Improved (and Practical) Public-key Authentication for UHF RFID Tags

Sébastien Canard[1], Loïc Ferreira[2], and Matt Robshaw[2]
firstname.lastname@orange.com

[1] Applied Cryptography Group, Orange Labs
42 rue des Coutures BP 6243, 14066 Caen Cedex, France
[2] Applied Cryptography Group, Orange Labs
38-40 rue de General Leclerc, 92794 Issy les Moulineaux, France

Abstract. CRYPTOGPS has been promoted as a public-key technology suitable for UHF RFID tag authentication. Since it is a classical *commitment-challenge-response* (CCR) scheme, it can be converted into a signature scheme using the transformation proposed by Fiat and Shamir. Previously this signature variant has not been considered for RFID, but in this paper we show how to achieve this transformation in a way that yields a compact and efficient scheme. Further, the three-pass CCR scheme is turned into a regular challenge-response scheme with the attendant protocol and implementation improvements. Since we use a block cipher rather than a hash function for the transformation, we justify our approach using results in the *ideal cipher model* and the net result is a variant of CRYPTOGPS that offers asymmetric UHF tag authentication with reduced communication and protocol complexity.

1 Introduction

The terms RFID, Internet of Things, sensor network, and pervasive computing are frequently used to indicate the anticipated widespread deployment of computationally limited devices. The difficulty of providing (reasonable) security on such devices, in a way that makes economic sense, is by now well-established.

The radio-frequency identification (RFID) tag is a particularly interesting case, with billions of tags in deployment, and while it is an over-simplification, two main operating frequencies are of particular interest. The short-range HF tag (13.56 MHz) is used, for instance, in public-transport applications and underpins the area of *Near Field Communication (NFC)*. However, tags that operate over UHF (860-960 MHz) [10] are cheaper, smaller, and can be read at a distance and it is typically these devices that one has in mind when discussing RFID and cryptography (since advanced standardised cryptography on an HF tag is readily available). One particularly interesting application when using cheap UHF tags is that of *product authentication* and there are many proposals to use cheap UHF RFID tags as part of an anti-counterfeiting solution [1,25,27]. Among the different approaches that might be used, it is dynamic cryptographic tag authentication that offers the greatest long-term promise. But since UHF tags are a demanding implementation environment, it is not so straightforward to identify particularly efficient cryptographic algorithms.

2 Cryptography and RFID tags

The challenging physical constraints posed by RFID tags have been a significant spur to cryptographic research. Perhaps most success has been in the field of symmetric cryptography where we now have a range of block ciphers including PRESENT [4] and a range of stream ciphers [41] that might be suitable for UHF RFID deployment. Over the years some of these may feature in products; indeed some such as PRESENT already appear in ISO standards [23].

The field of asymmetric encryption is less clear. It could be that a symmetric-key solution works well enough, but that the kind of supporting key infrastructure that is required is somewhat at odds with the typical RFID model. Taking the supply chain as an example, millions of tags will be attached to products by a manufacturer with products being distributed to shops and customers worldwide. Ensuring the right key is available to the right reader at the right time is not trivial.

There is therefore considerable interest in any asymmetric solution that might yield a more flexible supporting key infrastructure. Unfortunately, since the typical algorithms from Internet and PC applications are not at all suited to UHF RFID tags, there are not so many alternatives. However, there has been some renewed interest in what are termed *commitment-challenge-response* (CCR) schemes, since these allow lightweight tag authentication. Among them is CRYPTOGPS.

2.1 CRYPTOGPS

The scheme CRYPTOGPS, due to Girault, Poupard, and Stern, is well-established in the cryptographic literature [12,16,33,40]. Several variants feature in ISO/IEC 9798-5 [21] while the most efficient variant, namely that based around elliptic-curves, is undergoing standardisation in ISO/IEC 29192-4 [24] which is devoted to asymmetric lightweight cryptography. Over the years several optimisations have been proposed [15,17] and the performance of the scheme has been studied by implementors [14,28,29,38].

The essential form of CRYPTOGPS using the typical optimisations one might expect to use is given in Figure 1. In implementation papers, the PRNG is typically instantiated using the lightweight block cipher PRESENT [4] in an appropriate mode of use and the most accurate (post-fabrication) implementation figures [38] show that all the on-tag cryptographic components can be implemented in around[3] 2800 GE with a processing time of around 720 cycles.

The small area required for CRYPTOGPS is due to one property and one optimisation. The property is that no modulo arithmetic is used on the tag. All integer computations are regular addition and multiplication. The optimisation comes in the form of coupons that contain the results of a pre-computation; this avoids the need to support elliptic curve operations on the tag. Certainly limited-use tokens are familiar in a wide range of applications from pre-paid telephone cards to public transport ticketing. However, their use is not to everyone's taste and they are not suitable for all use-cases.

[3] It is typical to use the *gate equivalent (GE)* to compare implementations. The physical area is divided by the size of a `nand` gate to give a broadly technology-neutral estimate of its size. While not perfect, it remains sufficiently useful.

