Open access • Book Chapter • DOI:10.1007/11605805_8

RFID-Tags for anti-counterfeiting — Source link

Pim Tuyls, Lejla Batina

Institutions: Philips, Katholieke Universiteit Leuven

Published on: 13 Feb 2006 - The Cryptographers' Track at the RSA Conference

Topics: Physical unclonable function, Authentication and Radio-frequency identification

RFID-Tags for Anti-Counterfeiting *

Pim Tuyls[1] and Lejla Batina[2]

[1] Philips Research Laboratories,
Prof. Holstlaan 4, 5656 AA, Eindhoven, The Netherlands
[2] Katholieke Universiteit Leuven, ESAT/COSIC,
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
{Pim.Tuyls,Lejla.Batina}@esat.kuleuven.ac.be
Pim.Tuyls@philips.com

Abstract. RFID-tags are becoming very popular tools for identification of products. As they have a small microchip on board, they offer functionality that can be used for security purposes. This chip functionality makes it possible to verify the authenticity of a product and hence to detect and prevent counterfeiting. In order to be successful for these security purposes too, RFID-tags have to be resistant against many attacks, in particular against cloning of the tag. In this paper, we investigate how an RFID-tag can be made unclonable by linking it inseparably to a Physical Unclonable Function (PUF). We present the security protocols that are needed for the detection of the authenticity of a product when it is equipped with such a system. We focus on off-line authentication because it is very attractive from a practical point of view. We show that a PUF based solution for RFID-tags is feasible in the off-line case.

Key Words: RFID, counterfeiting, authentication, ECC, Physical Unclonable Function (PUF)

1 Introduction

RFID-tags are low-cost pervasive devices that target to provide identification of goods. They consist of an antenna connected to a microchip. Because of the presence of this microchip, they can be considered as a next generation of bar codes with added functionality. In supply chain management they allow for tracking of a product in several stages and locations. Several applications are being developed that can process the data obtained from the tags for their own purposes, such as automated inventory management, automated quality control, access control, payment systems and general security applications. Clearly, one of the main success factors for a large deployment of RFID-tag based systems is the price of the tags. Currently the prices range from a few cents up to 1$. Very cheap

* Lejla Batina is funded by a research grant of the Katholieke Universiteit Leuven, Belgium. This work was supported by Concerted Research Actions GOA-Mefisto 2000/06 and GOA-Ambiorix 2005/11 of the Flemish Government and by the FWO projects (G.0141.03) and (G.0450.04).

tags do not carry a battery but obtain their power from the electromagnetic field generated by the reader querying the tag.

An emerging application that goes beyond identification, is the use of RFID-tags for anti-counterfeiting purposes [1]. By locating an RFID-tag with specific product and reference information on a product, one aims to verify the authenticity of the product. Loosely speaking the verification is performed as follows. When a product passes a reader, the reader checks whether the necessary and authentic product and reference information is present on the tag. For this purpose it runs a protocol with the tag. If the necessary information is there and verified to be authentic, the product is declared to be genuine and otherwise not. However, by capturing the necessary authentication information (obtained *e.g.* by eavesdropping the protocol between the tag and the reader), and by storing it in a new chip, the attacker has effectively made a clone of the original tag that cannot be distinguished from an original tag by a reader. In order to make this cloning of the tag infeasible, it should not be possible to derive the tag secrets by active or passive attacks. Recently a lightweight version of such a protocol was developed in [1].

We stress however that it is rather easy to physically clone a tag. This means that an attacker can capture the RFID-tag, investigate it, read out its memory (with reasonable effort) and in particular its security related data (identification number, reference information, keys, etc). Then she produces a new tag with exactly the same data in its memory. When this tag is embedded into a product, it is impossible for a reader to distinguish an authentic product from a fake one. In order to protect an RFID-tag against this type of cloning attack, one can of course attempt to prevent read out its memory by using several protective measures [23, 18]. However these measures will increase the price of the tag so much that it will become unacceptably high for its main application. In order to thwart the physical cloning attacks we propose to use Physical Unclonable structures (so-called PUFs) for storing secret key material in the tag. PUFs have been proposed as a cost-effective mean to produce unclonable tokens for identification [20, 21]. They are realized as a physical system such that the function is easy to evaluate but hard to clone.

The contributions of this paper are:

1. We identify the technological components of anti-counterfeiting technology and give a general protocol for verifying the authenticity of an item.
2. We propose a solution for anti-counterfeiting based on RFID-tags and PUFs [25, 22, 19, 20]. Our solution withstands physical cloning attacks as well as active and passive attacks on the verification protocols. In particular, we present a solution based on PUFs that are inseparably bound to an IC.
3. We present protocols for the off-line situation (as far as we are aware this is the first time that the off-line case has been considered). Our construction for the off-line case is designed in such a way that it inherits its security from the underlying cryptographic algorithms (signature and secure identification scheme) used.

4. We show that the construction that we propose is feasible on a constrained device such as an RFID-tag. In order to minimize the area constraints of a tag, we sacrifice slightly the efficiency of the involved cryptographic algorithms. The obtained performance is still sufficient for our application.

The paper is organized as follows. In Sect. 2 we identify the required technological components for anti-counterfeiting technology. Additionally, a general protocol for the verification of the authenticity of a product is given. Section 3 mentions related work. An overview of PUFs and associated key-extraction algorithms is given in Sect. 4. In Sect. 5 we introduce unclonable RFID-tags with an Integrated PUF on board. Furthermore, we present verification protocols for off-line authentication. Finally, in Sect. 6 we investigate the efficiency of the off-line verification protocol in detail.

