A Sub-0.5V Lattice-Based Public-Key Encryption Scheme for RFID Platforms in 130nm CMOS

Yu Yao, Jiawei Huang, Sudhanshu Khanna, abhi shelat,
Benton Highsmith Calhoun, John Lach, and David Evans

University of Virginia
{yy4y,jh3wn,sk4fs,as4bd,bcalhoun,jlach,evans}@virginia.edu

Abstract. Implementing public-key cryptography on passive RFID tags is very challenging due to the limited die size and power available. Typical public-key algorithms require complex logical components such as modular exponentiation in RSA. We demonstrate the feasibility of implementing public-key encryption on low-power, low cost passive RFID tags to large-scale private identification. We use Oded Regev's Learning-With-Error (LWE) cryptosystem, which is provably secure under the hardness assumption of classic lattice problems. The advantage of using the LWE cryptosystem is its intrinsic computational simplicity (the main operation is modular addition). We leverage the low speed of RFID application by using circuit design with supply voltage close to transistor threshold (V_t) to lower power. This paper presents protocols for using the LWE cipher to provide private identification, evaluates a design for implementing those protocols on passive RFID tags, and reports on simulation experiments that demonstrate the feasibility of this approach.

Keywords: RFID Privacy, Private Identification, LWE Public-Key Cryptosystems, Sub-Threshold Design, Lattice Encryption, Passive RFID

1 Introduction

Many RFID applications such as supply chain management require the ability to uniquely identify individual tags, while scaling to billions of items and limiting the cost of a tag to a few cents. Such applications raise privacy concerns when individuals do not wish to be tracked or businesses do not want competitors to learn too much about their logistics. Public-key cryptosystems offer an attractive solution but standard public-key algorithms cannot be implemented in the severe area and power constraints for passive RFID tags.

For large scale private identification, no provably secure public-key encryption algorithm has been found that can be implemented on passive RFID tags. Instead, light-weight symmetric key schemes or hash functions are used. However, symmetric key approaches must sacrifice privacy for scalability. The power available on the passive RFID tag is the main limiting factor for the choice of cryptosystem. Passive RFID tags capture all their energy from their antenna coupling with the reader, so the power available for cryptographic operations is extremely low, typically a few microwatts. Implementations of standard public-key cryptosystems such as RSA and El Gamal require far more power than is available on passive RFID tags. Eliptic curve cryptography (ECC) is the most promising one but still requires area complexity around 15K gates. New public-key schemes or variations of known public-key encryption algorithm have been proposed [3,30], but the security of ad hoc schemes is unclear due to the lack of reduction to a classical hard problems. Section 2 provides more details on previous work.

In this paper, we introduce a new approach to implementing public-key cryptosystems on RFID tags. The main idea behind our approach is to use a lattice-based cryptosystem that provides a high level of security while only requiring simple (modular addition) logical operations. The main challenge in implementing this cryptosystem on a passive RFID tag is the large key size needed. We address this by using sub-threshold design techniques to reduce the size and power consumption needed to store the public key in ROM. In particular, we make the following contributions:

- We demonstrate the feasibility of implementing a public-key encryption on low-end passive RFID tags. We adopt the Learning-With-Error (LWE) lattice-based cryptosystem proposed by Oded Regev and proved secure via a reduction to classical lattice problems [28]. (Section 4)
- We present a private identification protocol based on the LWE cryptosystem. The protocol protects privacy by ensuring that tracking an RFID tag is as hard as breaking the LWE cryptosystem in a game model similar to the chosen-plaintext-attack model. (Section 5)
- We describe and evaluate a design in 130nm CMOS. Our results show the logic required to implement our design (1545 GEs) is far smaller than any other known public-key cryptosystem implementation. By using a combination of sub-threshold and near-threshold circuits, the power consumption is as low as $9.19\mu W$ and is well within the requirements of passive RFID tags). (Section 7)

2 Related Work

Much previous work has focused on the problem of privately identifying an RFID tag. Since the tags send messages over radio transmissions that can easily be intercepted, private identification requires using cryptographic protocols that take advantage of secret keys known only to legitimate readers. There are two main approaches: symmetric schemes where the tags and readers have shared secret keys, and asymmetric schemes.

In a pure symmetric scheme, the reader has a unique shared key with each tag in the system [33]. Pure symmetric key schemes cannot scale to support billions of tags since the reader needs to try all secret keys in the system to decrypt the received message. The cost of identifying a tag on the RFID reader must scale sub-linearly with the size of the system. Tree-based hash protocols [25,4] address this problem by assigning shared secrets to tags. This achieves scalability but sacrifices privacy [26, 9, 4]. Another approach is to use symmetric keys that are updated after each successful read [5, 6, 15, 31, 34, 12]. This approach sacrifices either availability or privacy for scalability. De-synchronization attacks that prevent a legitimate reader from being able to read a tag after an adversary interacts with it maliciously pose the main threat to this approach. Another drawback is that it requires rewritable memory and high power consumption to rewrite data on NVRAM memory for each read.

Asymmetric schemes have the advantage that identification can be done in constant time and there is no privacy loss when key material stored on individual tags is lost. Due to severe restrictions on implementation area and power consumption, new public-key cryptosystems as well as variations of previous systems have been proposed. A variant of Rabin's public-key scheme was proposed by Shamir [30] and implemented by Oren and Feldhofer (WIPR) [27]. However, subsequent research by Jiang Wu identified a serious security flaw in WIPR [35]. The proposed remedy requires a cryptographic hash function, which is too expensive for low-end tags.

The NTRU public-key cryptosystem, first proposed by Hoffstein, Pipher and Silverman in 1996, is a lattice-based cryptography employing only simple polynomial multiplications instead of exponentiation. This system was implemented with 2.8K gates with dynamic power consumption of $1.72\mu W$ [3]. However, there is no formal security proof for NTRU and it suffers from the lattice reduction attack [18]. To date, no public-key cryptosystem has been found that is adequate for passive RFID tags.

3 Private Identification for RFID

A private identification protocol enables a legitimate RFID reader to identify a tag without providing a way for an adversary to track, profile, or identify tags.

