secure human-computer authentication scheme*, Tech. Report CMU-CS-00-139, Carnegie Mellon University, 2000.

[HB01] Nicholas J. Hopper and Manuel Blum, *Secure human identification protocols*, ASIACRYPT 2001 (Colin Boyd, ed.), LNCS, vol. 2248, Springer, December 2001, pp. 52–66.

[HLPS11] Guillaume Hanrot, Vadim Lyubashevsky, Chris Peikert, and Damien Stehlé, *Personal communication*, 2011.

[HMV03] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone, *Guide to elliptic curve cryptography*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[JW05] Ari Juels and Stephen A. Weis, *Authenticating pervasive devices with human protocols*, CRYPTO 2005 (Victor Shoup, ed.), LNCS, vol. 3621, Springer, August 2005, pp. 293–308.

[KPC$^+$11] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi, *Efficient authentication from hard learning problems*, EUROCRYPT, 2011, pp. 7–26.

[KS06a] Jonathan Katz and Ji Sun Shin, *Parallel and concurrent security of the HB and HB+ protocols*, EUROCRYPT 2006 (Serge Vaudenay, ed.), LNCS, vol. 4004, Springer, May / June 2006, pp. 73–87.

[KS06b] Jonathan Katz and Adam Smith, *Analyzing the HB and HB+ protocols in the "large error" case*, Cryptology ePrint Archive, Report 2006/326, 2006, `http://eprint.iacr.org/`.

[KSS10] Jonathan Katz, Ji Sun Shin, and Adam Smith, *Parallel and concurrent security of the HB and HB+ protocols*, Journal of Cryptology **23** (2010), no. 3, 402–421.

[KW05] Ziv Kfir and Avishai Wool, *Picking virtual pockets using relay attacks on contactless smartcard*, Security and Privacy for Emerging Areas in Communications Networks, International Conference on **0** (2005), 47–58.

[KW06] Ilan Kirschenbaum and Avishai Wool, *How to build a low-cost, extended-range RFID skimmer*, Proceedings of the 15th USENIX Security Symposium (SECURITY 2006), USENIX Association, August 2006, pp. 43–57.

[LF06] Éric Levieil and Pierre-Alain Fouque, *An improved LPN algorithm*, SCN 06(Roberto De Prisco and Moti Yung, eds.), LNCS, vol. 4116, Springer, September 2006, pp. 348–359.

[LLS09] Hyubgun Lee, Kyounghwa Lee, and Yongtae Shin, *AES implementation and performance evaluation on 8-bit microcontrollers*, CoRR **abs/0911.0482** (2009).

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev, *On ideal lattices and learning with errors over rings*, EUROCRYPT 2010 (Henri Gilbert, ed.), LNCS, vol. 6110, Springer, May 2010, pp. 1–23.

[Lyu05] Vadim Lyubashevsky, *The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem*, APPROX-RANDOM, 2005, pp. 378–389.

[MSP] *MSP430 datasheeet.*

[OOV08] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay, *On the security of HB$^\#$ against a man-in-the-middle attack*, ASIACRYPT, 2008, pp. 108–124.

[Poe] B. Poettering, *AVRAES: The AES block cipher on AVR controllers*, "`http://point-at-infinity.org/avraes/`".

[Reg09] Oded Regev, *On lattices, learning with errors, random linear codes, and cryptography*, J. ACM **56** (2009), no. 6.

[Tik] Jeff Tikkanen, *AES implementation on AVR ATmega328p*, "`http://cs.ucsb.edu/\~koc/cs178/projects/JT/avr\_aes.html`".

[Wik] WISP Wiki, *WISP 4.0 DL hardware*, "`http://wisp.wikispaces.com/WISP+4.0+DL`".