2 Model

In order to protect a product against cloning (counterfeiting) a detection mark is embedded into the product or its packaging. This detection mark consists of a physical and a digital part. The mark is put there by a legitimate authority. The attacker (counterfeiter) has access to all components of this detection mark; *i.e.* she can read it, remove it from the product and investigate it. Based on the information that she obtained from investigating the legal detection mark, she produces a fake detection mark. The goal of the attacker is to produce a fake detection mark that can only with small probability be distinguished from an authentic one.

2.1 Components of Anti-Counterfeiting Technology

In order to protect a product against counterfeiting, technological means are needed to verify whether the product is authentic or not. In order to make an item unclonable, the following two components are needed.

1. *Physical protection.* This is obtained by using unclonable physical structures embedded in the package (removal of the structure leads to its destruction). One or more unique *fingerprints* derived from the physical structure will be printed on the product for the verification of the authenticity of the product.
2. *Cryptographic protection* serving two goals. Firstly, cryptography provides techniques (digital signatures) to detect and prevent tampering with data (fingerprints) derived from a physical object. Secondly, it provides secure identification protocols to identify a product. Those protocols do not leak any necessary identification information to an eavesdropper attacking (actively or passively) the communication channel.

Good candidates for unclonable physical structures, that can be used for physical protection purposes, are so-called Physical Unclonable Functions (PUFs) [25].

2.2 A General Anti-Counterfeiting Protocol

We give intuition for protocols that can be used to check the authenticity of a product based on embedding a PUF in the product in combination with the use of cryptographic techniques.

First there is an enrollment phase, which is performed by some trusted authority. During this phase the following steps are performed.

1. Several fingerprints are derived from the PUF by challenging it with multiple challenges and recording the responses. These responses are then turned into binary fingerprints (and some auxiliary data are derived for use during the verification phase).
2. These challenges, fingerprints and auxiliary data are then signed with the secret key sk of the issuer of the product (the issuer is assumed to be trustworthy).
3. The signatures, the challenges (corresponding to the fingerprints) and maybe some auxiliary data (needed to perform processing during the authentication phase) are also printed on the product (and/or stored in a database).

During the verification phase, the authenticity is checked by running the following protocol.

1. The verification device reads the challenges and auxiliary data.
2. The verification device challenges the physical structure with one of the challenges printed on the product. After having measured the responses, it derives the fingerprint from the response based on the auxiliary data.
3. Then, using the fingerprint derived in step 2., the verification device checks the signature to verify that the fingerprint, challenges and auxiliary data were printed on the product by a legitimate authority. If the signature is not correct, the product is not authentic.

We briefly analyze the security of this protocol. An attacker who wants to counterfeit the product has to embed a fake physical structure on the product that produces correct fingerprints to the challenges (with correct signatures). Under the assumption that the physical structure is unclonable, she cannot produce a clone of the originally embedded physical structure. More precisely, we assume that given some challenges $c_1, \ldots, c_n$ and corresponding fingerprints $s_1, \ldots, s_n$ she cannot produce a (fake) physical structure that produces the same fingerprints $s_1, \ldots, s_n$ given the original challenges $c_1, \ldots, c_n$. On the other hand she can produce another structure and create challenges, auxiliary data and fingerprints $s'_1, \ldots, s'_n$ according to the procedures used during enrollment. However, since she does not know the secret key sk and the responses of her fake structure will be different with very high probability, she will not be able to put the correct signatures on these data. The verification device will detect that the signatures are not correct and reject this as a fake product.

We note that the number of fingerprints that can be verified during a verification session is very limited by time and space constraints. Furthermore, the attacker can easily capture the required fingerprints (by measuring the responses

according to the challenges printed on the product). Therefore the production of a clone only requires the fabrication of a physical structure (PUF) producing the same fingerprints for a limited number of challenges.

2.3 RFID Systems

The PUF based solution for preventing counterfeiting of goods that was presented above can be improved with active components, that are inseparably linked with a PUF. An example consists of an RFID-tag equipped with a microchip that is inseparably bound to a PUF. The precise construction is explained in Sect. 4. Because of the presence of a microchip a secure identification protocol can be run without revealing any information on the fingerprint of the PUF. Additionally, by inseparably linking the chip and the PUF, it becomes possible to prevent leakage of the PUF measurement to the outside world.

Typical RFID systems consist of the following two components: the *RFID-tag* and a *reader*. The reader will perform the verification to detect whether a tag is authentic or not. The RFID-tag consists of an antenna connected to a microchip that can store and read data and has possibly some dedicated hardware to perform a small amount of computations. Typically, the power for performing operations is obtained from the RF-field (by inductive coupling). A reader can read and write data from/on a tag. The reader is often linked with some system that can perform computations on the data that it receives from tags.

In order to use RFID-tags for anti-counterfeiting purposes, we proceed as follows. An RFID-tag containing reference information is embedded in a product. The (identification) data stored in the memory of the tag is signed with the secret key sk of the legitimate issuer. The tag communicates with a reader for verification purposes over a public channel. The ROM memory of the tag is accessible to the attacker. The reader has a certified public key pk corresponding the issuer's secret key for verification of the digital signatures.