We adopt Juels' and Weis' definition of privacy [21] with a parameterized privacy experiment. It captures the idea of classic indistinguishability under chosen plaintext attack. The adversary $\mathcal{A}$ first corrupts at most $N-2$ tags, where N is the number of tags in the system, and performs any computation within its parameter bounds. $\mathcal{A}$ selects two uncorrupted tags as challenge candidates. One of them is randomly picked and presented to $\mathcal{A}$. $\mathcal{A}$ perform any computation within its parameter bounds and responds with a bit b' indicating which tag is picked. $\mathcal{A}$ wins the privacy experiment if $\mathcal{A}$ guess the chosen bit correctly with probability noticeably more than 50%. We strengthen the adversary's ability by eliminating parameterized communication bounds and setting $\mathcal{A}$ as standard interactive probabilistic polynomial Turing Machine since we admit $\mathcal{A}$ similar to the public-key cryptosystem adversary model.

Assume we have public-key cryptosystem $\Pi = \{Gen, Enc, Dec\}$, where n as a security parameter (e.g., key length) and a system with N tags. We define the privacy experiment as:

The Private Identification Protocol Exp$^{privacy}_{\mathcal{A},\Pi}(n)$

1. $Gen(n)$ is run to obtain a key pair $\langle PK, SK\rangle \leftarrow Gen(n)$.
2. Assign each tag its unique ID and store the information necessary for encrypting the ID.
3. In the learning phase, Adversary $\mathcal{A}$ is allowed to break at most $N-2$ tags and acquire all the information on the tag.
4. In the challenge phase, $\mathcal{A}$ picks two uncorrupted tags Tag^0 and Tag^1, a random bit $b \in \{0,1\}$ is chosen, denote $ID^b = ID$ of Tag^b. Then $c = Enc_{PK}(ID^b)$ is computed and given to $\mathcal{A}$.
5. $\mathcal{A}$ is allowed to interact with the tags in the system as follows: A can query $q \in \{0, N-1\}$. In response, A receives $Enc(ID^q)$, and outputs a bit b'.
6. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

Definition 1. *A protocol is* ***private*** *if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there exists a negligible function ϵ such that:*

$$Pr[\boldsymbol{Exp}^{privacy}_{\mathcal{A},\Pi}(n) = 1] \leq \tfrac{1}{2} + \epsilon(n)$$

In the above game, the adversary's objective is to perform malicious profiling or tracking attacks by distinguishing any two tags it picks, which threats a wide range of RFID applications.

4 The LWE Public-Key Cryptosystem

Our private identification protocols use the LWE public-key cryptosystem proposed by Oded Regev [28, 24] and proven to be chosen-plaintext-attack (CPA) secure based on the learning with error (LWE) problem. The hardness of LWE follows from known hard lattice problems, namely the decision version of the shortest vector problem (GapSVP) and the shortest independent vectors problem (SIVP). Unlike factoring-based asymmetric cryptosystems such as RSA, there is no known quantum algorithm to solve these problems.

The LWE problem assumes we have a secret vector $\mathbf{S} = [s_1, s_2, ...s_n] \in Z_P^n$ and polynomial random equations modulo prime P with errors:

$$\begin{cases} a_{11}s_1 + a_{12}s_2 + ... + a_{1n}s_n \approx b_1 \bmod P, \\ a_{21}s_1 + a_{22}s_2 + ... + a_{2n}s_n \approx b_2 \bmod P, \\ \qquad .... \\ a_{m1}s_1 + a_{m2}s_2 + ... + a_{mn}s_n \approx b_m \bmod P \end{cases} \tag{1}$$

Given $a_{ij} \in Z_P$, $b_i \in Z_P$ and P, where $i \in \{1, m\}, j \in \{1, n\}$, learning secret $\mathbf{S}$ from a set of equations with error is provably as hard as solving classic worst-case lattice problems [28].

Algorithm 1: The LWE based public-key Cryptosystem [24]

Parameters n, m, l, t, r, q, δ (all operations are done in modulo q)

Private Key Choose $\mathbf{S} \in \mathbb{Z}_q^{n\times l}$ uniformly at random. The private key is $\mathbf{S}$.

Public Key Choose $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ uniformly at random and $\mathbf{E} \in \mathbb{Z}_q^{m\times l}$ from a distribution determined by δ. The public key is $(\mathbf{A}, \mathbf{P} = \mathbf{AS} + \mathbf{E}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^{m\times l}$.

Encryption Given an element of the message space $\mathbf{v} \in \mathbb{Z}_t^l$ and a public key $(\mathbf{A}, \mathbf{P})$, choose a vector $\mathbf{a} \in \{-r, -r+1, ...r\}^m$ uniformly at random, and output the ciphertext $(\mathbf{u} = \mathbf{A^T a}, \mathbf{c} = \mathbf{P^T a} + f(\mathbf{v})) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^l$

Decryption Given a ciphertext$(\mathbf{u}, \mathbf{c}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^l$ and a private key $\mathbf{S} \in \mathbb{Z}_q^{n\times l}$, output $f^{-1}(\mathbf{c} - \mathbf{S^T u})$

The LWE cryptosystem proposed by Oded Regev is shown in Algorithm 1. For instance, the public key constructed from the set of equations (1) is:

$$PK = \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} & b_1 \\ a_{21} & a_{22} & ... & a_{2n} & b_1 \\ ... & ... & ... & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} & b_m \end{pmatrix} \tag{2}$$

To illustrate how LWE encryption works, consider each row in public key $\mathbf{a_i} = [a_{i1}, a_{i2}, ..a_{im}]$, since each equation satisfies $\mathbf{a_i} \cdot \mathbf{S} \approx b_i \mod P$, for a random subset $R \subseteq \{1, ..m\}$, we have $\sum_{i\in R} \mathbf{a_i} \cdot \mathbf{S} \approx \sum_{i\in R} b_i \mod P$. To encrypt a message compute the sum of a random subset of the rows, which is statistically close to uniform distribution if m is large enough [1, 28], and shift a small distance by a function of the message. For example, the encryption of 0 is $(\mathbf{c_1}, c_2) = (\sum_{i\in R} \mathbf{a_i} \bmod P, \sum_{i\in R} b_i \bmod P)$, and the encryption of 1 is $(\mathbf{c_1}, c_2) = (\sum_{i\in R} \mathbf{a_i} \bmod P, \sum_{i\in R} b_i + P/2 \bmod P)$. To decrypt with the decryption key $\mathbf{S}$, simply check if $\mathbf{c_1} \cdot \mathbf{S} \approx c_2$ to reveal the encrypted bit. Thus, encryption is done by summing up random rows in the public key $(\mathbf{A}, \mathbf{P})$ and adding a shift $f(\mathbf{v}) : \mathbb{Z}_t^l \to \mathbb{Z}_q^l$. The shift, $f(\mathbf{v})$, could be a simple function such as $\frac{t}{q}\mathbf{v}$.

