Coupon Recalculation for the Schnorr and GPS Identification Scheme: A Performance Evaluation

Christoph Nagl and Michael Hutter

Institute for Applied Information Processing and Communications (IAIK),
Graz University of Technology, Inffeldgasse 16a, 8010 Graz, Austria
cnagl@sbox.tugraz.at, Michael.Hutter@iaik.tugraz.at

Abstract. One of the most important challenges in the last few years has been the integration of authentication services to low-cost RFID tags. Especially elliptic-curve-based implementations have proven to be a good option for asymmetric and light-weight cryptography. In this article, we evaluate two elliptic-curve based versions of the Schnorr and GPS identification schemes that have been designed for use in resource-constrained environments. Both schemes have been implemented on an RFID-tag prototype that runs at 13.56 MHz. Our results show that if the schemes make use of a pre-computation approach using *coupons*, Schnorr leads to a higher performance in terms of memory consumption, computational complexity, and communication bandwidth. Furthermore, we show that the challenge-response calculation of the Schnorr identification scheme can be performed even within the frame-delay timings of most common RFID standards such as ISO/IEC 14443 and ISO/IEC 15693 which encourages the use of Schnorr in scenarios where coupons are recalculated on the tag to allow fast and "on-the-fly" authentication.

Keywords: Radio-Frequency Identification, RFID, Authentication, Schnorr, GPS, Elliptic Curve Cryptography, ECC, Information Security.

1 Introduction

Radio-Frequency Identification (RFID) tags provide advantages in many practical applications. They have been not only integrated in inventory control and supply-chain management applications but also in security-related systems such as cashless payment and electronic passports. For these applications, tags have to implement cryptographic protocols in order to provide security services such as authentication, confidentiality, data integrity, and non-repudiation. This article focuses on the evaluation of two light-weight protocols that are based on identification schemes in order to provide authentication services for RFID tags.

RFID tags mainly consist of a tiny microchip that is attached to an antenna. The tags are able to communicate with a reader which generates an electromagnetic field. This field is used to power the tags on the one hand and to allow a mutual communication on the other hand. RFID tags that are powered passively

by the field of the reader have to provide a low-power consumption in order to allow a certain reading range. Furthermore, they have to be light-weight in terms of chip area to reduce the production costs for a large deployment.

In the last decades, many protocols have been proposed and designed to be applied in resource-constrained environments. The protocols provide authentication of entities using different schemes such as identification and signatures. While signature schemes allow a transferable proof of knowledge, identification schemes are used to authenticate entities in an actual communication. Next to challenge-response schemes, there exist witness-challenge-response schemes which prove the knowledge of a secret in a probabilistic rather than absolute way. One of the most prominent identification schemes is due to C. P. Schnorr [12]. He introduced a public-key scheme that allows entity authentication using a zero-knowledge proof-of-knowledge, i.e. the second party does not learn anything about the used secret. A similar scheme was proposed by M. Girault et al. [3] which provides faster authentication by following the approach of so-called *coupons*. Coupons are pre-calculated values that are independent of the input of the verifier. These coupons can be stored in the internal memory of RFID tags or can be transferred by the verifier during an authentication process. Due to the use of coupons, the authentication of the tag can be performed "on-the-fly" and much faster than other proposed schemes which have to calculate all needed values during one authentication phase.

However, the main drawback of stored coupons is a potential denial-of-service attack where an adversary might be able to exhaust all coupons of the tag. The coupons can then be either reloaded by an authenticated reader or they can be sent by the reader which has access to online precomputed coupons [2]. In the light of this fact, G. Hofferek et al. [5] proposed a solution by recalculating the coupons during the idle time of tags. When the tags are not actively participating in a communication with the reader, they are able to precompute the coupons by themselves. In particular, if there are many tags in the field, there is sufficient time for the tags to refill the memory with new calculated coupons. This approach allows fast authentication but lacks in the higher amount of needed hardware resources to recalculate the coupons.