3 Related Work

The two most related papers to ours are [1] and [12]. Both deal with the cloning problem of RFID-tags and hence with the problem of using RFID-tags for anti-counterfeiting purposes. The focus of these papers is on efficient protocols for authenticating these tags. In these papers, one focuses on authentication of RFID-tags in the on-line situation; *i.e.* the reader shares a secret with the RFID-tag that is being authenticated. Clearly, when RFID-tags will become widely used, this is not a reasonable assumption.

4 Physical Unclonable Functions

For the sake of clarity we start with a definition of a PUF [2].

Definition 1 *A Physical Unclonable Function is a function that maps challenges to responses and that is embodied in a physical object. It satisfies the following properties:*

1. *Easy to evaluate: the physical object can be evaluated in a short amount of time.*
2. *Hard to characterize: from a number of measurements performed in polynomial time, an attacker who no longer has the device and who only has a limited (polynomial) amount of resources can only obtain a negligible amount of knowledge about the response to a challenge that is chosen uniformly at random.*

More formally the PUF model is as follows. We denote the PUF response to a challenge C during the enrollment phase by $X \in \mathbb{R}^n$ and during the verification phase by $Y \in \mathbb{R}^n$ (the pair (C, X) is called a Challenge-Response pair or CRP). The PUF response according to a fake PUF is denoted by Z. The responses X, Y, Z are modeled as random variables with probability distribution $\mathbb{P}_{X,Y,Z}$.

Definition 2 *Let $\delta, \epsilon_a, \epsilon_e \geq 0$. A joint distribution $\mathbb{P}_{X,Y,Z}$ on $(\mathbb{R}^n)^3$ is called $(\delta, \epsilon_a, \epsilon_e)$-reliable if it satisfies i) $Prob(d(Y, X) > \delta) \leq \epsilon_a$ and ii) $Prob(d(Z, X) \leq \delta) \leq \epsilon_e$; here the probabilities are over the joint distribution $\mathbb{P}_{X,Y,Z}$.*

This definition implies that if the enrollment and authentication measurements (according to the same challenge C) are performed on the same PUF, then these responses are with high probability very close to each other. When on the other hand the measurements are performed on different PUFs (modeling the fact that the PUF used during authentication might be fake), the responses are with high probability far apart.

We propose to equip the microchip on an RFID-tag with a PUF that is inseparably linked to the chip. More precisely we define this as follows.

Definition 3 *An Integrated Physical Unclonable Function (I-PUF) is a PUF that additionally satisfies the following properties.*

1. *The I-PUF is inseparably bound to a chip which means that any attempt to remove the PUF from the chip leads to the destruction of the PUF and the chip.*
2. *It is impossible to tamper with the communication (measurement data) between the chip and the PUF.*
3. *The output of the PUF is inaccessible to an attacker.*

In the remainder of the paper we will only use I-PUFs, while we will often use just the abbreviation PUF.

The two best known examples of such I-PUFs are silicon PUFs [9] and coating PUFs [19]. For coating PUFs it is expected that the additional measurement circuit requires less than 1000 gates.

4.1 Key Extraction

In this paper, the term key extraction always refers to key extraction from noisy data. Generally speaking a key extraction algorithm is built on a Secret Extraction Code [24][3]. For the sake of simplicity we describe the algorithm in terms of a *shielding function* [14] or (G, W)-pair [26], which generates a special set of Secret Extraction Codes, while having all the necessary properties.

A function $G(.,.) : \mathbb{R}^n \times \mathcal{W} \to \{0,1\}^k$ is called δ-contracting if for all X there exists *helper data* $W \in \mathcal{W}$ such that for all X' that lie within a sphere of radius δ of X ($||X' - X|| \leq \delta$) $G(X', W) = G(X, W)$ ($\mathcal{W}$ denotes the space of helper data. At this point it has to be considered as some abstract space.). We use δ-contracting functions to extract keys $S = G(X, W)$ from noisy data X using *helper data* W. A function $G(.,.)$ is called ϵ-revealing if the helper data W leaks less than ϵ bits on S (in the information theoretic sense), *i.e.* $\mathbf{I}(W; S) \leq \epsilon$. An (ϵ, δ)-shielding function $G : \mathbb{R}^n \times \mathcal{W} \to \{0,1\}^k$ is a function that is δ-contracting and ϵ-revealing. It is used to extract a secret of length k from the PUF response as follows.

- **Enrollment Phase:** The PUF is subjected to a challenge C and the response X is measured. Then a random key S is chosen from $\{0,1\}^k$ and helper data W is computed by solving $G(X, W) = S$ for W. The quadruplet $(\mathrm{ID}_{\mathrm{PUF}}, C, W, S)$ is then stored in a CRP database.
- **Verification Phase:** When the PUF is inserted into the reader the PUF's identity is sent to the verifier. The verifier chooses a random challenge C from his database and sends it to the PUF together with the corresponding helper data W. Then the PUF is subjected to the challenge C and its response X' is measured. A key S' is then computed as $S' = G(X', W)$.

Notice that if $G(.,.)$ is δ-robust and if $\mathbb{P}_{X,Y,Z}$ is $(\delta, \epsilon_a, \epsilon_e)$-reliable, then we obtain $\mathrm{Prob}(G(Y, W) = S) \geq 1 - \epsilon_a$ and $\mathrm{Prob}(G(Z, W) = \perp) \geq 1 - \epsilon_e$, which expresses that FRR (False Rejection Rate) and FAR (False Acceptance Rate) are at most ϵ_a and ϵ_e respectively. In the case of a passive attacker, the extracted key S can then be used securely since $\mathbf{I}(W; S) \leq \epsilon$. Note that by adding a privacy amplification this can be guaranteed (if the Rényi entropy is sufficiently large). Also note that this procedure can be used to set up a shared secret key between an I-PUF and a verifier (reader).