To reduce the encryption blowup, the parameter l, t is introduced so that multiple bits can be encrypted in one round. To reduce the size of public key and increase security, each row can be added or subtracted up to r times instead of just 0 or 1 times. Figure 1 depicts various parameters in Algorithm 1.

The LWE cryptosystem has three notable advantages for RFID systems: (1) The only logical operation in encryption is modular addition which can be implemented cheaply in hardware; (2) It has proven security and resistance to quantum attacks; (3) It is a randomized encryption scheme so there is no linkability between any two ciphertexts for the same message.

4.1 Cyclic Key

Though the LWE logic unit is inherently simple, the memory size for storing the public key would dominate the die size and consequently the manufactur-

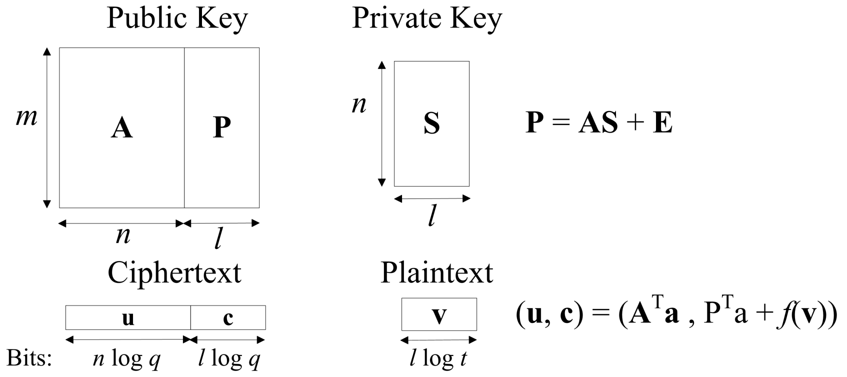


Fig. 1. Parameters in LWE Cryptosystem

ing cost. As indicated in Regev's paper [24], the size of the public key is in the order of megabits, which is out of reach for a low-end passive tag. A compact way of representing the public key without jeopardizing security is necessary.

The size of the public key $(\mathbf{A}, \mathbf{P})$ could be reduced dramatically by replacing the random matrix $\mathbf{A}$ with a cyclic matrix as proposed by Micciancio [23]. In a cyclic matrix, each column is a cyclic rotation of the first column. This reduces the key storage from $m(n + l)$ elements to $m(1 + l)$ elements. This twist takes the toll on the original security proof by Regev and replaces the hardness assumption on classic general lattice problems with cyclic lattice problems [23]. However, no algorithms are known so far that solve cyclic versions of the lattice problems more efficiently than the classic ones. It is assumed solving cyclic lattice problems is also hard [24]. Several efficient constructions such as the SWIFFT hash function [2] are based on cyclic lattices.

5 Private Identification Using LWE Cryptosystem

For private identification, a tag has to deliver its ID to a legitimate reader without revealing any information to malicious attackers. The LWE public-key cryptosystem has been proven to be CPA-secure and could be simply employed to encrypt the tag ID and deliver the ciphertext. The protocol is show in Figure 2.

Tag		Reader
Public key PK, ID_i		Private Key SK
	$\xleftarrow{query}$	
$C = Enc_{PK}(ID_i)$	$\xrightarrow{C}$	$Dec_{SK}(C)$

Fig. 2. Private identification Protocol 1

Theorem. The LWE Private Identification Protocol is *private*.
Proof sketch. To satisfy the privacy definition, we need to prove an adversary has no non-negligible advantage in the privacy game:

$$\Pr[\mathbf{Exp}^{privacy}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$$

The LWE cryptosystem has been proven to be CPA secure [28]. Comparing the CPA game $\mathbf{Pub}^{cpa}_{\mathcal{A},\Pi}(n)$ with the privacy game $\mathbf{Exp}^{privacy}_{\mathcal{A}',\Pi}(n)$, we notice that the two games are very similar except that the adversary $\mathcal{A}'$ in the privacy game has the power to break the tags and "decrypt" the message while the adversary $\mathcal{A}$ in the CPA game only has access to an encryption oracle. It seems that for $\mathcal{A}$ to invoke $\mathcal{A}'$, $\mathcal{A}$ needs to provide $\mathcal{A}'$ a "decryption" oracle. However, arbitrary ciphertexts are not "decryptable" by $\mathcal{A}'$ since $\mathcal{A}'$ has to find the tag which generates the message to break. This "decryption" procedure actually could be simulated by using $\mathcal{A}$'s the encryption oracle. During the challenging phase, $\mathcal{A}'$ gets to "interact" with the tags before outputting a guess. The LWE scheme works because when the scheme is "re-randomizable CPA-secure" then it can handle this by giving new randomizations of the received challenge ciphertext. Therefore $\mathcal{A}$ could successfully invoke $\mathcal{A}'$ in the CPA game and output what $\mathcal{A}'$ outputs. We show if an adversary $\mathcal{A}'$ wins $\mathbf{Exp}^{privacy}_{\mathcal{A},\Pi}(n)$ with non-negligible probability, then there exists an $\mathcal{A}$ to win the CPA experiment with non-negligible probability. Thus, breaking the privacy of the protocol implies breaking the LWE cryptosystem. (See the Appendix for proof details.)