Tag		Reader
	PARAMETERS	
Curve $\mathcal{C}$, point P		Curve $\mathcal{C}$, point P
	KEYS	
Secret key *(sk)* $s \in_R \{0,1\}^\sigma$ Secret key $k \in_R \{0,1\}^\kappa$		Public key *(pk)* $V = -sP$
	COUPON PRE-COMPUTATION WITH PRNG	
For $0 \leq i \leq n-1$ Let $r_i = \text{PRNG}_k(i)$ where $\lvert r_i \rvert = \rho$ Set $x_i = \lceil \text{HASH}(r_iP) \rceil_t$ Store coupon x_i		
	PROTOCOL USING ON-TAG PRNG	
At time i fetch x_i	$\xrightarrow{x_i}$	
	$\xleftarrow{c_i}$	Pick $c_i \in_R \{0,1\}^\delta$
Generate $r_i = \text{PRNG}_k(i)$		
$y_i = r_i + (s \times c_i)$	$\xrightarrow{y_i}$	$\lceil \text{HASH}(y_iP + c_iV) \rceil_t \overset{?}{=} x_i$

Fig. 1. The typical description of CRYPTOGPS using the most common implementation optimisations, where PRNG is a pseudo-random generator, HASH is a hash function, and where σ, κ and t and δ are security parameters that will be discussed further (see Section 5.1).

However, this is not the focus of the paper and issues around the use of coupons, a topic that is broader than their use with CRYPTOGPS, are discussed in the Appendix. Instead, we will be concerned with the well-known Fiat-Shamir conversion [11] of a basic CCR scheme into a signature scheme. And our goal is to make this conversion, and the resultant scheme, more computationally efficient than was previously recognised.

2.2 This paper

It is well-known that an interactive identification scheme can be converted into a digital signature scheme [11,31] and the security provided by this conversion was proved by Pointcheval and Stern [36,37]. Indeed two signature variants of CRYPTOGPS have already been standardised within ISO/IEC; details are available in ISO/IEC 14888-2 [22].

Classically the tool to perform this conversion is a hash function HASH. In general terms, the commitment x_i from the original CCR scheme is combined with the message m to be signed using a hash function, $\text{HASH}(x_i, m)$. The output from $\text{HASH}(x_i, m)$, or part of it, is then used as the challenge c_i. Some previous work in the literature has tried to establish the implementation profile of such a scheme when using CRYPTOGPS [30] but this only confirms its unsuitability for UHF RFID tags, at least in this classical form. To change this view we need something new.

As a first step we observe that ISO/IEC 14888-2 makes a distinction between two types of signatures. The first is referred to as a *transmissible signature*; that is a digital signature that can be verified by a third party at any time. The second type of signature is referred to as a *non-transmissible signature* and is used solely in a dynamic setting. Here the "message" to be signed comes from the verifier. The prover (or tag) computes the signature on this message and returns the result. The verifier can set a time-limit, or time-out, to fix the amount of time that is available for the tag to respond. If the tag responds in time (with a valid signature) then the verifier is convinced that the tag is genuine. However, the verifier would be unable to convince a third-party of this unless he can further guarantee that the signature was computed within a certain time on a fresh challenge. Nevertheless we have what we want; we have a dynamic tag authentication scheme that means an interrogator can be certain a tag is genuine.

While helpful, we would still have a proposal that is too large for UHF RFID deployment. The technical contribution of this paper, therefore, is to find a more efficient (and secure) way of making the CCR-to-signature conversion. In this paper we outline a full solution with preferred parameter sets. In fact it is only by replacing the function HASH that we can derive a practical scheme. This, along with the opportunity to use pre-existing components on the tag, allows us to move to a new improved variant of CRYPTOGPS with only a moderate increase in area on the tag. And we reap the operational advantage that the scheme now becomes *challenge-response* instead of *commitment-challenge-response*, improving both the communication burden and the system complexity.

To provide context, we note that various papers consider the practicability of implementing elliptic curve schemes on RFID tags. These suggest that the area needed to implement elliptic curve operations is in excess of 13 000 GE and the time to process an operation requires several tens of thousands of cycles, *e.g.* [48,47]. These numbers are markedly greater than what one could expect from an implementation of the non-transmissible signature variant of CRYPTOGPS with the coupon optimization (see Appendix). This paper relies on the use of a hash function based around a block cipher. This has been [45,46], and is likely to remain, an active area of research which may have some future bearing on the work considered in this paper.

3 Moving to Non-transmissible Signatures

The classical CCR-signature conversion requires the use of a hash function. Yet typical[4] hash functions are not at all suitable for UHF RFID tags [5]. Instead we would like to instantiate the conversion using a block cipher, particularly since one already implements PRESENT on the tag in support of CRYPTOGPS [38].

[4] Some lightweight hash functions have been recently proposed [2,6,18] but they are new and tend to have long processing times.

Tag		Reader
	PARAMETERS	
Curve $\mathcal{C}$, point P		Curve $\mathcal{C}$, point P
	KEYS	
Secret key *(sk)* $s \in_R \{0,1\}^\sigma$ Secret key $k \in_R \{0,1\}^\kappa$		Public key *(pk)* $V = -sP$
	COUPON PRE-COMPUTATION WITH PRNG	
For $0 \leq i \leq n-1$ Let $r_i = \text{PRNG}_k(i)$ where $\lvert r_i \rvert = \rho$ Set $x_i = \lceil \text{HASH}(r_i P) \rceil_t$ Store coupon x_i		
	PROTOCOL USING ON-TAG PRNG	
	$\xleftarrow{w}$	Pick $w \in_R \{0,1\}^\delta$
At time i fetch x_i Compute $c_i = \text{F}(x_i, w)$ Generate $r_i = \text{PRNG}_k(i)$		
$y_i = r_i + (s \times c_i)$	$\xrightarrow{y_i, c_i}$	Compute $x' = \lceil \text{HASH}(y_i P + c_i V) \rceil_t$ $\text{F}(x', w) \overset{?}{=} c_i$

Fig. 2. The non-transmissible signature variant of CRYPTOGPS for dynamic authentication. We propose that the function F be built around the block cipher PRESENT, see Section 3.1.