Since the PUF responses are often analog data[4], the helper data typically consists of three parts. The first part W_1 allows to quantise the signal into a binary representation while the second part W_2 implements the error correction and the random key choice on the binary data. The third part is used for privacy amplification. For a detailed example, we refer the reader to [22] for the case of optical PUFs.

[3] This construction can be applied to discrete and continuous data. An equivalent construction for the discrete case, called Fuzzy Extractors, was developed by Dodis *et al.* in [8].

[4] In the case of an optical PUF the PUF response is a speckle pattern which can be seen as an analog picture. In the case of a coating PUF the responses are given by capacitance values which are analog signals.

4.2 Example

We present a brief example of key extraction from noisy (binary) data. It shows that the required processing at the side of the RFID-tag is low. Assume for the sake of simplicity that the responses X are uniformly random binary strings of length k, *i.e.* $X \in \{0,1\}^k$. Furthermore, we assume that the authentication measurement performed during the verification phase can be modeled as a noisy observation over a binary symmetric channel with cross-over probability p. Let $\mathcal{C}$ be an error correcting code, with l codewords. Then, for a key $s \in_R \{0, \ldots, l-1\}$ the helper data $w(x,s) = x \oplus c_s$ is generated during the enrollment phase (where $c_s \in \mathcal{C}$). During the verification phase, the tag measures y and computes $G(y, w(x;s)) = \mathsf{Dec}(y \oplus w(x;s))$ (Dec denotes the decoding algorithm of the error-correcting code $\mathcal{C}$). Clearly, if y corresponds to the same challenge (and the same PUF), s is obtained after decoding while otherwise a random codeword is obtained or a decoding error. Hence, the tag has to perform an XOR operation and a decoding operation. On a tag with some S-RAM (Static RAM) available (which most tags have), the decoding costs less than 1000 gates[5].

5 Unclonable RFID-Tags

5.1 Set-up

In order to make unclonable RFID-tags, we introduce RFID-tags whose microchips are equipped with an I-PUF.

In our construction, the PUF is used as a secure memory for storing secret keys. The secret key s which is usually stored in (protected) ROM or EEPROM is derived from the PUF, when needed. In order to enable the generation of the secret key s during authentication, helper data w is stored in (publicly accessible) ROM (EEPROM). The key s is derived from the response X of the PUF by means of a key extraction algorithm (Fuzzy Extractor and the helper data w are used here). It was mentioned in Sect. 4.1 that the public helper data w reveals only a negligible amount of information on the key s. Given our assumption on I-PUFs in Def. 3, it follows that the key s is securely stored in the PUF.

5.2 Off-Line Authentication

We introduce our PUF-Certificate-Identity-based Identification scheme (PUF-Cert-IBI) by following the definition of Certificate-based IBI in [4]. Let $\mathcal{SI} = (K_g, P, V)$ denote a standard identification scheme (SI-scheme) where K_g denotes the key generation algorithm, and P, V denote the interactive protocols run by the prover and verifier respectively. Let $\mathcal{SS} = (\mathrm{SK}_g, \mathrm{Sign}, V_f)$ be a standard signature scheme (SS-scheme) [7] with SK_g denoting the key generation algorithm, Sign denoting the signing algorithm and V_f the verification algorithm

[5] In the case of coating PUFs the codewords are relatively short (200 bits) and the information rate is high. In that case BCH codes are efficient in use.

run by a verifier. We assign to each tag an identity I (this might be the serial number or EPC-code of the tag or the serial number of the product in which it has been embedded). To the PUF, the $\mathcal{SI}$, the $\mathcal{SS}$ scheme and the identity I an Identity-Based Identification scheme $(\mathrm{MK}_g, \mathrm{UK}_g, \hat{P}, \hat{V})$ is associated as follows.

During **enrollment** the issuer uses SK_g as the master-key generation algorithm MK_g. This means that the master key msk is used for generating signatures and the corresponding public key mpk for verification of the signatures. The user key generation algorithm UK_g consists of the following steps. For each RFID-tag, having identity I, the issuer then creates a public-secret key pair (pk, sk) using the algorithm K_g on input 1^k. The couple (pk, sk) is the public-secret key pair for the SI-scheme. The issuer runs the following protocol with the tag.

- It requests the tag to challenge its PUF with a challenge c and to measure the response $x(c)$.
- The tag sends $x(c)$ to the issuer.
- Based on the knowledge of $x(c)$ and sk, the issuer determines the helper data w such that $sk = G(x, w)$.
- The helper data w are written into the ROM (EEPROM) memory of the tag.

Finally, the issuer creates the following certificate that is also stored in the ROM of the tag $\mathrm{Cert} \leftarrow (pk, \mathrm{Sign}(msk, pk||I))$. The usk is then put to $usk \leftarrow (\mathrm{PUF}, \mathrm{Cert})$.

During **authentication**, the tag (in the role of the prover) runs the following steps with a verifier.