5.1 Application in Data Sensitive Scenarios

In certain applications such as e-passports, the ID itself could be sensitive information which is risky to store on the tag. One solution to this is to store only indices on the tag and require readers to perform a back-end database lookup. This has the disadvantage that it requires readers to be online to gain any information from the tag. The LWE cryptosystem enables a solution that allows a tag to convey a meaningful ID directly to a legitimate reader without storing that ID on the tag except in encrypted form.

Figure 3 shows the protocol. The ID is encrypted once and stored on the tag as a constant. During every encryption performed on the tag, message 0

Tag		Reader
public-key $PK, C_{ID_i} = Enc_{PK}(ID_i)$		Private Key SK
	$\xleftarrow{query}$	
$C = C_{ID_i} + Enc_{PK}(0)$	$\xrightarrow{C}$	$Dec_{SK}(C)$

Fig. 3. LWE Protected Private Identification Protocol

is encrypted by the randomized encryption block and added to the encryption of ID. It takes the advantage of the malleability property with LWE cipher $ID = Dec(Enc_{PK}(ID) + Enc_{PK}(0))$. Each time, C_{ID_i} is the same value but $Enc_{PK}(0)$ keeps changing in a randomized way. (See the Appendix for the proof for privacy.)

Now, even an adversary who can physically break the tag only learns the encrypted ID value, and has no advantage for obtaining the plaintext tag ID.

5.2 Forward Security

Forward security (or forward traceability) ensures that revealing tag information at any time will not put in danger the security or indistinguishability of previously sent messages. Thus even if the adversary $\mathcal{A}$ breaks the tag at some point, $\mathcal{A}$ still has little advantage at tracing back the identity of the tag in previously recorded sessions. This is another form of a tracking attack that could jeopardize consumer privacy.

Also directly inheriting from the CPA security of the LWE cipher, the simple private identification protocol preserves the forward security. Since the adversary is the one who chooses two plaintexts and thus has the knowledge of the potential plaintext given the challenge of two ciphertexts. Therefore, even with the knowledge of the encrypted ID, an adversary has no advantage at distinguishing the ciphertexts from random guessing and the simple private identification scheme preserves forward security.

6 Parameter Selection

Table 1 summarizes the LWE parameters. Our goal is to find parameters that provide adequate security and response time, while minimizing implementation area and power consumption. We consider five metrics in Table 2.

Parameter	Meaning
n	number of columns in $\mathbf{A}$
m	number of rows in public-key ($\mathbf{A}$ and $\mathbf{P}$)
l	number of columns in $\mathbf{P}$
t	size of one character in the message space $v \in Z_t^l$
r	maximum number of times each row is selected by vector $\mathbf{a}$
q	the modulus
δ	used to the compute the distribution Φ_α with standard deviation $\alpha q/\sqrt{2\pi}$ from which the noise matrix E is generated and $\alpha = 4 \cdot max\{\frac{1}{q}, 2^{-2\sqrt{nlog(q)log(\delta)}}\}$

Table 1. LWE Parameters

6.1 Computation Time Model

The first three metrics are calculated using formulas from the LWE paper [24]. To derive the computation time to encrypt one message, we analyze the time complexity of processing each row of the public-key. If the generated random number is i, we need $|i|$ cycles to process this number before moving on to the next one. Since the value of i is uniformly distributed in the range $[-r, r]$, the average number of cycles to process a number is: $\frac{\sum_{i=-r}^{r} |i|}{2r+1} = \frac{r^2+r}{2r+1}$. The public-key has m rows and $n+l$ columns, so the expected time to encrypt one message is: $\frac{m(n+l)}{f \cdot N_{adder}} \frac{r^2+r}{2r+1}$, where f is the operating frequency and N_{adder} is the number of $\llcorner \log q \lrcorner$-bit modular adders.

Metrics	Measurement	Default
Security level (Lattice dimension in attack)	$\sqrt{n\log(q)/\log(\delta)}$	> 325
Encryption blowup	$\frac{(l+n)\log(q)}{l\log(t)}$	< 60
Error rate (per letter)	$2(1-\Phi(\frac{1}{2t\alpha} \cdot \sqrt{\frac{6\pi}{r(r+1)m}}))$	$< 0.9\%$
Computation time (s)	$\frac{m(n+l)}{f \cdot N_{adder}} \frac{r^2+r}{2r+1}$	$< 0.8s$
Storage for public-key (GEs)	$m(l+1)log(q)/\beta$	$\approx 6K$

Table 2. Algorithm Level Metrics

6.2 Gate Equivalents

To derive the area for storing the public-key, we consider 1 GE as the average area of 2-input low strength basic logic gates in a standard cell library. We looked at multiple commercial technology nodes from 130nm down to 65nm and found that 1 GE is about 10μm^2 in 130nm and increases by a factor of two as we go to the higher technology node. Ricci [29] describes a standard cell library for an RFID tag implementation and reports a number close to 20μm^2 for a GE in 0.18μm technology, which fits in the area and scaling trend we suggest for 1GE. This definition of GE allows comparisons of implementations across technology nodes, and also fits well with commercial standard cell libraries. We use this definition of GE to estimate area of both our scheme and the previous work.

6.3 ROM Area Model

We use a ROM to store the public-key, which is fixed and uniform across all tags. To estimate the area required for the ROM, we use previously published results. NAND ROM bit-cell area of less than 0.15μm^2 (in 90nm technology) has been reported by Chang [22] and Harris [16]. We have shown before that 1GE for 130nm is 10μm^2. Since bit-cell sizes scale regularly over technology nodes, a ROM bit-cell is equivalent to roughly 0.033 GE per bit, assuming 75% array efficiency. Thus, we estimate the GEs based on ROM bit-cells.

6.4 Parameter Selection

Based on the requirements of a large-scale private identification application in supply chain management, we set the default requirements on each metric as shown in Table 2. We estimate gate equivalent (GEs) for the storage of public-key by dividing the number of bits need to store by $\beta = 30$ (0.033 GE per bit as justified in Section 6.3).