3.1 Choice of conversion function

The conversion of an three-pass identification scheme into a signature scheme has been well-studied in the literature. The first proposals by Fiat and Shamir [11] remain the foundation for this conversion and proofs of security followed when a more rigorous theoretical foundation had been established [36].

Our essential requirement is that the output of the conversion function, which we will denote F, is unpredictable for different inputs while the same inputs yield the same output. A simple and elegant way to construct F, while respecting the preferred parameters derived in Section 3.2 is to set

$$c_i = \text{F}(x_i, w) = \text{ENC}_{x_i \| w}(0^n)$$

where encryption is performed using the 128-bit key version of PRESENT [4].

Digression. There might be some interest in understanding how we arrived at this choice for F. Indeed, at first sight it is not clear that the *same* function F is required at both the tag and the reader and we could consider a protocol as follows:

PROTOCOL USING ON-TAG PRNG

$$\begin{array}{rcl} & \xleftarrow{\quad w \quad} & \text{Pick } w \in_R \{0,1\}^{\delta} \\ \text{At time } i \text{ fetch } x_i & & \\ \text{Compute } c_i = \mathrm{F}(x_i, w) & & \\ \text{Generate } r_i = \mathrm{PRNG}_k(i) & & \\ y_i = r_i + (s \times c_i) & \xrightarrow{\quad y_i, c_i \quad} & \text{Compute } x' = \lceil \mathrm{HASH}(y_i P + c_i V) \rceil_t \\ & & \mathrm{VERIF.}(x', w, c_i) \overset{?}{=} \mathrm{TRUE} \end{array}$$

In this case, some hypothetical candidates for F might include:

$\mathrm{F}(x_i, w)$	$\mathrm{VERIF.}(x', w, c_i)$
$c_i = \mathrm{ENC}_{x_i \Vert w}(x_i)$	$\mathrm{DEC}_{x' \Vert w}(c_i) = x'$
$c_i = \mathrm{ENC}_{x_i \Vert w}(w)$	$\mathrm{DEC}_{x' \Vert w}(c_i) = w$
$c_i = \mathrm{ENC}_{w}(x_i)$	$\mathrm{DEC}_{w}(c_i) = x'$
$c_i = \mathrm{ENC}_{x_i}(w)$	$\mathrm{DEC}_{x'}(c_i) = w$

It can be easily seen that not all of these approaches are secure. Further, we concentrated our efforts on the simplest and most efficient-to-implement schemes. In turn, this matched the theoretical analysis presented in Section 4. In the remainder of the paper, therefore, F will refer to the following transformation:

$$c_i = \mathrm{F}(x_i, w) = \mathrm{ENC}_{x_i \Vert w}(0^n).$$

3.2 Setting parameter sizes

In this section we consider some attacks that help us better understand the trade-offs between different parameters. As a baseline, however, we assume that all the secret keys held on the tag, both for PRESENT and CRYPTOGPS, have a length that is intended to provide 80-bit security.

For all challenge-response protocols there are some basic on-line attacks, *i.e.* without any pre-processing. These attempts to fool the reader into accepting a fake tag as genuine and have a certain probability of success at each run of the protocol.

1. The attacker chooses random y_i and c_i and sends these as a response. The probability that $\mathrm{F}(\lceil \mathrm{HASH}(y_i P + c_i V) \rceil_t, w) = c_i$, where w was sent by the verifier, is given by $2^{-|c_i|}$ so the probability of success is related to the size of c_i.
2. The attacker picks x_i at random and computes c_i on receiving w. The attacker then randomly chooses y_i and sends the response. He will be successful if $\lceil \mathrm{HASH}(y_i P + c_i V) \rceil_t = x_i$. The probability of success is $2^{-|x_i|}$ and is related to the coupon size.

The net result of these attacks is to set $|x_i| = |c_i| \geq z$ if we are aiming for an impersonation probability less than 2^{-z}. This per-session forgery probability can be improved in an obvious way using off-line pre-computation and storage. Essentially, one uses either of the two techniques above to construct a valid and consistent $\{x_i, w, c_i, y_i\}$ quadruple, though only w, c_i, and y_i need to be stored. Using the first approach, computing d quadruples take an off-line work effort proportional to $d2^{|c_i|}$ operations while the second requires a work effort of $d2^{|x_i|}$ operations. The probability of success for each session then becomes $d2^{-|w|}$ since for d potential values of w the fake tag contains a good response.

A related result stems from the phenomenon of u-collisions, explored by Girault and Stern in two papers [13,17]. Since the coupons are constructed using a hash function it is possible that $\lceil \text{HASH}(y_iP + c_iV)\rceil_t = \lceil \text{HASH}(y_i'P + c_i'V)\rceil_t$ for two potentially different sets of inputs. This is the familiar hash function collision and the probability of finding a 2-collision is related to the birthday paradox. If we move to larger values of u then, depending on the size of x_i and the amount of computation devoted to an off-line attack, an attacker can expect to find some x_i for which u values of $\{y_i, c_i\}$ will hash to x_i. In more detail, he fixes y_i and searches over c_i storing the resulting x_i. With a work effort of, say, 2^{80} operations and coupons of size $|x| = 64$ he can expect to have a 2^{16}-collision for a given value of x_i and y_i. This then means that 2^{16} values of c_i will allow a forgery, and the probability of alighting on one of these values in practice—when w is chosen at random by the verifier—is $2^{16-|c|}$.