- The tag runs the protocol $\hat{P}$ which consists of the following steps.
  - It challenges the PUF with c, measures the response $y(c)$ and computes $sk \leftarrow G(y(c), w)$.
  - Initialisation of the prover protocol P of the $\mathcal{SI}$ scheme with sk.
  - It includes the certificate Cert in the first step of the algorithm P.
- The verifier uses (mpk, I) as input for the verification algorithm $\hat{V}$.
- When the verifier receives Cert from the tag, it first verifies Cert by running $V_f(mpk, pk||I, \mathrm{Sign}(msk, pk||I))$.
- If the certificate Cert is invalid the protocol is aborted.
- If Cert is valid, the verifier initializes V with pk and runs it.
- If V accepts, then the verifier accepts.

The security of our PUF-Certificate-Identity-based identification scheme follows from the following theorem. This theorem is very similar to theorem 4.2 in [4]. The proof of the theorem presented there, can be applied here with minor modifications and is therefore omitted.

Theorem 1 *Let $\mathcal{SI}$ be an SI-scheme and SS a uf-cma* [6] *secure SS-scheme. Let PUF-Cert-IBI be the corresponding PUF-Certificate-Identity based Identification*

[6] *uf-cma*: existential unforgeability under chosen message attack.

scheme presented above. If the scheme $\mathcal{SI}$ is impersonation-atk secure then PUF-Cert-IBI is impersonation-atk secure for atk $\in \{pa, aa, ca\}$ (pa: passive attack, aa: active attack, ca: concurrent attack).

It follows from this theorem, that by choosing an appropriate SI-scheme (withstanding a *pa*, *aa* or *ca*) the PUF-Cert-IBI inherits the same property. If only resistance against passive attacks is needed, the Schnorr Identification scheme can be used. It is known that this scheme is secure against passive attacks under the discrete logarithm assumption. It is also secure against active attacks under the one-more-discrete-logarithm assumption. An alternative is to use Okamoto's identification scheme [16], which is secure against passive, active and concurrent attacks under the discrete logarithm assumption.

5.3 Storage Requirements:

In order to minimize the size of the ROM memory of the tag as small as possible, we propose to use Elliptic Curve Discrete Log based secure identification schemes. This makes an implementation on an Elliptic Curve (EC) possible. For the signature algorithm $\mathcal{SS}$ we take then the ECDSA approach. This makes the size of the signatures no larger than 326 bits. The identification protocol investigated in detail is the Schnorr identification protocol. For the sake of completeness the ECC version of the protocol is given in Appendix. The total storage requirement for the public information (sP, Cert) is in total at most 500 bits.

6 Implementation

In this section, we discuss implementation issues, *i.e.* efficiency and size of the hardware if the off-line RFID identification protocol is implemented on an RFID-tag. As an example we take the Schnorr identification protocol, which allows a user to prove knowledge of x given the public information g^x in a group where the discrete log problem is difficult. For the sake of efficiency, we investigate the efficiency of this protocol on an elliptic curve over $\text{GF}(2^{163})$.

6.1 Elliptic Curves over $\text{GF}(2^n)$

Elliptic Curve Cryptography (ECC) relies on a group structure induced on an elliptic curve. The set of points on an elliptic curve (with one special point added, the so-called point at infinity $\mathcal{O}$) together with point addition as a binary operation has the structure of an abelian group. Here we consider a finite field of characteristic 2, *i.e.* $\text{GF}(2^n)$. A non-supersingular elliptic curve E over $\text{GF}(2^n)$ is defined as the set of solutions $(x, y) \in \text{GF}(2^n) \times \text{GF}(2^n)$ of the equation:

$$y^2 + xy = x^3 + ax^2 + b\,, \tag{1}$$

where $a, b \in \text{GF}(2^n)$, $b \neq 0$, together with $\mathcal{O}$.

The point or scalar multiplication is the basic operation for cryptographic protocols based on ECDLP; it is easily performed via repeated group operations. One can visualize these operations in a hierarchical structure. Point multiplication is at the top level. At the next (lower) level are the point operations, which are closely related to the coordinates used to represent the points. The lowest level consists of finite field operations such as addition, subtraction, multiplication and inversion required to perform the group operations.

The easiest way to calculate the point or scalar multiplication is by means of the basic double-and-add algorithm [16].

The point addition in affine coordinates is performed according to the formulae in [6]. In either case, the computation requires one field inversion (I), two field multiplications (M) and one squaring (S), or $1I + 2M + 1S$. As we are interested in hardware implementations, we count squarings and multiplications together as they are both executed on the same multiplier.

The inversion operation is very costly in hardware and can be avoided by choosing one of many options for projective coordinates. However, the number of multiplications is increased in this case, which makes the choice of a multiplier even more crucial for an efficient implementation. To summarize, we consider squaring as a special case of multiplication and inversion is ignored. The addition of two field elements requires the modulo 2 addition of the coefficients of the elements. In hardware, a bit-parallel adder requires n XOR gates and the sum can be computed in one clock cycle.

Another option for scalar or point multiplication is to use the so-called "Montgomery ladder"[11]. According to López and Dahab [15], the Montgomery representation requires less memory and offers a better protection against side-channel attacks. These both facts are very useful in this case as memory *i.e.* registers are very "expensive" in hardware implementations. Also, side-channel attacks are an issue on RFID tags and also some cheap protection *i.e.* by means of balanced implementations is desirable.

