A low-frequency RFID to challenge security and privacy concerns

Tim Good
Dept. of Electrical and Electronic Engineering
University of Sheffield, Sheffield, S1 3JD, UK
Email: t.good@sheffield.ac.uk

Mohammed Benaissa
Dept. of Electrical and Electronic Engineering
University of Sheffield, Sheffield, S1 3JD, UK
Email: m.benaissa@sheffield.ac.uk

***Abstract*—This paper presents a practical realization of a secure passive (battery-less) RFID tag. The tag consists of an off the shelf front end combined with a bespoke 0.18μm ASIC assembled as a credit card sized prototype. The ASIC integrates the authors' ultra low power novel AES design together with a novel random number generator and a novel protocol which provides both security and privacy. The analysis presented shows a security of 64-bits against many attack methods. Both modeled and measured power results are presented. The measured average core power consumed during continuous normal operation is 1.36μW.**

I. Introduction

Radio Frequency Identification (RFID) systems have grown in popularity in recent years. Such systems have three principal components, a radio tag or transponder, a reader and a database [1]. The tags comprise small integrated circuits connected to typically a small wire coil antenna and attached to an item or carried by a person to facilitate electronic identification. This can be in terms of an electronic product code (EPC) [2] or a unique serial number. The reader emits a radio signal which provides the challenge to tag and in the case of passive tags also provides the source of energy for the tags operation. The RFID process is non-contact, does not require line of sight and depending on the selected RF band and antenna design can be carried out at ranges from several centimeters to several meters. Typically a database is then queried using the tag's identifier to provide further details.

The technology already pervades our daily lives from management of the supply chain (attached to goods in retail stores, car tiers, etc) through to the 'chip' in you car key which operates the immobilizer [3]. RFID already offers very many benefits to society however there have been a number of privacy and security concerns raised regarding the proliferation and standardization of RFID together with real world examples of exploitation of the negative aspects [4]–[8]. The privacy concern arises from the ability to remotely interrogate an RFID at a distance to ascertain some information about the individual or individual's property [9]. One particular concern is the association of tag response(s) with a specific individual disclosing their location, often referred to as location privacy [10].

The majority of the population of tags form part of the supply chain and are removed or disabled ('killed') at a point of sale [11]. It has been argued [12] that if the disposition of tags is tracked by so called 'smart shelves' during shopping then the individuals shopping habits can be ascertained. However, there is a second class of RFID, where as part of its normal lifecycle remains active whilst in the possession of an individual, thus poses far greater privacy and security concerns. Examples include: identity cards, car keys, car tires, medicine packaging and some higher value retail products.

Economics plays a large role in the design of RFID tags, they must be fundamentally low cost as they are frequently attached to low value items. Their deployers are normally interested in issues such as product authentication, counterfeit detection and supply chain efficiency. It has been stated that to be economic any tag-borne security measures must fit within an area of 250-3000 gate equivalents [8], [10], [11].

Research in RFID technology is currently very active and is summarized in two recent review papers [8], [13]. The challenge is to develop secure protocols for RFID which do not leak sufficient information (i.e. an identifier) which in turn may be used to derive personal information about its owner / bearer. Previous attempts have focused exclusively on privacy at the expense of security, and vice-versa. Even the best previous attempts at such protocols [10], [14]–[18] have vulnerability to either Denial of Service (DoS) attacks, radio-relay attack [19] or allow user tracking via a unique constellation of non-unique identifiers [10].

Modeled results for the baseband part of a UHF RFID tag using the AES were reported in [20]. However, the design is rather large (approx. 3 times larger than this work) with only simulated power results and with unknown duty cycle. Further, a Tausworthe PRNG is used which may initialize to a known state facilitating a number of attacks. The on-tag storage of a long-term secret-key shared between a large set of tags and readers makes a tempting target for reverse engineering or side channel analysis of a tag.

In this paper, work is presented that shows the practicality of integrating a strong cryptographic primitive into a battery-less RFID together with a secure protocol and supporting random number generator (RNG) to produce a working prototype tag without the need for writing to non-volatile memory during its normal operation. This is believed to be the first reported ASIC implementation for such a design. A number of innovations, in terms of very low power, very low number of cycles and very low area for the strong crypto. primitives, were

978-1-4244-5113-5/09/$25.00

made to achieve the required performance within the stringent constraints imposed by low frequency passive RFIDs. The results from a manufactured prototype (Oct 2007), including a dedicated 0.18 CMOS chip, are presented to demonstrate functionality and performance.

Section II of this paper discusses the challenges for passive (battery-less) RFID, section III describes a novel protocol which provides both security and privacy together with an initial security analysis. Sections IV and V present the strong cryptographic primitive and novel random number generator which are both sufficiently low power and low area to be suitable for a battery-less RFID tag. This is followed by section VI describing the results from a manufactured prototype and finally the conclusions are presented in section VII.

II. Engineering Challenges for Battery-less RFID

Typically such tags are powered by rectifying the applied RF field and use this same field as the clock source. This constrains the design to operate on very tight energy / power budgets and effectively fixes the clock rate. The limited power also limits the available area in terms of static power consumption together with economic factors.

The challenge-response model requires two-way communication, however, the tag derives its clock from the readers transmission thus cannot discern time/phase changes thus the only suitable modulation for reader-to-tag communication is often basic on-off keying. Conversely the tag-to-reader channel may select a more efficient modulation.