We swept through the parameter space to find several interesting design points summarized in Table 3. The *Low Cost* parameters offer reasonable security within small ROM area and power consumption. The *Fast Encryption* parameters parallel adders to speed up. Since lattice encryption algorithm has a highly parallel dataflow and this can be easily exploited by having multiple modular adders working in parallel. The increasing power on adders is offset by the decreasing power of ROM due to the reduced frequency. For the *Fast Encryption* and *Low Power* designs, we use four adders to minimize the total power consumption. The *Low Power* parameters reduce the power consumption by decreasing the operating frequency and the *Strong Security* selects parameters that produce a high security level as estimated by lattice dimension.

Parameter	Low Cost	Fast Encryption	Low Power	Strong Security
n	152	152	152	198
m	1005	1005	1005	1238
l	12	12	12	12
t	16	16	16	16
r	2	2	2	2
q	8219	8219	8219	6803
δ	1.013	1.013	1.013	1.011
# adders	1	4	4	1
Freq (KHz)	800	800	200	800
Security (Dim)	326	326	326	400
Storage (GEs)	6036	6036	6036	6904
Blowup	48	48	48	57
Error rate	0.69%	0.69%	0.69%	0.742
Time (ms)	494.46	123.6	494.46	779.94

Table 3. Parameter Selection

7 Implementation

In this section, we describe our implementation of the private identification protocol on RFID tags based on the LWE encryption algorithm and discuss the low area, low power techniques for components such as logic block, memory and random number generator.

7.1 Ultra-Low Power Logic

Sub-threshold operation, or operation of a circuit below the threshold voltage of a transistor, has been shown to lower power in memory [10], processor [32] and system design [20]. Lowering voltage increases circuit delay as well, and thus power (CV^2f) decreases at a fast rate. We leverage sub-threshold and near-threshold operation in the implementation of our scheme. Since RFID encryption schemes work at sub-1MHz frequencies, such low voltages are sufficient to provide the necessary performance.

At supply voltages near the threshold voltage, excessive leakage and variation start becoming more pronounced. To lower the impact of these effects we choose an older technology (130nm) for our implementation. We simulate the design generated by the synthesis tool (RTL Compiler) and the place and route tool using circuit level simulator Ultrasim. This step eliminates possible errors that may be caused as these tools use circuit data characterized at nominal voltages (1.2V).

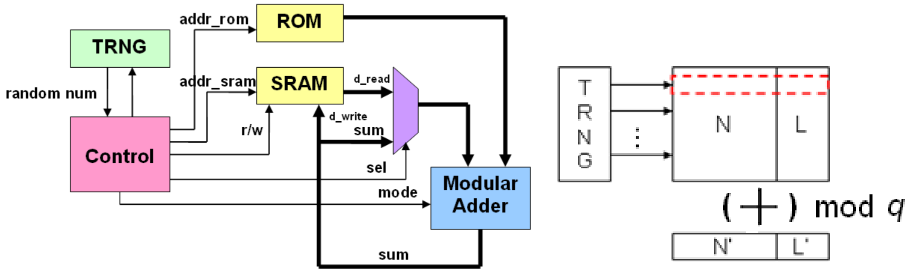
7.2 Design Architecture

In order to evaluate the performance, area and power consumption of the LWE encryption design, we implemented the encryption circuit in VHDL and synthesized it with RTL compiler from Cadence. Automatic place and route was done by SOC Encounter. The final extracted netlist was simulated using the Ultrasim simulator. We obtain the encryption time using behavioral RTL simulation. Area is gathered from the Encounter gatecount report, and power is calculated by averaging the simulated current waveform over 1000 cycles.

Figure 4 shows the architecture of cyclic lattice cipher and the logical operations being performed. The public-key is stored in ROM at manufacturing time. A *true random number generator* (TRNG) generates random numbers in the range of $[-r, r]$ for row selection (Section 7.4. The modular adder performs the actual computation. The running sum is stored in an SRAM, which provides two ports for simultaneous read and write in a cycle. The control module coordinates the whole encryption process. The final values stored in the SRAM are transmitted as the encryption output.

7.3 Encryption Logic

The encryption logic consists of a control and a modular adder unit. Since the modular adder can only process one public-key element per cycle, it needs to process all the elements of a given row before starting the the next row (*row-wise scheme*). Another *column-wise scheme* accumulates the elements in a given column first. The former scheme is adopted because it greatly reduces the operating frequency and power of the RNG. This scheme requires a



(a) (b)

Fig. 4. (a) Top level architecture of our cyclic lattice cipher; (b) Main computation involved in lattice cipher.

small SRAM for storing intermediate sums. The SRAM has both read and write ports. They keep the modular adder busy for highest performance.

Whenever the RNG generates a zero, it must waits for 40 cycles before generating the next number. This enables the RNG to run 40 times slower than the main logic, significantly lowering its power consumption. The time overhead is small because it takes much more than 40 cycles to process a non-zero row.

The modular addition/subtraction is performed by the *modular adder*. Its output is connected back to one of its input ports, achieving the effect of accumulation. The *mode* pin controls the type of operation (modular addition/subtraction) to be performed. The latency from the input to the output is one cycle, so no pipelining is necessary.

The logic part of the circuit operates at under 0.5V, the lowest voltage to reliably perform the encryption at 800KHz clock frequency. Operating at lower voltages than 0.5V has diminishing returns for power as the leakage power starts to dominate.

7.4 Random Number Generation

Cryptographic applications require cryptographically strong random numbers, and many low-power random number generators (RNG) have been proposed for RFID applications [8, 11, 7, 14]. Bucci [8] implemented a true RNG which consumes 2.3mW of power while delivering a throughput of 10Mbps. This RNG fulfills the NIST FIPS and correlation-based tests for randomness. Since power is roughly a linear function of frequency and quadratic function of V_{dd}, we scale down the throughput and supply voltage from 10Mbps, 1.8V down to the needed 40Kpbs, 0.5V and estimate the power to be 0.35μW.

7.5 Sub-Threshold ROM

In this section we focus on estimating the read power for the ROM that we need for public-key storage. ROM design in sub-threshold is challenging because of code-dependent read noise in the presence of bit-line leakage, charge sharing, and crosstalk. To estimate ROM power we choose a design that has been demonstrated in silicon in the sub-threshold region by Chang et al. [22]. This ensures that our estimates for ROM power reflect the design modification needed in a ROM for working at low voltages.