The idea of Montgomery dealt with speeding up the calculation of only the x-coordinate of the result. More precisely, to add two points their difference is used as an input parameter while the y-coordinate is not used in the algorithm. This fact is justified by cryptographic applications that rarely use the y-coordinate. The algorithm for scalar multiplication is a variant of the binary method and was considered by López and Dahab [15]. They have also introduced an option for recovering the y-coordinate.

We introduce the following notation: $P_4 = (x_4, y_4) = P_2 - P_1$, $P_5 = (x_5, y_5) = 2P_1$ and $P_3 = P_1 + P_2$. The point P_4 is included because the method for point multiplication, as introduced by Montgomery, is defined by the fact that to add two points their difference should be known (while y-coordinate is not needed).

For point operations (addition and doubling) we consider the formulae of López and Dahab in GF(2^n). The operation count is $A : D = 5M : 6M$ (2). Here, A and D are the point operations and M is a field multiplication. We remind the reader that field addition in hardware for GF(2^n) is just a simple bit-wise XOR operation and therefore is not taken into account. We use the

formulae for point operations in the case of simple projective coordinates *i.e.* $x_i = (X_i/Z_i), i = 1, 2$. The results of point doubling and point addition, *i.e.* $X_5 = X(P_5)$ and $X_3 = X(P_3) = X(P_1 + P_2)$ respectively, are calculated as:

$$\begin{aligned} X_5 &= {X_1}^4 + b \cdot {Z_1}^4 \\ Z_5 &= {X_1}^2 \cdot {Z_1}^2 . \end{aligned} \tag{2}$$

$$\begin{aligned} X_3 &= x_4 \cdot Z_3 + (X_1 \cdot Z_2) \cdot (Z_1 \cdot X_2), \\ Z_3 &= (X_1 \cdot Z_2 + X_2 \cdot Z_1)^2. \end{aligned}$$

6.2 ECC operations

In this section we describe ECC operations at each level by following the top-down approach.

Point Multiplication: For the point multiplication we chose the method of Montgomery that maintains the relationship $P_2 - P_1$ as invariant [17]. It uses a representation where computations are performed on the x-coordinate only.

Point Addition and Doubling: We start from Eqs. (2), but the goal is to save some registers, as it is known that this part is usually the largest portion of the total area. As the previous formulae require 3 intermediate registers (2 for addition and 1 for doubling) [15], we eliminate 2 intermediate registers by introducing a few additional steps (cf. Algorithm 1). Therefore, we get the sequences of operations that require only one intermediate variable (T). Moreover, this value is manipulated only twice for addition and it could be even stored in some RAM. In this way we made a trade-off between speed and area as point operations require now 7 and 8 multiplications for addition and doubling (instead of 5 and 6 M respectively). Furthermore, point operation can be also easily balanced to achieve some simple side-channel protection such as in [3].

Algorithm 1 EC point addition and doubling

Require: $X_1, Z_1, X_2, Z_2, x_4 = x(P_2 - P_1)$
Ensure: $X(P_1 + P_2) = X(P_3) = X_3, Z_3$
1. $Z_3 \leftarrow X_2 \cdot Z_1$
2. $X_3 \leftarrow X_1 \cdot Z_2$
3. $Z_3 \leftarrow X_3 + Z_3$
4. $Z_3 \leftarrow {Z_3}^2$
5. $X_3 \leftarrow X_1 \cdot Z_2$
6. $X_3 \leftarrow X_3 \cdot X_2$
7. $X_3 \leftarrow X_3 \cdot Z_1$
8. $T \leftarrow x_4 \cdot Z_3$
9. $X_3 \leftarrow X_3 + T$

Require: $b \in \mathrm{GF}(2^n)$, X_1, Z_1
Ensure: $X(2P_1) = X(P_5) = X_5, Z_5$
1. $Z_5 \leftarrow {Z_1}^2$
2. $Z_5 \leftarrow {Z_5}^2$
3. $Z_5 \leftarrow b \cdot Z_5$
4. $X_5 \leftarrow {X_1}^2$
5. $X_5 \leftarrow {X_5}^2$
6. $X_5 \leftarrow X_5 + Z_5$
7. $Z_5 \leftarrow {X_1}^2$
8. $Z_5 \leftarrow Z_5 \cdot Z_1$
9. $Z_5 \leftarrow Z_5 \cdot Z_1$

An algorithm for field multiplication: The standard way to compute the product $c(x) = a(x) \cdot b(x) \mathrm{mod} f(x)$ is the one that uses convolution and to which we refer to as the classical algorithm [5].

The most compact architecture for this multiplication is the classical bit-serial multiplier (the MSB or the LSB multiplier) [5].

6.3 A Prototype Elliptic Curve Processor

The Elliptic Curve Processor (ECP) is shown in Fig. 1. The operation blocks are as follows:

- Control Unit(CU)
- Arithmetic Unit (ALU)
- Registers
- Memory: RAM

The Control Unit takes care of scalar multiplication, point operations and all conversions to suitable representation. It also commands the ALU which performs field multiplication, addition and inversion.

The largest part of the ALU is finite field multiplier, which is the MSB bit-serial multiplier [5]. The inversion operation is also performed by the multiplier using Fermat's theorem.