There are a number of protocols which have been proposed which require the tag to update its NVRAM, such write operations are typically expensive in terms of power and time and also raise data integrity issues (due to loss of power) which must be addressed by additional complexity. Further, if there is a requirement to write to the NVRAM this opens up a set of DoS attacks with a high degree of permanence.

III. Protocol

For an authentication system, the objective is to prove knowledge of an identifier without compromising the identifier to any potential attacker.

The existing protocols for battery-less RFID systems fit into a number of categories that have already been reviewed in detail in [8], [13] and briefly summarized below:

staticID: When energized the tag responds by returning a string of bits composing a fixed identifier, for example an electronic product code. Such schemes are common in the retail sector and use write-enable 'unlocking' passwords and are typically removed or disabled at the point-of-sale.

refreshed ID: As with static ID the tag repeatedly broadcasts its identifier when energized, however, on successful reading by a legitimate reader a new identifier is generated by the reader and sent to the tag typically with an write-enable 'unlocking' password.

hashed ID: The tag performs a hashing operation, $H(ID)$, on its own identifier, storing the new result and transmits part of the result to the reader as its temporary identifier. All tags in this category require to write to their NVRAM a new identifier.

keyed authentication: The tag performs a keyed hashing operation with its identifier with a once-only random number (N) to yield an authentication code which is transmitted to the reader. There are a number of variations in this category depending on the source of N; these are tag generates Nt, reader transmits Nr to tag or both.

Table I summaries the communication and tag operation overhead (all are assumed to include transmission, reception and non-volatile memory storage) for the various schemes.

TABLE I
Types of tag and their operations

Type	reader to tag	on-tag operations	tag to reader
static	none	-	ID
refreshed	newID+passwd	NVwrite	ID / newID
hashed	none	H(ID), NVwrite	part H(ID)
auth(R)	Nr	H(ID,Nr)	H(ID,Nr)
auth(T)	none	gen(Nt), H(ID,Nt)	Nt,H(ID,Nt)
auth(M)	Nr	gen(Nt), H(ID,Nr,Nt)	Nt,H(ID,Nr,Nt)

A protocol may be analyzed as a set of games played out by the legitimate participants and would-be attackers of the system. The holder of the tag may be considered to be the prover and the reader system the verifier.

authentication game: the verifier seeks a message from the prover to show they know some secret (the identifier). Typically in order to avoid replay attacks this involves a unique challenge issued by the verifier. An attacker seeks to permanently prevent future legitimate authentication (denial of service).

counterfeit game: An attacker may try to copy a tag or the tag's responses with the aim of either compromising the tag's identifier or seek to duplicate responses from an authentic tag.

anonymity game: An attacker seeks a static or predictable identifier with the aim of tracking the tag or its carrier. There is a totalitarian variation of this game where the attacker is a legitimate reader of tags and seeks to track all tags. There are very many different scenarios and methods of attack which can be played out. A brief summary is given in table II.

The novel authentication protocol developed here falls into the auth(Mutual) category. Random numbers are generated by both reader and tag and a keyed hash operation used to produce the authentication code. This novel protocol is based on [21] with a repeated challenge being used to avoid the inevitable auth code collisions (birthday paradox) and provide security of at least $O(2^{64})$.

It should be noted that the XOR operation is not suitable for combining Nr and Nt (the tag has not committed to Nt thus could attempt to cheat the reader), thus concatenation should be used instead.

Such a protocol avoids the tag having to perform any NVwrite operations however does require transmission of Nr by the reader. The inclusion of Nr (a random the reader is content with) prevents the trivial replay attack.

TABLE II
TAG ID PROTECTION

Type	Authentication DoS	Counterfeit	Anonymity
static	× large no of duplicated fake IDs cannot determine real one	× ID in clear can directly copy	× ID is in clear
refreshed	× mandates a single authority / database for refreshing ID, data link vulnerable	✓ ID in clear but of limited lifetime	× reader can use predictable sequence for 'new ID'
hashed	× desynchronization / loss of chain-of-IDs possible and irrecoverable	✓ strong	× future values of ID can be precomputed
auth (Reader)	✓ strong	✓ strong	× reader can use standardized challenge
auth (Tag)	× really needs single database to mitigate replay attacks ideally store all responses impractical!	× tag can use small set of Nt with known auth codes	✓ strong
auth (Mutual)	✓ strong	✓ strong	✓ strong

The protocol is summarized by the following algorithm:

Require: reader
  $Nr \leftarrow random\ number$
  transmit Nr
Require: tag
  $Nr \leftarrow received\ value$
  $Nt \leftarrow random\ number$
  $X \leftarrow H(ID, Nr|Nt)$
  transmit Nt and (part) X
Require: reader
  $Nt_1, pX_1 \leftarrow received\ value$
  for all IDs in database **do**
    $X' \leftarrow H(IDi, Nr|Nt_1)$
    if (part) $X' = pX_1$ **then**
      $Nr_2 \leftarrow random\ number$
      transmit Nr_2
      wait for tags new response
      $Nt_2, pX_2 \leftarrow received\ value$
      $Y' \leftarrow H(IDi, Nr_2|Nt_2)$
      if (part) $Y' = pX_2$ and $pX_1 \neq pX_2$ **then**
        tag is IDi
      end if
    end if
  end for

As the internal identifier of the tag remains unchanged there is easy support for multiple reader databases and privacy is protected by the dependence of the authentication message on the unpredictable Nt (a random the tag is content with). This forces the reader / database to perform comparison with all known tags for a match (computationally expensive) thus limits the size of databases. For databases, say at the national level, this may be mitigated by the user of the tag supplying some (any) additional information to reduce the search space (eg a pin number or date of birth).