There are two complementary aspects to this observation. The first is that the size of the coupons has an impact on the success probabilities of both on-line and off-line attacks. The second is that, in practice, an attacker is unlikely to use 2^{80} operations to gain a per-session advantage. Since the same work effort can recover the long-term secret CRYPTOGPS key, which was chosen to offer 80-bit security, this latter attack would in fact be preferable. Indeed, since efforts to recover s are probabilistic [31], even a work effort significantly less than 2^{80} operations will have some probability of yielding the long-term secret key (and fully compromising the tag). So the cost-benefit for the attacker in devoting vast amounts of pre-computation to giving an advantage in a single run of the authentication protocol is not clear.

Finally, we observe that a device impersonating a genuine passive UHF tag might not, itself, be a passive UHF tag. It could be a tag that is connected via a relay to a much more powerful device, or it could even be self-powered; if the "tag" is not visually inspected, *e.g.* because it is inside a crate, then it could be anything. So after receiving w a false "tag" can choose/search x_i and y_i until finding values for which $\lceil \text{HASH}(y_iP + c_iV)\rceil_t = x_i$. Whether or not this is likely to be accomplished within a specified response time depends on the parameter values and the computational complexity of the emulating/remote device.

4 Security Foundations

The previous section was concerned with the practical aspects of setting parameter sizes. Here we consider the theoretical foundations of our preferred conversion method.

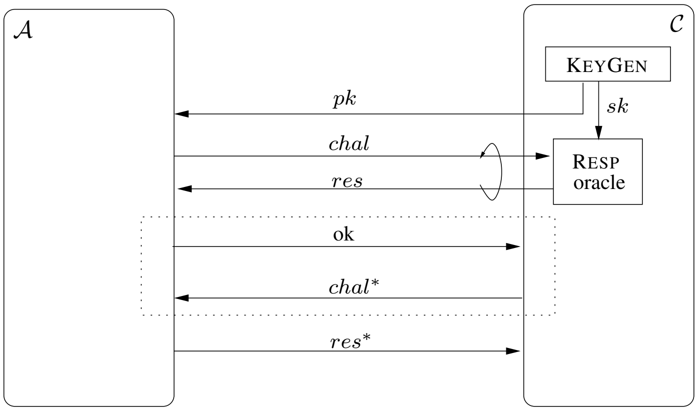

Fig. 3. Soundness: $\mathcal{A}$ wins if $\mathsf{Verif}(pk, chal^*, res^*) = 1$ and $(chal^*, res^*)$ does not come from the Res oracle.

For this it might be helpful to consider the different components of an (RFID) authentication scheme.

- KEYGEN: on input a security parameter λ, KEYGEN generates key pair (pk, sk), possibly certified by some certification authority CA.
- CHALL: the Reader generates a challenge $chal$, on input the public key pk of the Tag.
- RESP: the Tag uses sk and the challenge $chal$ to generate a string res.
- VERIF: on input pk, $chal$ and res, the Reader outputs a bit 0/1 to denote either reject or accept.

Clearly we require *correctness*, that is if (sk, pk) is output by KEYGEN and if res is computed using both sk and $chal = \text{CHALL}(pk)$, then $\text{VERIF}(pk, chal, res) = 1$ with overwhelming probability. The background to the security notion, *soundness*, is illustrated in Figure 3. A challenger $\mathcal{C}$ is matched against $\mathcal{A}$, an adversary against soundness. $\mathcal{A}$ can receive legitimate challenge-response pairs but is then required to reply to a previously unseen challenge. We say that an authentication scheme is secure if, and only if, the probability that the adversary $\mathcal{A}$ can provide a good response is negligible (in the security parameter).

The security of our proposal is substantiated in several steps.

From signature to authentication. Given a signature scheme, it is easy to design a 2-pass authentication scheme. The reader sends a challenge $chal$ and the tag produces the response res as a signature on the message $chal$. The verification procedure is given by signature verification. It is well known that such an authentication scheme is secure (*i.e.* sound) if the used signature scheme is unforgeable.

(In fact this can be seen from Figure 3 since the signature unforgeability experiment is similar to that described in Figure 3 where the response res^* is composed of both a challenge message w^* and the corresponding forged signature σ^*.)

The Fiat-Shamir transformation and the random oracle model. As we have seen before, the Fiat-Shamir heuristic [11] consists in replacing the random challenge c by the output of a hash function HASH taking as input the prover's commitment x_i and the message w: $c = \text{HASH}(x_i, w)$. Pointcheval and Stern proved [36,37] that the resulting signature scheme is unforgeable, assuming that the hash function HASH is a random oracle [3]. In a nutshell, a random oracle idealizes the hash function that behaves as a random function that gives unpredictable outputs (but the same input always gives the same output).

As said previously, hash functions are not suitable for UHF RFID tags and we should instead find something more interesting.

ICM. The *ideal cipher model* (ICM) is an idealized model [42] in which a random block cipher (seen as an ideal cipher) with an n-bit input/output and a κ-bit randomly-chosen secret key is computationally indistinguishable from a randomly chosen n-bit permutation. More formally, given an ideal cipher denoted $\text{ENC} : \{0,1\}^\kappa \times \{0,1\}^n \longrightarrow \{0,1\}^n$, an adversary having the possibility to make both encryption and decryption queries to the ideal block cipher, for any key, cannot distinguish a given output from that of a randomly-chosen permutation. In such a case, we can consider the ideal cipher ENC outputs as being those of a randomly chosen n-bit permutation.