A Man-in-the-Middle Attack

In this section, we sketch a man-in-the-middle attack against the protocol in Figure 1 that recovers the secret key in time approximately $O\left(n^{1.5} \cdot 2^{\lambda/2}\right)$ when the adversary is able to insert himself into that many valid interactions between the reader and the tag. For a ring $\mathsf{R} = \mathsf{F}_2[X]/(f)$ and a polynomial $g \in \mathsf{R}$, define the vector $\boldsymbol{g}$ to be a vector of dimension $deg(f)$ whose i^{th} coordinate is the X^i coefficient of g. Similarly, for a polynomial $h \in \mathsf{R}$, let $Rot(h)$ be a $deg(f) \times deg(f)$ matrix whose i^{th} column (for $0 \leq i < deg(f)$) is $\overrightarrow{h \cdot X^i}$, or in other words, the coefficients of the polynomial $h \cdot X^i$ in the ring R. From this description, one can check that for two polynomials $g, h \in \mathsf{R}$, the product $\overrightarrow{g \cdot h} = Rot(g) \cdot \boldsymbol{h} \bmod 2 = Rot(h) \cdot \boldsymbol{g} \bmod 2$.

We now move on to describing the attack. The i^{th} (successful) interaction between a reader $\mathcal{R}$ and a tag $\mathcal{T}$ consists of the reader sending the challenge c_i, and the tag replying with the pair (r_i, z_i) where $z_i - r_i \cdot (s \cdot \pi(c_i) + s')$ is a low-weight polynomial of weight at most $n \cdot \tau'$. The adversary who is observing this interaction will forward the challenge c_i untouched to the tag, but reply to the reader with the ordered pair $(r_i, z_i' = z_i + e_i)$ where e_i is a vector that is strategically chosen with the hope that the vector $z_i' - r_i \cdot (s \cdot \pi(c_i) + s')$ is *exactly* of weight $n \cdot \tau'$. It's not hard to see that it's possible to choose such a vector e_i so that the probability of $z_i' - r_i \cdot (s \cdot \pi(c_i) + s')$ being of weight $n \cdot \tau'$ is approximately $1/\sqrt{n}$. The response (r_i, z_i') will still be valid, and so the reader will accept. By the birthday bound, after approximately $2^{\lambda/2}$ interactions, there will be a challenge c_j that is equal to some previous challenge c_i. In this case, the adversary replies to the reader with (r_i, z_i''), where the polynomial z_i'' is just the polynomial z_i' whose first bit (i.e. the constant coefficient) is flipped. What the adversary is hoping for is that the reader accepted the response (r_i, z_i') but rejects (r_i, z_i''). Notice that the only way this can happen is if the first bit of z_i' is equal to the first bit of $r_i \cdot (s \cdot \pi(c_i) + s')$, and thus flipping it, increases the error by 1 and makes the reader reject. We now explain how finding such a pair of responses can be used to recover the secret key.

Since the polynomial expression $z_i' - r_i \cdot (s \cdot \pi(c_i) + s') = z_i' - r_i \cdot \pi(c_i) \cdot s - r_i \cdot s'$ can be written as matrix-vector multiplications as

$$\boldsymbol{z_i'} - Rot(r_i \cdot \pi(c_i)) \cdot \boldsymbol{s} - Rot(r_i) \cdot \boldsymbol{s'} \bmod 2,$$

if we let the first bit of $\boldsymbol{z_i'}$ be β_i, the first row of $Rot(r_i \cdot \pi(c_i))$ be $\boldsymbol{a_i}$ and the first row of $Rot(r_i)$ be $\boldsymbol{b_i}$, then we obtain the linear equation

$$\langle \boldsymbol{a_i}, \boldsymbol{s} \rangle + \langle \boldsymbol{b_i}, \boldsymbol{s'} \rangle = \beta_i.$$

To recover the entire secret s, s', the adversary needs to repeat the above attack until he obtains $2n$ linearly-independent equations (which can be done with $O(n)$ successful attacks), and then use Gaussian elimination to recover the full secret.