For wide public acceptance ensuring privacy is very important. This approach does so as either the database of known tags must be small or other information must be volunteered / known in order to reduce the search space. It prevents the totalitarian all readers track all the tags game.

IV. STRONG CRYPTOGRAPHY

The starting place for selection of a suitable keyed hash or block cipher primitive is to determine the required strength. A brute-force attack strength of 2^{64} is selected as a suitable design strength. Thus Nt and Nr must both be 64-bits. Concatenation gives 128-bit block size. To avoid collisions order 2^N a key length of $2N$ is needed, i.e. 128 bits. To prevent lookup table creation of low order, the authentication code needs to be 64-bits and then only part (i.e. half) the response. Thus a low resource, accepted as cryptographically strong, primitive with 128-bit block size and 128-bit key is sought. The obvious choices are SHA [22] and the AES [23], recent work [24], [25] has shown the AES is lower resource than SHA. The authors have previously designed and fabricated an even lower resource implementation of the AES on $0.13\mu m$ CMOS [26], [27] which is believed to be the first to consume less than a microwatt. Measured results for this design showed a 1411% power-latency-area performance improvement over the previous state-of-the-art chip [28]. Table III shows comparison to the previous work of other and that of ourselves.

TABLE III
COMPARISON OF AES ASIC DESIGNS

Design	Type	Tech, μm	Power, μW	Area, kgates	Latency	Efficiency, P-A-T μJ-gates
Kuo [29]	Chip	0.18	56,000	173	12cyc/154MHz =77.9ns	54mW*173k*77.9 =754
Feldhofer [28]	Chip	0.35	4.5	4.4	1032cyc/100kHz =10.32ms	4.5uW*4.4k*10.32ms =204
Hsiao [30]	Synth	0.18	34,000	15	10cyc/104MHz =96.2ns	34mW*15k*96.2ns =49
Kaps [24]	Synth	0.13	20.23	4.1	534cyc/500kHz =1.07ms	20.23uW*4.1k*1.07ms =89
Lin [31]	Synth	0.13	40,900	86.2	10cyc/333MHz =30.0ns	40.9mW*86.2k*30ns =106
Hsai [32]	Synth	0.18	5.3	11.277	cycles not stated	insufficient data to calculate
Ricci [33]	Synth	0.18	2.1 (@0.6V)	6	cycles not stated	insufficient data to calculate
Kim [25]	Synth	0.25	not stated	3.868	870cyc/10MHz =87us	insufficient data to calculate
Our work [26]	Chip	0.13	0.692 (@0.7V)	5.5	356cyc/100kHz =3.56ms	692nW*5.5k*3560us =13.5
This work	Chip	0.18	2.76 (@1.8V)	4.7	356cyc/100kHz =3.56ms	2.76uW*4.7k*3560us =46.2

The security analysis for the system is as follows: a legitimate verifier knows a list of possible IDs and is seeking to verify that the tag has one of these. Two challenge-response cycles provides authentication to 2^{64} against counterfeiting. The attacker has no control over Nr but still has a number of attack options all at least $O(2^{64})$:

1) store a partial table of $(Nr, Nt|X)$ for a tag, a time memory trade-off, attempting to send correct response to Nr twice $O(2^{32\times 2})$

2) brute force test all IDs $O(2^{128})$
3) attempt key recovery attack on AES (only half of X is known) at least $O(2^{64})$ [34], [35]
4) respond with random auth code twice $O(2^{64\times 2})$

The tag once programmed is write protected (or could even have the key uniquely defined during manufacture eg laser writing) thus long term DoS attacks are not possible. It is an essential part of the system security that the tag IDs be assigned from a set of uniform random numbers.

Anonymity can be tested against an adversary who does not have prior knowledge of the ID. The best attack is to choose a fixed Nr, however the tag generates its own Nt thus both Nt and (part)X appear random $O(2^{64})$. The adversary gains two random numbers and can only attempt AES key recovery as the best attack $> O(2^{64})$.

The totalitarian sub-game on first inspection appears somewhat easier in that the list of tags, length M ($M \ll 2^{64}$) is known, however, the reader must do work $O(M)$ to recover the ID for each tag. Thus to recover all tags $O(M^2)$ per reader, say there are coincidentally $M^{0.5}$ readers (a conservative assumption) then total work is at least $O(M^{2.5})$. So for an M of 50 million then approx $O(2^{64})$.

There remains the possibility of a radio relay in which the attacker shares a covert radio link with a legitimate tag thus gaining an assumption of possession at a distance. This cannot be countered by cryptographic means and instead is protected by using either screened reader enclosure or additional factor of identification.

V. Random Number Generator

There are very many software methods for generating random numbers however their "goodness" depends on the application. For cryptographic security a random number generator must be both unpredictable and uniform. There are two main sets of tests currently used for such random number generators, Diehard [36] and NIST [37], used to provide a measure of confidence for uniform random key generation. It must be remembered that such tests are only a guide, as for example, running a strong cipher such as the AES in say counter mode starting with all zero IV and an all zero key will generate a output bit string which would pass all these statistical tests, however would be wholly predictable. It is argued that for a random generator used to protect privacy, unpredictability is the most important feature, for which there are no specific mathematical tests and small non-uniformity in statistics may be tolerated as part of a compromise where power and area are critical.

RNGs are defined as pseudo random number generators (PRNG) or true random number generators (TRNG). To be of use for cryptography deterministic PRNGs must have an internal state which is undeterminable by an attacker.