We calculate dynamic and leakage energy separately. Of the dynamic energy, 10% is allocated to the timing block of the ROM, and this remains fixed across various ROM sizes. The rest of the dynamic energy is consumed in the bit-lines of the ROM. Bit-line size increases linearly with the number of rows, and the number of bit-lines increases linearly with the number of columns. Thus 90% of the dynamic energy of a large ROM scales linearly with the ROM capacity. Leakage in a large ROM is consumed mainly in the bit-cells and the word-line drivers. Leakage per word-line driver is about 20% the of leakage of a row of 512 bit-cells. Thus, 20% of total cited leakage can be attributed to word-line drivers. This allows us to estimate the leakage per word-line driver using the number of word-line drivers from Chang et al.'s results [22]. The rest of the leakage is consumed by bit-cells, so we can also estimate the per bit-cell leakage. We then use the leakage per word-line and bit-cell to calculate the leakage for our ROM size.

To take into account the impact of technology node, we scaled dynamic energy, leakage power, and delay by $\sqrt{2}$x, 2x, and $\sqrt{2}$x as we go from one technology node to an older technology node. These factors are consistent with constant field scaling. A custom ROM built for the exact capacity that is desired would be optimized in both power and delay as compared to a model that's extrapolated from another point in the design space.

7.6 Results

Table 4 summarizes the results from our simulation experiments for the designs in Table 3. The power and area for each components are listed. As expected, several design points gives better results in corresponding metrics. Small area gives low cost — 8297 GEs is relatively small among the implementations of public-key schemes. By using four adders in parallel, the transaction time could be reduced to 132ms. Due to sub-threshold and near-threshold design, the power consumption is low and does not vary too much among the four design points. The lowest power achieved is $9.19 \mu W$. High security is achievable with moderate additional area cost, but still below 10K total GEs.

7.7 Comparison with Related Work

Table 5 compares our results with other public-key encryption implementations targeting RFID applications. Elliptic curve cryptography (ECC) has been regarded as the most promising widely-used public-key cryptosystem for RFID tags. However, the area and power are still beyond the reach of low-power, low-cost passive RFID tags.

We implemented WIPR-RNS [35] in 6793 GEs for logical components and 71GEs for memory We apply the subthreshold design to WIPR-RNS as well

			Low Cost	Fast	Low Power	Strong Security
Frequency (KHz)			800	800	200	800
Power (μW)	logic	modular adder	0.34	1.36	0.63	0.36
		rest	0.27	0.27	0.07	0.28
	memory	ROM	8.10	8.10	7.40	9.10
		SRAM	1.50	1.50	1.0	1.50
	RNG (@20KHz)		0.35	0.35	0.09	0.35
	total		10.56	11.58	9.19	11.59
Area (GEs)	logic	modular adder	352	1408	1408	329
		rest	489	489	489	495
	memory	ROM	6036	6036	6036	6904
		SRAM	620	620	620	784
	RNG ($20\mu m^2$/GE)		800	800	800	800
	total		8297	9353	9353	9312
Security (Lattice Dimension)			326	326	326	400
Transaction time (ms)			528	132	528	840
Energy per Tran (μJ)			5.57568	1.52856	4.8532	9.7356

Table 4. Cost and Performance Evaluation of Lattice Cipher

and the power consumption is very small. Unfortunately, WIPR-RNS cannot achieve satisfactory security due to the implementation flaw identified by Jiang Wu [35]. The proposed remedy requires a cryptographic hash function, which is too expensive for low-end tags.

The LWE-Cost, LWE-Power and LWE-Time are corresponding to the three design points (Low Cost, Low Power, Fast) from Table 3. They are suitable for applications with different requirements.

Another related work in public key cryptography for RFID is the GPS scheme [13] proposed by Girault,Poupard and Stern (GPS). GPS is a zero-knowledge authentication scheme, which has been implemented, fabricated and ISO standardized [19]. The RFID tag which possesses a secret key can prove its identify to the reader with cheap operations. However, it is not scalable for identification purposes and since it is designed for different functionality, its implementation results are not included in Table 5.

Algorithm	Area (GEs)	Freq (KHz)	Power (μW)	Cycles (k)	Trans (s)	Energy (μJ)	Tech
ECC-163 [17]	15K	106	8.57	296	2.79	23.91	180nm
ECC-192 [17]	23.6K	106	19.95	500	4.7	93.76	180nm
WIPR-RNS	6.9K	1 MHz	2.84	149	0.14874	0.42	130nm
LWE-Cost	9K	800	10.56	422	0.528	5.57	130nm
LWE-Power	11K	200	9.19	105	0.528	4.85	130nm
LWE-Time	11K	800	11.58	105	0.132	1.53	130nm

Table 5. Comparison with Other Public-Key Cryptographic Algorithms

8 Conclusion

Providing a high level of privacy at a low cost for large scale RFID applications remains an important and elusive goal. Our results provide reason for optimism that new developments in asymmetric cryptosystems will enable public-key encryption on RFID tags. Our simulation experiments and analyses show that an implementation of a private identification protocol based on the LWE cipher is within the power and area constraints for low-cost RFID systems. The LWE cipher offers many advantage over previous alternatives including it simple logic and provable security even against quantum attacks. Further we show how circuit techniques like sub-threshold and near-threshold operation help reduce power drastically in RFID applications where performance is not tightly constrained.