As we want to use instead of a hash function the block cipher PRESENT, the use of an ideal cipher may help us to obtain a (proven to be) secure construction.

The Fiat-Shamir heuristic and the ideal cipher. Luckily, Coron *et al.* proposed in [9] a black-box transformation of any ideal cipher into a random oracle. Given an ideal cipher $\text{ENC} : \{0,1\}^\kappa \times \{0,1\}^n \longrightarrow \{0,1\}^n$ and the message $(w_1 \| \cdots \| w_\ell)$ to be hashed, the construction works as follows:

- set y_0 to 0^n (or to any fixed IV);
- for $i = 1$ to ℓ do $y_i = \text{ENC}_{w_i}(y_{i-1}) \oplus y_{i-1}$;
- output y_ℓ.

With a single block w_1 this corresponds to the computation $y_1 = \text{ENC}_{w_1}(0^n)$. Thus, given an ideal cipher ENC, one can replace the hash function/random oracle of the Fiat-Shamir transform by the above construction and the security proof immediately follows, with a security parameter corresponding to the block size n used in ENC.

We can next use this construction within the Fiat-Shamir heuristic, which corresponds to our construction, in the particular case of CRYPTOGPS.

Signatures and CRYPTOGPS. In fact, CRYPTOGPS was proven to be a secure zero-knowledge proof [40,16]. Thus we can directly apply the above results. If, on input w and commitment x_i, the tag computes challenge $c = \text{ENC}_{x_i \| w}(0^n)$ and response y, then the signature $\sigma = (c, y)$ is that of an unforgeable signature scheme and the authentication scheme described in Figure 2 is also secure.

5 Implementation Perspectives

While the conversion of an identification scheme to a signature scheme is well-known, previous work on CRYPTOGPS has avoided this. The main reason is the additional complexity of supporting the conversion function F.

However, in this paper we show that existing components can be re-used and we instantiate the ideal cipher with the 128-bit key version of PRESENT. The cipher has been analysed widely in the community [7,8,26,32,34,43,44] including under related-key attacks [35]. So while the CCR-to-signature conversion necessarily implies some overhead to the area that is required on the tag, these overheads are slight and can be predicted with a reasonable level of confidence. In return, the gain is significant in terms of system complexity as the three-pass CCR scheme is replaced by a simple challenge-response protocol.

5.1 Preferred parameter sets

It is typical in environments where computational devices are quite constrained to aim for a security level of "80 bits". Of course, if a greater security level can be comfortably accommodated then all the better. But this can compromise performance or even, in the worst case, mean security cannot be implemented at all on a passive UHF tag.

For implementations of CRYPTOGPS there are two per-tag secrets and the loss of either would entirely compromise the tag. The first is the secret CRYPTOGPS key. This could be attacked using techniques to solve the elliptic curve problem and we can turn to the established literature to establish an appropriate security level. Since any key is specific to a single tag, it is unlikely that we would need to protect against a widely distributed Internet-based effort. Thus the security level of 2^{80} seems reasonable and is attained using a CRYPTOGPS secret of length $|s| = 160$. Each tag also uses a PRNG to regenerate r_i. This requires a per-tag secret key and, again, an 80-bit key would be appropriate. This fits well with the goal of using PRESENT in this role, though since we aim to use PRESENT with 128-bit keys as the basis for the function F, we can either use a 128-bit key with PRESENT-128 as the PRNG or we can use a padded 80-bit key. There is no significant impact in performance for either choice.

Taking into account the security analysis given in Section 3.2, we propose to use a reader-provided challenge w of size 64 bits, and to fix the size of the coupons x_i and the derived challenges c_i to 64 bits. And, in turn, we can set $|y_i| = |r_i| = \rho = |s| + |c_i| + 80 = 304$.

Turning to wider considerations, the per-tag public key needs to be delivered to the reader in an authenticated way. There are a variety of architectural ways that this might be done. However, the conventional solution would be to sign the per-tag public key—using a system-side signature algorithm—and the tag can deliver its public key and associated signature to the reader. The reader holds the signing verification key needed to authenticate the per-tag public key. Unlike the per-tag public-private keys, the system-wide signing key would be a single point of failure for a widely-deployed system and a security level greater than 80-bit is likely to be preferred.

5.2 Area, time, and complexity

There are many factors to consider when implementing cryptography on an RFID tag. In this section we take the preferred parameter set outlined above and estimate the impact of implementing the non-transmissible signature variant of CRYPTOGPS. Our calculations are best-effort, but since they are based on a wealth of data from synthesized and fabricated versions of PRESENT and CRYPTOGPS we expect them to be reasonably accurate.

The on-tag requirements for the CRYPTOGPS computation are PRESENT and an integer multiplication. There are numerous implementations of PRESENT with 80-bit keys, some fabricated [38] but not so many with 128-bit keys. Instead we refer to [4] where the same technology is used to synthesize both variants and requiring 1570 GE for PRESENT-80 and 1886 GE for PRESENT-128. Both require the same time to complete an encryption operation.