To meet our design requirements for battery-less RFID the generator cannot store its current state in the NVRAM, the generator must reach the random state within 10's of milliseconds and use very little power (at most a few microwatts). Many generators can take some considerable time to accumulate sufficient entropy to reach a random state.

Hardware random number generators, rely on random processes in the physical world, such as thermal noise and chaos. Unfortunately, many such processes generate non-uniform statistics, examples include Gaussian noise and 1/f 'noise' from the quantum nature of the electron. Frequently a 'corrector' circuit (PRNG) to compensate for non-uniform behavior of the physical world is necessary.

Table IV presents a summary of existing methods used for hardware random number generators, together with the most applicable reference for low-frequency battery-less RFID. As can be seen from the table, none were found to be suitable for this application. The closest was the design of Balachandran et al, however, this made use of the UHF RF as a high frequency oscillator which is not applicable to a 125kHz RFID.

TABLE IV
HARDWARE RNG APPROACHES

Citation	**Type**	**Tech,** μm	**Power,** μW	**Area,** mm^2	**Latency, until random**	**Bit rate**
Amplified Noise, Petrie [38]	chip	2	3.5mW @100kHz	1.5	not stated	1.4 Mbps
Oscillator bank [39], Schellekens [40]	FPGA	0.13	not stated	973 slices (∼6k GE)	not stated, low	40 Mbps
Metastability, Hollerman [41]	chip	0.35	$2.92\mu W$	0.031	200 seconds	500 bps
NL analog chaos map, Zhou [42]	sim	0.18	3.025mW	not stated	not stated	20 Mbps
LF-HF clock jitter, Bucci [43]	chip	0.18	2.3mW	0.0016	not stated, low	10 Mbps
LF-HF clock, Bucci [44]	chip	0.09	$240\mu W$ @1.2V	0.006	not stated, low	1.74 Mbps
UHF carrier+LF osc, Balachandran [45]	chip	0.13	$0.53\mu W$ @1.2V	0.0056	not stated, low	320 kbps
Our design	chip	0.18	$3.36\mu W$ @1.8V	0.001 (110 GE)	$<$ 20ms	3.125 kbps

For this application, a random bit is required approx. every 500μs (2kbps). The available clock is 125kHz, attempting to generate a very low frequency oscillator at 2kHz would require relatively large components and not be viable. The alternative is to generate an oscillator $\gg$125kHz however if running continuously would consume much power. A second engineering issue arises from the weak power supply which may provide a convenient mechanism for the slow and fast oscillators to lock together. This problem is overcome using free running fast oscillators which are enabled only during the transitional periods of the low frequency clock.

The aims for the generator are to provide near uniform and unpredictable random Nt soon after power up and continue to do so to prevent an attacker from obtaining a fixed (but encrypted) identifier thus defeating the anonymity game. Conversely the security relies on the generator within the reader which is not so resource critical. This is a somewhat weaker requirement than an influence free truly uniform distribution

mandated for random key generation.

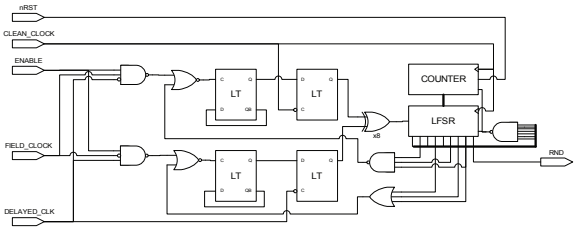

Fig. 1. Circuit diagram of random generator

Two gated high speed oscillators are used, with a free running frequency of ∼2GHz. The gating functions being defined such that one oscillator conditionally runs during the rising edge of the (relatively slow unstable) clock and the second the falling edge. Both operate approximately 1/8 cycles with the possibility of both operating during the same cycle. Both oscillators never run at the same instant in time. Further closely placed latches are used to minimize power consumption and prevent meta-stability which may otherwise incur an additional power penalty. These outputs are then combined with a feedback polynomial in a linear feedback shift register. To prevent adverse statistics from the all-zero state a counter is used to restart the generator uniformly.

The testing of a hardware random number generator needs to be made under normal operating conditions, for this application at only a few transactions per second, it presents a practical problem in collecting the approx 11Mbytes of data required by the test suites. For this system, it takes approx 3 days to acquire sufficient data to assess the continuously powered operation. However for the more usual scenario of the tag powering down between each series of a small number of challenges takes considerable longer, approx 3 weeks allowing sufficient powered-off period to avoid memory remnance.

To date the RNG in Fig. 1 passes 14/15 of the DIEHARD tests, failing part of the count-the-ones test. Further refinement of the generator and testing is still a work in progress. One option, we have already tested, which passed all the tests was to feed the (not quite uniform) random bits into the AES key and plaintext inputs and perform the encryption operation. As the AES engine is already present do does not increase the area and only adds 2.8ms to the response time.

VI. Prototype System

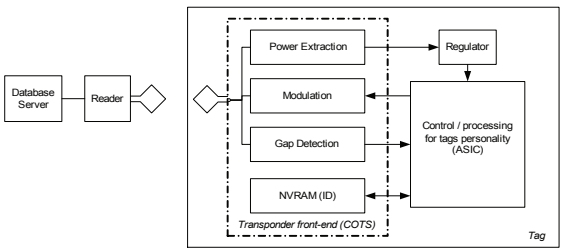

Fig. 2. Block diagram of system

In order to control cost whilst demonstrating the practicality of a strong cryptographic protocol on a low frequency, 125kHz, battery-less RFID, an off the shelf front end and NVRAM integrated circuit was used [46]. This is shown in Fig. 2. In a monolithic implementation the integration would remove the need for many of the I/O drivers and further reduce the total power consumption.