References

1. Ajtai, M.: Generating Hard Instances of Lattice Problems (Extended Abstract). In: Twenty-Eighth Annual ACM Symposium on Theory of Computing (1996)
2. Arbitman, Y., Dogon, G., Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: SWIFFTX: A Proposal for the SHA-3 Standard (2008)
3. Atici, A.C., Batina, L., Fan, J., Verbauwhede, I., Yalcin, S.B.O.: Low-Cost Implementations of NTRU for Pervasive Security. In: IEEE International Conference on Application-Specific Systems, Architectures and Processors (2008)
4. Avoine, G., Martin, B., Martin, T.: Tree-Based RFID Authentication Protocols Are Definitively Not Privacy-Friendly. In: Workshop on RFID Security (2010)
5. Avoine, G., Oechslin, P.: A Scalable and Provably Secure Hash-Based RFID Protocol. In: Third IEEE International Conference on Pervasive Computing and Communications Workshops (2005)
6. Bolotnyy, L., Robins, G.: Physically Unclonable Function-Based Security and Privacy in RFID Systems. In: International Conference on Pervasive Computing and Communications (2007)
7. Brederlow, R., Prakash, R., Paulus, C., Thewes, R.: A Low-Power True Random Number Generator using Random Telegraph Noise of Single Oxide-Traps. In: Solid-State Circuits Conference (2006)
8. Bucci, M., Germani, L., Luzzi, R., Trifiletti, A., Varanonuovo, M.: A High-Speed Oscillator-Based Truly Random Number Source for Cryptographic Applications on a Smart Card IC. IEEE Transactions on Computers Vol 52.(4) (April 2003)
9. Buttyán, L., Holczer, T., Vajda, I.: Optimal Key-Trees for Tree-Based Private Authentication. In: Workshop on Privacy Enhancing Technologies (2006)
10. Calhoun, B.H.; Chandrakasan, A.: A 256kb Sub-Threshold SRAM in 65nm CMOS. In: International Solid-State Circuits Conference (2006)
11. Che, W., Deng, H., Tan, W., Wang, J.: A Random Number Generator for Application in RFID Tags. In: Networked RFID Systems and Lightweight Cryptography (2008)
12. Erguler, I., Anarim, E.: Scalability and Security Conflict for RFID Authentication Protocols. In: Cryptology ePrint Archive (2010)
13. Girault, M., Poupard, G., Stern, J.: On the Fly Authentication and Signature Schemes Based on Groups of Unknown Order. In: Journal of Cryptology (2006)

14. Gueler, U., Erguen, S.: A High Speed IC Random Number Generator Based on Phase Noise in Ring Oscillators. In: 2010 IEEE International Symposium on Circuits and Systems (ISCAS) (2010)
15. Ha, J., Ha: LRMAP: Lightweight and Resynchronous Mutual Authentication Protocol for RFID System. In: 1st International Conference on Ubiquitous Convergence Technology (2007)
16. Harris, N.W.: CMOS VLSI Design A Circuits and Systems Perspective. Addison Wesley (2004)
17. Hein, D., Wolkerstorfer, J., Felber, N.: ECC is Ready for RFID - A Proof in Silicon. In: Workshop on RFID Security (2008)
18. Howgrave-Graham, N.: A Hybrid Lattice-Reduction and Meet-In-The-Middle Attack Against NTRU. In: 27th Annual International Cryptology Conference on Advances in Cryptology (2007)
19. ISO/IEC: 9798: Information Technology - Security Techniques - Entity Authentication - Part 5: Mechanisms using Zero-Knowledge Techniques. (2006)
20. Jocke S.C., Bolus J.F, C.B.: A 2.6-uW Sub-Threshold Mixed-Signal ECG SoC. In: 2009 Symposium on VLSI Circuits (2009)
21. Juels, A., Weis, S.: Defining Strong Privacy for RFID. In: International Conference on Pervasive Computing and Communications (2007)
22. Meng-Fan Chang, S.M.Y.: A 0.29V Embedded NAND-ROM in 90nm CMOS for Ultra-Low-Voltage Applications. In: International Solid-State Circuits Conference (2010)
23. Micciancio, D.: Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions from Worst-Case Complexity Assumptions. In: 43rd Symposium on Foundations of Computer Science (2002)
24. Micciancio, D., Regev, O.: Lattice-based Cryptography . In: Post-Quantum Cryptography (2009)
25. Molnar, D., Wagner, D.: Privacy and Security in Library RFID: Issues, Practices, and Architectures. In: Conference on Computer and Communications Security (2004)
26. Nohl, K., Evans, D.: Quantifying Information Leakage in Tree-Based Hash Protocols. In: International Conference on Information and Communications Security (2006)
27. Oren, Y., Feldhofer, M.: A Low-Resource Public-Key Identification Scheme for RFID Tags and Sensor Nodes. In: 2nd ACM Conference on Wireless Network Security (2009)
28. Regev, O.: On Lattices, Learning With Errors, Random Linear Codes, and Cryptography. In: Thirty-Seventh Annual ACM Symposium on Theory of Computing (2005)
29. Ricci A, G.M.: Design of a Low-Power Digital Core for Passive UHF RFID Transponder. In: 9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools (2006)
30. Shamir, A.: Memory Efficient Variants of Public-Key Schemes for Smart Card Applications. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques (1994)
31. Song, B.: Scalable RFID Authentication Protocol. In: 3rd International Conference on Network and System Security (2009)
32. Wang, A.; Chandrakasan, A.: A 180mV FFT Processor using Subthreshold Circuit Techniques. In: International Solid-State Circuits Conference (2004)
33. Weis, S., Sarma: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: International Conference on Security in Pervasive Computing (2003)

34. Wu, J., Stinson, D.R.: A Highly Scalable RFID Authentication Protocol. In: 14th Austrialasian Conference on Information Security and Privacy (2009)
35. Wu, J., Stinson, D.R.: How to Improve Security and Reduce Hardware Demands of the WIPR RFID Protocol. In: IEEE International Conference on RFID (2009)

Appendix

Proof for Private Identification Protocol 1

Tag		Reader
Public key PK, ID_i		Private Key SK
	$\xleftarrow{query}$	
$C = Enc_{PK}(ID_i)$	$\xrightarrow{C}$	$Dec_{SK}(C)$

Fig. 5. Private identification Protocol 1

Proof: To show this protocol is private, we need to prove $\Pr[\mathbf{Exp}_{\mathcal{A},\Pi}^{privacy}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$. Recall that Π here is the LWE crypto which has been shown to be CPA. The CPA experiment is summarized below for convenience:
$\mathbf{Pub}_{\mathcal{A},\Pi}^{cpa}(n)$