When we turn to the computation of $y = r + sc$, the most useful source of information data is [39]. There, different strategies for implementing the computation of sc are described and compared. More usefully, the implementations outlined in [39] give area and time estimates for combining the multiplication operation with the regeneration of r when using PRESENT. This is an important issue since the combined operation can suffer from unexpected latencies unless optimisations to the two components are done in a coherent manner. Happily, the results in [39] cover the case of using a 160-bit secret s and a 64-bit challenge c and so there is no need to take any liberties in extrapolating from smaller parameters. The anticipated area and time requirements for the non-transmissible signature variant of CRYPTOGPS, denoted CRYPTOGPS-NTS, are given below for two different implementations strategies, denoted (A) and (B), which give different area/time trade-offs.

	CRYPTOGPS		CRYPTOGPS-NTS	
F	-		PRESENT-128	
PRNG	PRESENT-80		PRESENT-128	
	(A)	(B)	(A)	(B)
estimated area (GE)	3 424	3 300	3 740	3 616
estimated time (cycles)	389	713	421	745

In short, the area overhead in moving to CRYPTOGPS-NTS could be as little as 316 GE. However, during fabrication there are inevitable increases (typically of the order of 18-20%) to the area that are not reflected in synthesis results. Further, we have an additional complexity on the tag; namely the computation of $\text{ENC}_{x_i\|w}(0^n)$. While this won't add too much in terms of area, there will be an additional complexity to the implementation as the key for the PRESENT unit is swapped with the key k that is used to generate r. This will be reflected in some increase to the control logic and some additional time. The time for changing the key will not be significant; it consists merely of writing over the key state with a new value and this will depend on the internal operand size. It will likely remain a small fraction of the total processing time on the tag. For the increase in the control logic, we note from [39] that the control logic for the implementation of the computation $r + sc$ consumes around 6-10% of the total area. Even a doubling of the control logic, which is somewhat unlikely, would yield an additional overhead of 10%

to the complete implementation, which is within the margin of error that this kind of computation inevitably carries.

In terms of time for the on-tag computation, the computation of F will be an overhead, but depending on the working unit for the computation it is likely to be 32 or 64 cycles. This additional cost will be more than compensated for by the fact that the protocol is now challenge-response. The protocols used in RFID applications require that the tag responds to the reader. To illustrate, for a CCR scheme we would expect something like the following schema:

$$\begin{array}{rcl}
\text{Reader} & & \text{Tag} \\
\texttt{start} & \longrightarrow & \\
 & \longleftarrow & \texttt{send}\, x_i \\
\texttt{send challenge}\, c & \longrightarrow & \texttt{compute}\, y_i \\
 & \longleftarrow & \texttt{send}\, y_i
\end{array}$$

For the signature variant we would have

$$\begin{array}{rcl}
\text{Reader} & & \text{Tag} \\
\texttt{send challenge}\, w & \longrightarrow & \texttt{compute}\, c_i = \mathrm{F}(x_i, w) \\
 & & \texttt{compute}\, y_i \\
 & \longleftarrow & \texttt{send}\, c_i\, \texttt{and}\, y_i
\end{array}$$

The amount of operational data sent would be the same in both cases, namely 432 bits[5] for our preferred parameter sets. However, in a multi-tag (potentially multi-reader) environment a single challenge-response interchange is easier and more reliable to maintain. Further, each message sent between reader and tag has an operational overhead; there is header information and trailing information that carries the results of a CRC computation. Much depends on the specific formats of the commands and messages, but a saving of around 40 bits in total is likely. This may not sound like a lot, but each bit counts and suggests a reduction of around 10% in the total communication overhead.

To summarise, we estimate that the non-transmissible signature version of CRYPTOGPS can be implemented in around 4000 GE within around 800 cycles. Given the increased reliability and simplicity when using a challenge-response protocol, it seems likely that this variant of CRYPTOGPS will be of some interest in future prototyping. Of course, it should be noted that we have concentrated on performance issues such as area and processing time. In fact, the average and peak power consumption are also crucial and while existing work on PRESENT and CRYPTOGPS are very promising in this regard, this aspect of the scheme we have presented will be considered further in future work.

6 Conclusions

In this paper we have considered a signature variant of the CRYPTOGPS commitment-challenge-response (CCR) scheme. This variant has not been widely considered for UHF RFID tag deployment, despite featuring as an ISO standard, since the classical

[5] According to the parameters choice (see Section 5.1), $|x_i|+|c_i|+|y_i| = 64+64+304 = 432$.

conversion from CCR to signature scheme is too costly on the tag. However, we have shown that it is possible to re-use the block cipher that is already required to support CRYPTOGPS and to define a different conversion method. Fully supported by theoretical security arguments, the preferred parameter set for this variant of CRYPTOGPS appears to be well-suited for UHF RFID deployment.

Acknowledgements

This work has been partially supported by the European Commission under Contract ICT-2007-216676 ECRYPT II. We are moreover grateful to the anonymous referees for their interesting and useful comments.