The air interface for the module has been defined to be minimalist. It uses on-off keying average bit rate RF/27 for reader to tag communication and Manchester modulation at RF/16 (data rate of RF/32) for the return channel. The tag acts as a slave to the reader and processes four commands to completely define the protocol and permit tag programming.

The tags configuration register in NVRAM (if write enabled) may be updated with a CFG(m) command to clear the write-enable status, set the operating mode for the random number generator and anti-collision on read mode.

A second command KEY(k) if the tag is write-enabled permits modification of the tag's 128-bit key in NVRAM. The IV(Nr) command supplies the tag with the readers 64-bit random and triggers the tag to perform its cryptographic operation (the tag has already generated a random, Nt):

$$X = AES\,(key, Nr|Nt) \tag{1}$$

The tag then transmits half of X as the authentication code together with Nt (total 128-bits). This is encapsulated between synchronization tokens and repeated until the SILENCE command is received (or power is lost). At which time the tag resets refreshing its random, Nt. In anti-collision mode the reply is punctuated by periods of silence of between 1 to 16 message periods. Repetition of the same message to a challenge is helpful for environments attempting to read a number of tags within the same time frame. It is also possible then to be extended to the classical singulation methods (eg tree-walking). An ASIC has been designed and fabricated on $0.18\mu m$ CMOS to interface to the front end and integrates a random number generator, AES crypto primitive, modem, NVRAM interface, controlling protocol and clock management circuitry (fig. 3).

The use of a low frequency RF (sinusoid) as a clock combined with a relatively high impedance power source results in a need to 'clean-up' the clock to prevent unintended clock transitions as the slow rising edge approaches the threshold voltage due the varying current demands of the on-tag logic. This is done using a delayed version of the clock created using a simple RC delay and the circuit shown in fig. 4.

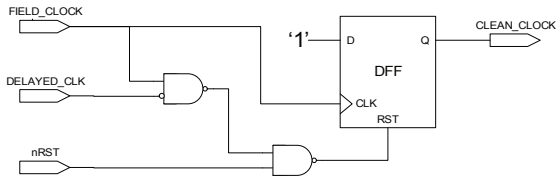

Fig. 4. Clock cleaning circuit

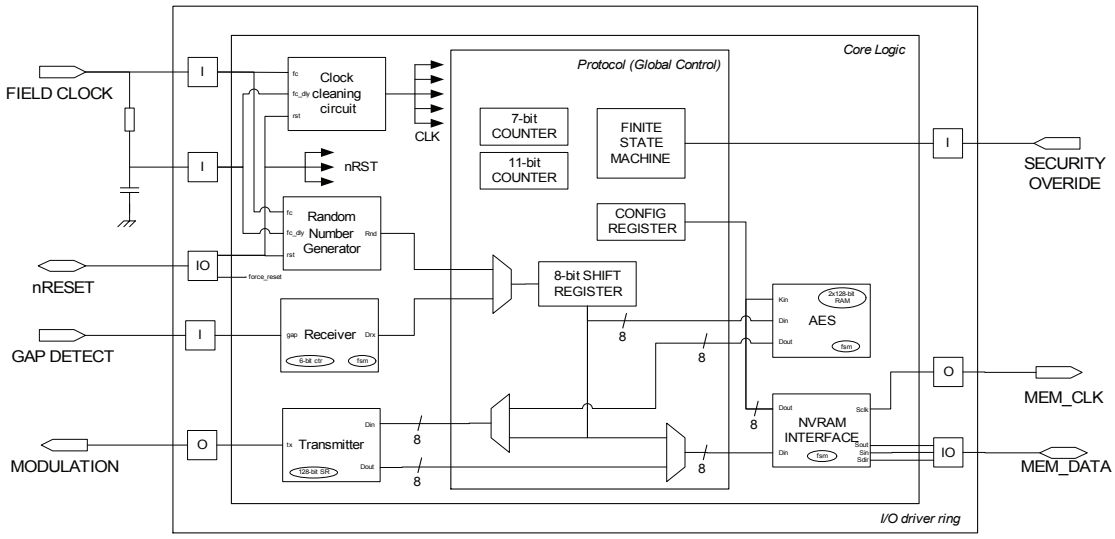

Fig. 3. Block diagram of ASIC

The protocol requires a source of random bits which may also be conveniently generated from the poor clock source using the random number generation circuit described in Section V. The generator is only enabled when required, to conserve power. The NVRAM has a bidirectional serial interface for read and write commands this is converted to a more conventional 8-bit RAM style interface by the interface module shown in Fig. 3. The NVRAM is used to store a configuration word (read on reset into the tags configuration register) and the tag's ID.

The receive (Rx) module decodes the OOK data sent by the reader and passes it to the protocol controller for interpreting the commands. The controller includes timeouts to prevent the tag from locking up due to communication errors. The authors low power 8-bit design for AES encryption is keyed using the tags ID and is used to 'hash' the random number generated in the tag and the IV sent by reader to create the required authentication code. This design uses a single 8-bit implementation for SubBytes and requires 356 cycles (inclusive of key and data I/O) to perform AES encryption whilst maintaining very low power consumption [26], [27].