1. Gen is run to obtain keys $(pk, sk) \leftarrow Gen(1^n)$
2. Adversary $\mathcal{A}$ is given pk and oracle access to $Enc_{pk}(\cdot)$. It outputs two messages m_0, m_1 of the same length $(m_0, m_1) \leftarrow \mathcal{A}^{Enc_{pk}(\cdot)}(pk, n)$
3. A random bit $b \leftarrow \{0, 1\}$ is choosen. A ciphertext $c \leftarrow Enc_{pk}(m_b)$ is computed and given to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a bit $b' \leftarrow \mathcal{A}^{Enc_{pk}(\cdot)}(c)$
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

A public-key cryptosystem Π has indistinguishable encryptions under chosen-plaintext attack if for all probabilistic polynomial-time adversaries $\mathcal{A}$, there exists a negligible function ϵ such that:

$$\Pr[\mathbf{Pub}_{\mathcal{A},\Pi}^{cpa}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$$

Comparing the privacy game $\mathbf{Exp}_{\mathcal{A},\Pi}^{privacy}(n)$ with the CPA game, we notice that we allow $\mathcal{A}$ to access the plaintexts of received messages by breaking the tags physically. However, this is not equivalent to a decryption oracle which enables $\mathcal{A}$ to access plaintexts of arbitrary ciphertexts $\mathcal{A}$ picks. We show if an adversary $\mathcal{A}'$ wins $\mathbf{Exp}_{\mathcal{A},\Pi}^{privacy}(n)$ with non-negligible probability, then there exists $\mathcal{A}$ to win the CPA experiment with non-negligible probability. Here is a way for $\mathcal{A}$ to win the CPA experiment by invoking $\mathcal{A}'$:

1. $(pk, sk) \leftarrow Gen(1^n)$

2. $\mathcal{A}$ is given pk and oracle $Enc_{pk}(\cdot)$
   (a) assign each tag with pk
   (b) randomly generate $ID_0, ID_1, ..., ID_N$ with same length
   (c) invoke $\mathcal{A}'$. Use oracle $Enc_{pk}(ID_i)$ as responses to the query of i_{th} tag by $\mathcal{A}'$, and reveal ID_i if $\mathcal{A}'$ chooses to break the i_{th} tag.
   (d) Output the two IDs $\mathcal{A}'$ picks as m_0, m_1
3. $b \leftarrow \{0, 1\}$, and $c \leftarrow Enc_{pk}(m_b)$ is presented to $\mathcal{A}$
4. $\mathcal{A}$ feed c to $\mathcal{A}'$
5. Use oracle $Enc_{pk}(ID_i)$ as responses to the query of i_{th} tag by $\mathcal{A}'$ when $\mathcal{A}'$ interact with tags
6. $\mathcal{A}$ outputs a bit b' as $\mathcal{A}'$ outputs

Therefore, if $\mathcal{A}'$ wins the the privacy game with non-negligible probability, $\mathcal{A}$ is able to win the CPA game with non-negligible probability.

Proof for Private Identification Protocol 2

Tag		Reader
Public key PK, $Enc_{PK}(ID_i)$		Private Key SK
	$\xleftarrow{query}$	
$C = Enc_{PK}(ID_i) + Enc_{PK}(0)$	$\xrightarrow{C}$	$Dec_{SK}(C)$

Fig. 6. Private identification Protocol 2

Proof: To show this protocol is private, we need to prove Pr[$\mathbf{Exp}^{privacy}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$. We prove by reduction. Assume we have adversary $\mathcal{A}'$ that breaks $\mathbf{Exp}^{privacy}_{\mathcal{A},\Pi}(n)$ with non-negligible probability:

1. $Gen(n)$ is run to obtain key pair $\langle PK, SK \rangle$ for system
2. Store on each tag $C_{ID_i} = Enc_{PK}(ID_i)$ and PK
3. In the learning phase, Adversary $\mathcal{A}'$ is allowed to break at most $N - 2$ tags and acquire C_{ID_i} and PK on the tag, where N is the number of tags in the system
4. In the challenge phase, $\mathcal{A}'$ picks two uncorrupted tags Tag^0 and Tag^1, a random bit $b \in \{0, 1\}$ is chosen, denote $C^b_{ID} = C_{ID}$ of Tag^b. $c = C^b_{ID} + Enc_{PK}(0)$ is computed and given to $\mathcal{A}'$
5. $\mathcal{A}'$ is allowed to communicate with two uncorrupted tags and output a bit b'
6. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise

Now we can construct Adversary $\mathcal{A}$ to break the CPA game:

1. $(pk, sk) \leftarrow Gen(1^n)$
2. $\mathcal{A}$ is given pk and oracle $Enc_{pk}(\cdot)$
   (a) assign each tag with pk
   (b) randomly generate $ID_0, ID_1, ..., ID_N$ with same length
   (c) use oracle $Enc_{pk}(\cdot)$ to produce $C_{ID_0}, C_{ID_1}, ..., C_{ID_k}$
   (d) invoke $\mathcal{A}'$. Use oracle $Enc_{pk}(ID_i)$ as responses to the query of i_{th} tag by $\mathcal{A}'$, and reveal C_{ID_i} and PK if $\mathcal{A}'$ chooses to break the i_{th} tag.
   (e) Output the two IDs (denote as ID^0, ID^1) corresponding two C_{ID} (denote as C^0_{ID}, C^1_{ID}) $\mathcal{A}'$ picks
3. $b \leftarrow \{0, 1\}$, and $c \leftarrow Enc_{pk}(m_b)$ is presented to $\mathcal{A}$
4. $\mathcal{A}$ feed $C^0_{ID} + (c - Enc_{pk}(ID^0))$ to $\mathcal{A}'$
5. Use oracle $Enc_{pk}(0) + C_{ID_i}$ as responses to the query of i_{th} tag by $\mathcal{A}'$ when $\mathcal{A}'$ interact with tags
6. outputs a bit b' as $\mathcal{A}'$ outputs

Here, if $b == 0$, $c - Enc_{pk}(ID^0)$ is equal to $Enc_{pk}(0)$ and $\mathcal{A}'$ should have non-negligible probability to break it. Thus if Pr[$\mathbf{Exp}^{privacy}_{\mathcal{A}',\Pi}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$, then the success probability for $\mathcal{A}$ is: Pr[$\mathbf{Exp}^{privacy}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \frac{1}{2}\epsilon(n)$.