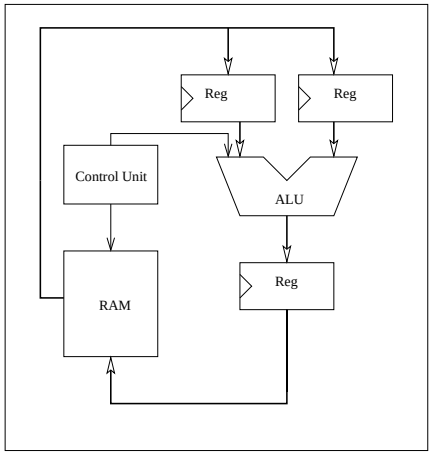


Fig. 1. Architecture of the elliptic curve processor.

6.4 Estimated results

Here we estimate the performance of the ECC processor for the field GF(2^{163}). The irreducible polynomial is the pentanomial $f(x) = x^{163} + x^7 + x^6 + x^3 + 1$.

One point multiplication takes $163 \cdot 15M = 2445M$. Conversion of coordinates $A \rightarrow P$ and $P \rightarrow A$ takes respectively $2M$ and $I + 2M$. Assuming that inversion is done by means of Fermat the total for conversion is around $300M$. This all together results in approximately $3000M$. One field multiplication (M) takes 163 cycles, which results in 489000 cycles for point multiplication. With a clock frequency of even $1MHz$ one point multiplication would take less than half a second, which is reasonably fast.

The estimated area complexity for the bit-serial multiplier is around $16n$, so for $n = 163$ we get around 2.6 kgates. Modular addition takes 163 XOR gates, so it sums up to around 3 kgates. The complexity of the FSMs used is hard to estimate, but as those are only some control logic it should not be too large. However, the registers that are required might take quite large area as 1FF is at least 6 NANDs. This is the most crucial aspect of the design. However, as 3 registers are absolutely necessary for ALU, we believe that this hardware component can be of the order of 5 kgates, depending on technology. We assume that EC parameters as well as other pre-calculated input values can be stored in memory blocks. It may further slow-down the performance but there is certainly enough margin for that according to the RFID specifications [1]. This also follows from the fact that the operating frequency for RFID tags is actually $13.56MHz$ according to the ISO 18000-3 standard while our estimates were made assuming the operating frequency of $1MHz$. Another option to minimize hardware complexity would be to decrease the field size. Namely, 163 bit long key sizes correspond to RSA keys that are much longer than 1024 bits [13]. More precisely, one could achieve that level of security with around 130 bits long ECC keys. Consequently, scaling down ECC parameters would result in a roughly linear decrease of hardware complexity. The fact that ECC is a suitable technology for RFIDs was also concluded in the work of Wolkerstorfer [27]. That work is the first complete ECC low-power and compact implementation that meets the constraints imposed by the EPC standard. Yet, our solution can be even smaller as our off-line authentication do not require full ECDSA algorithm to be executed on a single tag. That allows for further optimization with respect to area.

7 Concluding Remarks

In this paper we have shown that by equipping RFID-tags with I-PUFs, the tags become unclonable and hence suitable for anti-counterfeiting purposes. Using our protocols, both the physical cloning attack as well as the cloning attack based on (actively or passively) attacking the protocol between the tag and the reader can be prevented. It has been shown that the required protocols are feasible on an RFID-tag in the off-line situation.

Acknowledgement The authors thank Gregory Neven for his comments on an earlier version on this manuscript and for the nice and constructive discussions on this topic.

References

1. S. A. Weis A. Juels. Authenticating pervasive devices with human protocols. In V. Shoup, editor, *Advances in Cryptology: Proceedings of CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 293–308. Springer-Verlag, 2005.
2. M. van Dijk B. Gassend, D. Clarke and Srinivas Devadas. Controlled physical random functions. *In Proceedings of the 18th Annual Computer Security Conference*, December 2002.
3. L. Batina, N. Mentens, B. Preneel, and I. Verbauwhede. Side-channel aware design: Algorithms and architectures for elliptic curve cryptography over GF(2^n). In *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP'05)*, Samos, Greece, July 23-15 2005. IEEE Computer Society Press.
4. Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. In C. Cachin and J. Camenisch, editors, *Proceedings of Eurocrypt 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 268–286. Springer-Verlag, 2004.
5. T. Beth and D. Gollmann. Algorithm engineering for public key algorithm. *IEEE Journal on Selected Areas in Communications*, 7(4):458–465, May 1989.
6. I. Blake, G. Seroussi, and N. P. Smart. *Elliptic Curves in Cryptography*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1999.
7. Y. Dodis, J. Katz, S. Xu, and M. Yung. Strong Key-Insulated Signature Schemes. In Y. Desmedt, editor, *Proceedings of 6th International Workshop on Practice and Theory in Public Key Cryptosystems (PKC 2003)*, number 2567 in LNCS, pages 130–144. Springer-Verlag, 2003.
8. Y. Dodis, M. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In C. Cachin and J. Camenisch, editors, *Proceedings of Eurocrypt 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer-Verlag, 2004.
9. B. Gassend et al. Silicon physical unknown functions. *Proc. 9th ACM Conference on Computer and Communications Security*, November 2002.
10. D. Johnson and A. Menezes. The elliptic curve digital signature algorithm (ECDSA). Technical Report CORR 99-34, Department of Combinatorics & Optimization, University of Waterloo, Canada, February 24 2000. http://www.cacr.math.uwaterloo.ca.
11. M. Joye and S.-M. Yen. The montgomery powering ladder. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Proceedings of 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, number 2523 in Lecture Notes in Computer Science, pages 291–302. Springer-Verlag, 2002.
12. A. Juels. Strengthening EPC Tags against Cloning. March 2005. manuscript.
13. A. Lenstra and E. Verheul. Selecting cryptographic key sizes. In H. Imai and Y. Zheng, editors, *Proceedings of Third International Workshop on Practice and Theory in Public Key Cryptography (PKC 2000)*, number 1751 in Lecture Notes in Computer Science, pages 446–465. Springer-Verlag, 2000.
14. J.P. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In J. Kittler and M. Nixon, editors, *Proc. of the 3rd Conference on Audio and Video Based Person Authentication*, volume 2688 of *Lecture Notes in Computer Science*, pages 238–250. Springer-Verlag, 2003.
15. J. López and R. Dahab. Fast multiplication on elliptic curves over GF(2^m). In Ç. K. Koç and C. Paar, editors, *Proceedings of 1st International Workshop on*