The random and auth code are loaded into a 128-bit register for transmission. This simplifies extension to multi-tag environments. The transmit (Tx) module encodes responses using Manchester coding and serially outputs this modulation to the antenna.

The design was described using VHDL and synthesized, placed, routed and taped-out using Cadence tools. As with most designs on deep-sub micron processes it is routing limited. After cell placement and routing the back-annotated netlist was simulated using ModelSim and validated as a system using a behavioral model for the rest of the system and against known test vectors. Modeled power results were obtained using the system model to generate switching activity together with extracted layout parasitics. The area results, post layout including clock tree, are expressed using the process independent measure of NAND Gate Equivalent (GE).

Table V shows the modeled power consumption and table VI the timing results for a typical challenge-response cycle. These are reinforced with actual measurements later in this paper. For comparison the relatively lengthy write times and power consumption for EEPROM makes the total time to receive and write a new tagID (128-bit key) 330ms. This validates the assumption to avoid NVRAM writes during normal challenge-response operation.

TABLE V
MODELED RESULTS, BIAS 1.8V 125KHZ CLOCK

Module	**Power, μW**	**Area, GE**
Controller	0.50	899
RandGen	3.36 (34%)	110
Crypto(AES)	2.76 (28%)	4655 (56%) (2x128bit mem 2700 GE)
TxUnit	1.06	1481 (128bit tx reg 1350 GE)
The rest	2.29	1115
TOTAL	9.97	8260

TABLE VI
TIMING FOR CHALLENGE-RESPONSE CYCLE

State	**Time, ms**	**Notes**
key from NVram	2.8	direct to AES module
generate random	33 (38%)	
receive IV	11.1 (min)	
	14.3 (typ)	2+64 bits
	17.4 (max)	
crypto (AES)	2.7	356-16 cycles
transmit auth	33.8 (39%)	4+64+64 bits
receive silence command	0.8	2+2 bits
TOTAL	84.2 (min)	
	87.4 (typ)	11 Hz
	90.5 (max)	

The core area of the ASIC is 397 by 395μm (0.157mm^2).

This is surrounded by the power rings, I/O driver cells and pads. The design has 4 power pads, $4\times$Inputs, $2\times$Outputs and $2\times$Bidirectional pins. The total chip size is $<1mm^2$. The layout and a micrograph of a manufactured chip packaged in SOIC16 are shown in fig. 5.

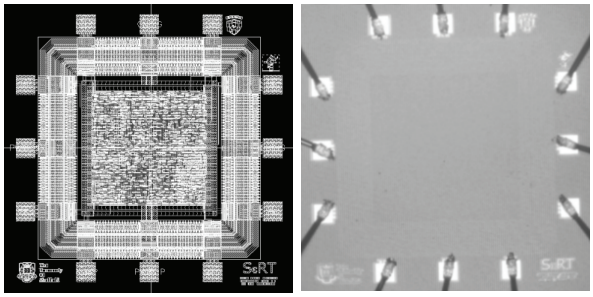

Fig. 5. CAD layout and die micrograph $0.18\mu m$ CMOS

A credit-card size prototype has been assembled using a wound wire antenna and a small PCB containing the secure RFID transponder (SϵRT) ASIC (this work), COTS transponder front end [46] and a 1.8V regulator. This is shown in Fig. 6.

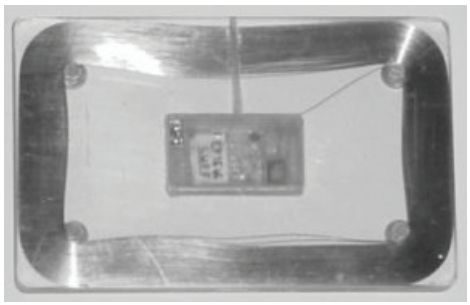

Fig. 6. Prototype battery-less secure tag

A simple reader was constructed using off the shelf components and controlled using a PC to test the system. A number of different tests have been performed including a week continuous operation. Overall, the tag responded to 99.75% of challenges by the reader. The challenge-response cycle (including 9600 baud serial communication to the PC, database lookup and comparison) on average could be performed 6.28 times per second.

The measured performance results for the prototype are tabulated in table VII. The power results are for standard process options, the low power option could not be selected due to incompatibility with other designs in the multi-project wafer.

Table VIII compares the results against previous attempts to produce a battery-less RFID which makes use of the AES for security. The power figures shown are for the digital baseband part of the design. The previous works have concentrated on the UHF band as this affords longer range and less demanding in terms of cycle count for the AES.

TABLE VII
MEASURED RESULTS

Dimension	Parameter	Value
RF	centre frequency	126.2 kHz
Area	core dimensions	397 x 395 μm
	core area	0.157 mm^2
	chip dimensions	956 x 956 μm
	chip area	0.914 mm^2
Power	core, RNG off	1.19 μW
(0.8Vcore)	core, RNG for IV	1.36 μW
Power	core (RNG off)	6.4 μW
(1.8Vcore)	core (RNG for IV gen)	9.6 μW
	core (RNG IV+noise)	11.1 μW
	Demand on front end @ 2.8V	138 μW
Time	IV comms	14.3 ms
	tag computation	2.68 ms
	Auth comms	33.4 ms
	Transactions (whole system)	6.28 Hz

TABLE VIII
COMPARISON WITH OTHER RFID DESIGNS USING AES

Citation	Type	Band	Tech	Power (core voltage)
Man [20]	synth	UHF	$0.18\mu m$	$4.7\mu W$(1.8V)
Hsai [32]	synth	UHF	$0.18\mu m$	$5.7\mu W$(1.5V)
Ricci [33]	synth	UHF	$0.18\mu m$	$2.5\mu W$(0.6V)
This work	chip	125kHz	$0.18\mu m$	$1.36\mu W$(0.8V)

VII. CONCLUSION

This is believed to be the first real demonstration of a passive (battery-less) RFID using the AES in a secure authentication protocol whilst maintaining a good notion of privacy. The tag has a measured average core power consumption of $1.36\mu W$ when operated in its normal mode at 125kHz with a bias of 0.8V.