References

1. M. Aigner, T. Burbridge, A. Ilic, D. Lyon, A. Soppera, and M. Lehtonen. RFID Tag Security, BRIDGE white paper. Available via `www.bridge-project.eu`.
2. J-P. Aumasson, L. Henzen, W. Meier and M. Naya-Plasencia. Quark: A Lightweight Hash. In S. Mangard and F.-X. Standaert, editors, *Cryptographic Hardware and Embedded Systems - CHES 2010*, volume 6225 of LNCS, pages 1-15. Springer, 2010.
3. M. Bellare and P. Rogaway. Random Oracles are Practical: A paradigm for designing efficient protocols. In ACM CCS'93, pages 62-73. ACM, 1993.
4. A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In P. Paillier and I. Verbauwhede, editors, *Proceedings of CHES '07*, volume 4727 of LNCS, pages 450-466. Springer, 2007.
5. A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. Robshaw, and Y. Seurin. Hash Functions and RFID Tags: Mind the Gap. In E. Oswald and P. Rohatgi, editors, Proceedings of CHES 2008, LNCS 5154, pages 283-299, Springer, 2008.
6. A. Bogdanov, M. Knezevic, G. Leander, D. Toz, K. Varici, and I. Verbauwhede. SPONGENT: A Lightweight Hash Function. In B. Preneel and T. Takagi, editors, Proceedings of CHES 2011, LNCS 6917, pages 312-325, Springer, 2011.
7. J. Cho. Linear Cryptanalysis of Reduced-round PRESENT. In J. Pieprrzyk, editor, *Proceedings of CT-RSA 10*, volume 5985 of LNCS, pages 302-317, Springer, 2010.
8. B. Collard and F.-X. Standaert. A Statistical Saturation Attack Against the Block Cipher PRESENT. In M. Fischlin, editor, *Proceedings of CT-RSA 09*, volume 5473 of LNCS, pages 195-210, Springer, 2009.
9. J.S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgärd Revisited: How to Construct a Hash Function. In V. Shoup, editor, *Proceedings of Crypto '05*, volume 3621 of LNCS, pages 430-448. Springer, 2005.
10. EPCglobal. EPC Radio-Frequency Identity Protocols, Class-1 Generation-2 UHF RFID, Protocol for Communications at 860-960 MHz, version 1.2.0. October 23, 2008. Available via `www.epcglobalinc.org`.
11. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In A. Odlyzko, editor, *Advances in Cryptology - Crypto 86*, LNCS 263, 186-194, Springer-Verlag, 1987.
12. M. Girault. Self-certified Public Keys. In D.W. Davies, editor, *Proceedings of Eurocrypt '91*, volume 547 of LNCS, pages 490-497, Springer-Verlag, 1991.

13. M. Girault. Low-Size Coupons for Low-Cost IC Cards. In J. Domingo-Ferrer, D. Chan, and A. Watson, editors, *Proceedings of Smart Card Research and Advanced Applications*, pages 39-50, Kluwer Academic Press, 2001.
14. M. Girault, L. Juniot, and M. Robshaw. The Feasibility of On-the-Tag Public Key Cryptography. *RFIDsec 2007*, workshop record. Available via `rfidsec07.etsit.uma.es/slides/papers/paper-32.pdf`.
15. M. Girault and D. Lefranc. Public Key Authentication with One (Online) Single Addition. In M. Joye and J.-J. Quisquater, editors, *Proceedings of CHES '04*, volume 3156 of LNCS, pages 967-984, Springer-Verlag, 2004.
16. M. Girault, G. Poupard, and J. Stern. On the Fly Authentication and Signature Schemes Based on Groups of Unknown Order. Journal of Cryptology, vol. 19, pages 463-487, Springer, 2006.
17. M. Girault and J. Stern. On the Length of Cryptographic Hash-Values Used in Identification Schemes. In Y. Desmedt, editor, *Proceedings of Crypto '94*, volume 893 of LNCS, pages 202-215, Springer, 1994.
18. J. Guo, T. Peyrin, and A. Poschmann. The PHOTON Family of Lightweight Hash Functions. In P. Rogaway, editor, *Crypto 2011*, volume 6841 of LNCS, pages 222-239. Springer, 2011.
19. G. Hofferek and J. Wolkerstorfer. Coupon recalculation for the GPS Authentication Scheme. In G. Grimaud and F.-X. Standaert, editors, *Proceedings of Cardis 2008*, volume 5189 of LNCS, pages 162-175, Springer, 2088.
20. M. Hutter and C. Nagl. Coupon Recalculation for the Schnorr and GPS Identification Scheme: A Performance Evaluation. In proceedings of RFIDSec 2009. Available via `www.cosic.esat.kuleuven.be/rfidsec09/`.
21. ISO/IEC 9798: Information Technology – Security Techniques – Entity Authentication – Part 5: Mechanisms using Zero-Knowledge Techniques. Available via `www.iso.org`.
22. ISO/IEC 14888-2: Information Technology – Security Techniques – Digital Signatures – Part 2: Factoring Based Techniques.
23. ISO/IEC 29192-4: Information Technology – Security Techniques – Lightweight Cryptography – Part 2: Block ciphers.
24. ISO/IEC 29192-4: Information Technology – Security Techniques – Lightweight Cryptography – Part 4: Public key techniques.
25. J. Jenkins, P. Mills, R. Maidment, and M. Profit. Pharma Traceability Business Case Report. BRIDGE white paper, May 2007. Available via `www.bridge-project.eu`.
26. G. Leander. On Linear Hulls, Statistical saturation attacks, PRESENT and a cryptanalysis of PUFFIN. In K. Paterson, editor, *Proceedings of Eurocrypt 2011*, to appear, Springer, 2011.
27. M. Lehtonen, J. Al-Kassab, F. Michahelles, and O. Kasten. Anti-counterfeiting Business Case Report. BRIDGE white paper, December 2007. Available via `www.bridge-project.eu`.
28. M. McLoone and M.J.B. Robshaw. Public Key Cryptography and RFID Tags. In M. Abe, editor, *Proceedings of CT-RSA '07*, volume 4377 of LNCS, pages 372-384, Springer, 2007.
29. M. McLoone and M.J.B. Robshaw. New Architectures for Low-Cost Public Key Cryptography on RFID Tags. In *Proceedings of SecureComm '05*, pages 1827-1830. IEEE Computer Society Press, 2007.
30. M. McLoone and M.J.B. Robshaw. Low-cost Digital Signature Architecture Suitable for Radio-Frequency Identification Tags. In *IET Computers and Digital Techniques*, Vol. 4, Issue 1, pages 14-26, 2010.
31. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, Florida, USA, first edition, 1996.
32. J. Nakahara, P. Sepehrdad, B. Zhang, and M. Wang. Linear (hull) and Algebraic Cryptanalysis of the Block Cipher PRESENT. In A. Otsuka, editor, *Proceedings of CANS 09*, volume 5888 of LNCS, pages 58-75, Springer, 2009.