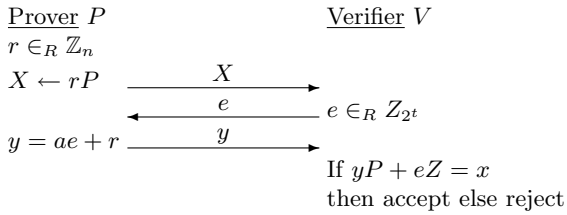
Cryptographic Hardware and Embedded Systems (CHES), volume 1717 of *Lecture Notes in Computer Science*, pages 316–327. Springer-Verlag, 1999.
16. A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
17. P. Montgomery. Speeding the pollard and elliptic curve methods of factorization. *Mathematics of Computation*, Vol. 48:243–264, 1987.
18. Michael Neve, Eric Peeters, David Samyde, and Jean-Jacques Quisquater. Memories: a Survey of their Secure Uses in Smart Cards. In *2nd International IEEE Security In Storage Workshop (IEEE SISW 2003)*, pages 62–72, Washington DC, USA, 2003.
19. B. Skoric P. Tuyls. Secret key generation from classical physics. *Philips Research Book Series*, September 2005.
20. R. Pappu. Physical one-way functions. *Science*, 297(6):2026, 2002.
21. G. J. Simmons. Identification of data, devices, documents and individuals. In *Proc. 25th Ann. Intern. Carnahan Conference on Security Technology*, pages 197–218, Taipei, Taiwan, ROC, October 1–3, 1991. IEEE.
22. B. Skoric, P. Tuyls, and W. Ophey. Robust key extraction from physical unclonable functions. In J. Ionnidis, A.D. Keromytis, and M. Yung, editors, *Proceedings of the Applied Cryptography and Network Security Conference 2005*, volume 3531 of *Lecture Notes in Computer Science*, pages 407–422. Springer-Verlag, 2005.
23. S. P. Skorobogatov and R. J. Anderson. Optical fault induction attacks. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, volume 2523 of *Lecture Notes in Computer Science*, pages 2–12. Springer-Verlag, 2002.
24. P. Tuyls and J. Goseling. Capacity and examples of template protecting biometric authentication systems. In D. Maltoni and A.K. Jain, editors, *Proceedings of Biometric Authentication Workshop*, volume 3087 of *Lecture Notes in Computer Science*, pages 158–170. Springer-Verlag, 2004.
25. P. Tuyls, B. Skoric, S. Stallinga, A.H.M. Akkermans, and W. Ophey. Information theoretical security analysis of physical unclonable functions. In A.S. Patrick and M. Yung, editors, *Proceedings of 9th Financial Cryptography and Data Security Conference*, volume 3570 of *Lecture Notes in Computer Science*, pages 141–155. Springer-Verlag, 2005.
26. M van Dijk and P. Tuyls. Robustness, reliability and security of biometric key distillation in the information theoretic setting. In N. Cerf and J. Cardinal, editors, *Proceedings of the 26th Benelux Symposium on Information Theory*, volume 26 of *Proceedings of the WIC*, 2005.
27. J. Wolkerstorfer. Scaling ECC Hardware to a Minimum. In ECRYPT workshop - Cryptographic Advances in Secure Hardware - CRASH 2005, September 6-7 2005. invited talk.

Appendix

Schnorr Identification Protocol based on ECDLP

Here we specify the Schnorr identification protocol based on ECDLP that could be performed in the case of off-line authentication. In this case a tag proves its identity to a reader in a 3-pass protocol.

1. **Common Input:** The set of system parameters in this case consists of: (q, FR, a, b, P, n, h). Here, q specifies the finite field, FR is a field representation, a, b, define an elliptic curve, P is a point on the curve of order n and h is the cofactor [10]. In this case of a tag authentication, most of these parameters are assumed to be fixed.
2. **Prover-Tag Input:** The prover's secret a such that $Z = -aP$.
3. **Protocol:** The protocol involves exchange of the following messages:



More precisely, steps of the protocol are:

- *Commitment by a Prover-Tag:* The tag picks $r \in_R \{0, \ldots, n-1\}$, and sends $x = rP$ to the reader.
- *Challenge from a Verifier-Reader:* The reader picks a number $e \in [1, 2^t]$ and sends it to the tag.
- *Response from a Tag:* The tag computes $y = ae + r$ and sends it to the reader.
- The verifier checks that $yP + eZ$ equals x. Check: $yP + eZ = (ae + r)P + eZ = aeP + rP + (-eaP) = rP = x$