It is argued that to achieve both security and privacy a tag must contain both an established secure strong cryptographic primitive and an unpredictable random source. To support one-time-programmable (OTP) tags it is highly desirable to avoid needing retained state variables (i.e. avoid writing to NVRAM); this effectively excludes PRNGs which must maintain their internal state when the tag is not powered. Thus an on-tag TRNG with relatively low latency and low power consumption is required.

It is further argued that mutual authentication is not a requirement for security and privacy, merely a lesser requirement of trust in own random number generation is needed.

It should be noted that the challenge-response cycle time is dominated by data transmission times together with on-tag random number generation. Similarly, random number generation tops the power table 34% followed by the AES at 28%. It is hoped that this will finally curtail statements such as "the AES is too heavy for RFID".

ACKNOWLEDGEMENT

This work was supported in part by a Doctoral Training Award from EPSRC. The authors wish to thank Jo Spreutels and Erwin Deumens of IMEC and Lisa Wong and Mark

Wilmott at the Microelectronic Support Unit, Rutherford-Appleton Laboratory for assistance with the design flow and foundry service.

REFERENCES

[1] T. Phillips, T. Karygiannis and R. Kuhn, *Security Standards for the RFID Market*, IEEE Computer Society, Security and Privacy, pp 8589, 15407993/05, 2005.

[2] EPCglobal Inc, [Online] Available: www.epcglobalinc.org

[3] The RFID Journal, [Online] Available: www.rfidjournal.com

[4] R.J. Anderson and M.G. Kuhn, *Low cost attacks on tamper resistant devices*, in Proc. Security Protocols Workshop, NewYork, LNCS, vol. 1361, pp 125136, Springer-Verlag, 1997.

[5] A.R. Peslak, *An Ethical Exploration of Privacy and Radio Frequency Identification*, Journal of Business Ethics 59: 327345, Springer, 2005.

[6] V. Lockton and R.S. Rosenberg, *RFID: The next serious threat to privacy*, Ethics and Information Technology 7:221231, Springer, 2006.

[7] S.L. Garfinkel, A. Juels and R. Pappu, *RFID Privacy: An Overview of Problems and Proposed Solutions*, IEEE Security & Privacy, May/June 2005.

[8] A. Juels, *RFID Security and Privacy: A Research Survey*, IEEE J. on Selected Areas in Communications, vol. 24 no. 2, pp 381394, invited paper, Feb 2006.

[9] Article-29 Data Protection Working Party, *Working document on data protection issues related to RFID technology*, WP 105, European Commission, Internal Market Directorate-General, Office No C100-6/136, Jan 2005.

[10] S.A. Weis, S.E. Sarma, R.L. Rivest and D.W. Engels, *Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems*, Security in Pervasive Computing 2003, LNCS, vol. 2802, pp 201212, Springer, 2004.

[11] M. Ohkubo, K. Suzuki and S. Kinoshita, *RFID Privacy Issues and Technical Challenges*, Communications of the ACM, Vol. 48, No. 9, pp 6671, Sept. 2005.

[12] Consumers Against Supermarket Privacy Invasion and Numbering (CASPIAN) [Online] Available: www.nocards.org, 2003.

[13] M. Lehtonen, T. Staake, F. Michahelles and E. Fleisch, *From Identification to Authentication A Review of RFID Product Authentication Techniques*, RFIDsec06, Graz Austria, July 2006.

[14] S. Engberg, M. Harning and C. Damsgaard-Jensen, *Zero-knowledge device authentication: Privacy & security enhanced RFID preserving business value and consumer convenience*, Conf. on Privacy, Security and Trust (PST), New Brunswick, Canada, Oct. 2004.

[15] C. Chatmon, T.v. Le and M. Burmester, *Secure anonymous RFID authentication protocols*, Technical Report TR-060112, Florida State University, Department of Computer Science, Tallahassee, Florida, USA, 2006.

[16] T. Dimitriou, *A Lightweight RFID Protocol to protect against Traceability and Cloning attacks*, IEEE SecureComm05, Sept 5-9, Athens, Greece, Sept 2005.

[17] J. Yang, J. Park, H. Lee, K. Ren and K. Kim, *Mutual Authentication Protocol for Low-cost RFID*, ECRYPT Workshop on RFID and Lightweight Crypto, Graz, Austria, July 14-15, 2005.

[18] G. Tsudik, *YA-TRAP: Yet Another Trivial RFID Authentication Protocol*, IEEE Intl. conf. on Pervasive Computing and Communications (PerCom06), Pisa, Italy, March 2006.

[19] Z. Kfir and A. Wool, *Picking virtual pockets using relay attacks on contactless smartcard systems*, [Online] Available: http://eprint.iacr.org/2005/052, 2005.