33. NESSIE consortium. Final Report of European Project IST-1999-12324: New European Schemes for Signatures, Integrity, and Encryption (NESSIE), April 2004. Available via `https://www.cosic.esat.kuleuven.be/nessie/`.
34. K. Ohkuma. Weak keys of reduced-round PRESENT for linear cryptanalysis. In M. Jacobson, V. Rijmen, and R. Safavi-Naini, editors, *Proceedings of SAC 09*, volume 5867 of LNCS, pages 249-265, Springer, 2009.
35. O. Özen, K. Varici, C. Tezcan, and C. Kocair. Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced-round PRESENT and HIGHT. In C. Boyd and J. Nieto, editors, *Proceedings of ACISP 09*, volume 5594 of LNCS, pages 90-107, Springer, 2009.
36. D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In U. Maurer, editor, *Proceedings of Eurocrypt '96*, volume 1070 of LNCS, pages 387-398. Springer, 1996.
37. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology, vol. 19, pages 361-396, Springer 2000.
38. A. Poschmann, M.J.B. Robshaw, F. Vater, and C. Paar. Lightweight Cryptography and RFID: Tackling the Hidden Overheads. In D. Lee and S. Hong, editors, *Proceedings of ICISC 2009*, volume 5984 of LNCS, pages 129-145, Springer, 2010.
39. A. Poschmann and M.J.B. Robshaw. On Area, Time, and the Right Trade-Off. In Y. Mu and W. Susilo, editors, Proceedings of ACISP 2012, LNCS 7372, pages 404-418, Springer, 2012.
40. G. Poupard and J. Stern. Security Analysis of a Practical "on the fly" Authentication and Signature Generation. In K. Nyberg, editor, *Proceedings of Eurocrypt '98*, volume 1403 of LNCS, pages 422-436. Springer-Verlag, 1998.
41. M.J.B. Robshaw. The eSTREAM Project. In M. Robshaw and O. Billet, editors, *New Stream Cipher Designs: The eSTREAM Finalists*, pages 1-6, Springer, 2008.
42. C. Shannon. Communication theory of secrecy systems. Bell System Technical Journal, Vol. 28, pages 656-715, 1949.
43. M. Wang. Differential Cryptanalysis of Reduced-round PRESENT. In S. Vaudenay, editor, *Proceedings of Africacrypt 08*, volume 5023 of LNCS, pages 40-49, Springer, 2008.
44. M. Z'aba, H. Raddum, M. Henricksen, and E. Dawson. Bit-pattern based Integral Attack. In K. Nyberg, editor, *Proceedings of FSE 08*, volume 5086 of LNCS, pages 363-381, Springer, 2008.
45. J. Black, P. Rogaway, and T. Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. In M. Yung, editor, *Proceedings of Crypto '02*, volume 2442 of LNCS, pages 320-335, Springer, 2002.
46. B. Preneel, R. Govaerts, and J. Vandewalle. Hash functions based on block ciphers: a synthetic approach. In D. R. Stinson, editor, *Proceeding of Crypto 1993*, volume 773, pages 368-378, Springer, 1993.
47. M. Braun, E. Hess, and B. Meyer. Using Elliptic Curves on RFID Tags. In Dr. J. M. Jun, editor, IJCSNS International Journal of Computer Science and Network Security, vol. 8, n° 2, February 2008.
48. L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. An Elliptic Curve Processor Suitable For RFID-Tags. In 1st Benelux Workshop on Information and System Security (WISSec 2006), 14 pages, Antwerpen, Belgium, November 8-9, 2006.

Appendix: On the Use of Coupons

As noted in the main text, CRYPTOGPS is ideally suited for use with coupons. A precomputed quantity that is used once and then discarded, coupons can be well-suited to many RFID applications. Often we expect tags to be read 10 to 20 times and then discarded or recommissioned. With a coupon of 64 bits, see Section 3.2, storing 10 or even 20 coupons does not pose a significant incremental cost for many applications.

However, the use of coupons is not to everyone's taste and certainly they are not suitable for all use-cases. Indeed some commentators are concerned that coupons could be consumed in a denial-of-service attack, *i.e.* by an attacker that maliciously exhausts coupons on a target RFID tag. This is true. But the benefit of such a time-consuming attack, that needs to be repeated on a tag-by-tag basis, is rarely if ever articulated. Nevertheless, in response to this concern there has been some work regarding on-the-tag coupon regeneration [19,20] though this does not seem to be realistic in deployment. Other more practical approaches have considered ways of reloading coupons and on mechanisms to deliver coupons directly to the reader so that they don't need to be carried on the tag.

All in all, the suitability of coupons depends fundamentally on the use-case and the kind of adversary we are likely to encounter.