[20] A.S.W. Man, E.S. Zhang, V.K.N. Lau, C.Y. Tsui and H.C. Luong, *Low Power VLSI Design for a RFID Passive Tag baseband System Enhanced with an AES Cryptography Engine*, 1st Annual RFID Eurasia conf., 5-6 Sept. 2007, pp 1-6, 2007

[21] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, *Strong Authentication for RFID Systems Using the AES Algorithm*, CHES 2004, LNCS 3156, pp. 357370, 2004.

[22] National Institute of Standards and Technology (NIST), *Secure Hash Standard (SHS)*, FIPS 180-3, [Online] Available: http://www.itl.nist.gov/fipspubs/, June 2007.

[23] NIST, *Advanced Encryption Standard (AES)*, FIPS 197, [Online] Available: http://www.itl.nist.gov/fipspubs/, Nov. 2001.

[24] J-P. Kaps and B. Sunar, *Energy Comparison of AES and SHA-1 for Ubiquitous Computing*, in proc. Embedded And Ubiquitous Computing (EUC06), Seoul, Korea, pp. 372-381, 1-4 Aug 2006

[25] M. Kim, J. Ryou, Y. Choi and S. Jun, *Low-cost Cryptographic Circuits for Authentication in Radio Frequency Identification Systems* IEEE Tenth Intl. Symp. on Consumer Electronics, (ISCE '06), pp. 1-5, 2006

[26] T. Good and M. Benaissa, *692nW Advanced Encryption Standard (AES) on 0.13um CMOS*, [to appear] IEEE TVLSI

[27] T. Good and M. Benaissa, *filed for encryption apparatus and method patent*

[28] M. Feldhofer, J. Wolkerstorfer and V. Rijmen, *AES implementation on a grain of sand*, IEE Proc. Information Security, Vol. 1, pp 13-20, 2005.

[29] H. Kuo, I. Verbauwhede and P. Schaumont, *A 2.29Gbits/sec, 56mW non-pipelined Rijndael AES encryption IC in a 1.8V 0.18um CMOS technology*, in proc. CICC, pp 147-150, 12-15 May, Orlando, Florida 2002.

[30] S-F. Hsaio, M-C. Chen and C-S. Tu, *Memory-Free Low-Cost Designs of Advanced Encryption Standard Using Common Subexpression Elimination for Subfunctions in Transformations*, IEEE TCAS-1, Vol. 53, No.3, pp 615-626, March 2006

[31] S-Y. Lin and C-T. Huang, *A High-Throughput Low-Power AES Cipher for Network Applications*, in proc. ASP-DAC 07, pp 595-600, 23-26 Jan, Yokohama, Japan, 2007

[32] M.L. Hsai, O.T. Chen, *Passive RFID transponder with power-aware encryption*, Midwest Symposium on Circuits and Systems, pp 838-841, 10-13 Aug, Knoxville USA, 2008.

[33] A. Ricci, M. Grisanti, I. De Munari, P. Ciampolini, *Design of a 2 μW RFID baseband processor featuring an AES cryptography primitive*, IEEE ICECS, pp 376-379, 31 Aug 3 Sept, Malta, 2008

[34] S. Murphy and M.J.B. Robshaw, *Remarks on the Security of the AES and the XSL Technique*, Electronic Letters, vol. 39, pp. 3638, 2003.

[35] N.T. Courtois and J. Pieprzyk, *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*, ASIACRYPT02, Queenstown, New Zealand, LNCS 2501, pp. 267287, Springer, 2002.

[36] G. Marsaglia, *DIEHARD tests*, [Online] Available: http://www.stat.fsu.edu/pub/diehard/

[37] NIST, *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, SP800-22, [Online] Available: http://csrc.nist.gov/publications/PubsSPs.html, 2001

[38] C.S. Petrie and J.A. Connelly, *A noise-based IC random number generator for applications in cryptography*, IEEE TCAS-I, Vol. 47, Iss. 5, pp. 615-621, May 2000

[39] B. Sunar, W.J. Martin and D.R. Stinson, *A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks*, Transactions on Computers, vol. 56, Iss. 1, pp. 109-119, Jan. 2007

[40] D. Schellekens, B. Preneel and I. Verbauwhede, *FPGA Vendor Agnostic True Random Number Generator*, Field Programmable Logic and Applications (FPL '06), pp. 1-6, Aug. 2006

[41] J. Holleman, B. Otis, S. Bridges, A. Mitros and C. Diorio, *A 2.92uW Hardware Random Number Generator*, Proc. of the 32nd European Solid-State Circuits Conf. (ESSCIRC 2006), pp. 134-137, Sept. 2006

[42] T. Zhou, Z. Zhou, M. Yu and Y. Ye, *Design of A Low Power High Entropy Chaos-Based Truly Random Number Generator*, IEEE Asia Pacific Conf. on Circuits and Systems (APCCAS 2006), 4-7 Dec. 2006, pp. 1955-1958, 2006

[43] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti and M. Varanonuovo, *A High-Speed Oscillator-Based Truly Random Number Source for Cryptographic Applications on a Smart Card IC*, IEEE Transactions on Computers, vol. 52, no. 4, April 2003

[44] M. Bucci, R. Luzzi, *Fully Digital Random Bit Generators for Cryptographic Applications*, IEEE TCAS-I, vol. 55, no. 3, pp 861-875, April 2008

[45] G.K. Balachandran and R.E. Barnett, *A 440nA True Random Number Generator for Passive RFID Tags*, IEEE TCAS-I, vol. 55, no. 11, December 2008

[46] Atmel Inc, *U3280 100-150kHz transponder interface datasheet*, [Online] Available: http://www.atmel.com/dyn/resources/prod_documents/doc4688.pdf