In this article, we evaluate the performance of the elliptic-curve based version of the Schnorr and GPS identification scheme. Our evaluation shows that if the elliptic-curve based approach of coupon recalculation on the tag is implemented in practice, the Schnorr scheme provides better performance in terms of memory usage, computational complexity, and communication bandwidth. Moreover, we show that the scheme of Schnorr provides fast (on-the-fly) authentication that can be performed within the frame-delay timings of common RFID standards such as ISO/IEC 14443 and ISO/IEC 15693. Our findings are based on results of practical experiments on an RFID-tag prototype running at 13.56 MHz. In the case of coupon recalculation we therefore suggest the use of Schnorr to allow faster computation and higher authentication throughput.

The paper is structured as follows. In Section 2, we present an evaluation of the Schnorr and GPS identification scheme based on elliptic curves. Section 3

describes the setup used to perform the evaluation. Section 4 gives details about our implementation of Schnorr and GPS on an RFID-tag prototype. Results of the performed evaluations are presented in Section 5. Conclusions are drawn in Section 6.

2 Elliptic-Curve based Versions of the Schnorr and GPS Identification Scheme

The Schnorr identification scheme was first introduced by Claus Peter Schnorr in 1991 [12]. It provides a zero-knowledge proof that a *prover* is in possession of a given secret which is also known as *proof of knowledge*. The prover's identity can be proven to the *verifier* depending on a challenge c without ever revealing the secret s.

A procedural view of Schnorr's identification scheme for elliptic curves is given in Figure 1. Each party holds the same elliptic-curve domain parameters. We will use the following notation throughout this work. The order of the elliptic curve is denoted as n, the curve's modulus is called p, and the secret key is given as s. For protocol descriptions, we use r as commitment, c as challenge, and y as challenge-response. All elliptic-curve points are given in capitalized letters where P is the curve's base point, X is the witness point, and Z is the point which corresponds to the public key.

An example of an hardware implementation of the protocol is given by Y. K. Lee et al. [8].

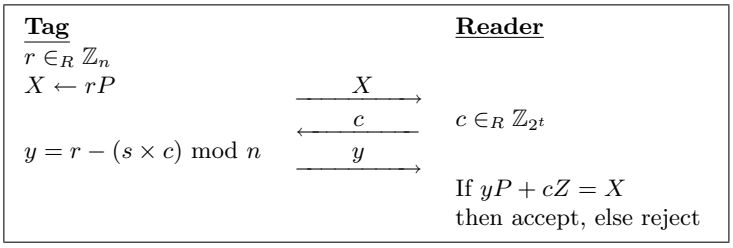

Fig. 1. ECSchnorr identification scheme.

The protocol starts with the prover generating a random commitment r used as point multiplication factor for a public point P (e.g. the curve's base point). The result of this point multiplication is called *witness* and is transferred to the verifier. Upon receipt, the verifier responds by sending a challenge c to the prover. The challenge is a random number limited by the *security parameter* t where $c \leq 2^t - 1$. The protocol of Schnorr is an interactive identification scheme that provides completeness, soundness, and honest-verifier zero-knowledge. That means that it provides the perfectly zero-knowledge property only when the tag interacts with a honest reader. For cheated readers, which may choose the challenge to be too large (super-polynomial), it loses the zero-knowledge property.

The protocol is secure against passive adversaries under the elliptic-curve discrete logarithm assumption but it is not secure against active and concurrent attacks.

$$y \cdot P + c \cdot Z \equiv X \tag{1}$$

$$\text{given } y = r - cs \bmod \text{n} \text{ and } Z = s \cdot P \tag{2}$$

$$r \cdot P - cs \cdot P + cs \cdot P \equiv r \cdot P \tag{3}$$

The public curve point Z is the public key, *i.e.* created by multiplication of the base point P with a secret key s according to the formula given in Equation 2. If Equation 1 holds, verification was successful and the prover has successfully authenticated to the verifier while otherwise not. Equation 3 shows the formulae of 2 inserted in Equation 1.

The second authentication protocol is an elliptic-curve based version of an interactive identification protocol proposed by M. Girault, G. Poupard, and J. Stern in 2001 (see Figure 2). It has been part of the European project NESSIE [10] and has been standardized in the ISO/IEC 9798-5 standard [6] in 2004. The protocol is similar to Schnorr but eliminates the modular reduction during the response calculation by performing the operations in $\mathbb{Z}$. Like the Schnorr protocol, GPS is proven to have the (statistical) zero-knowledge property if the challenge c is chosen not too large. It provides the honest-verifier zero-knowledge property and is thus only secure against active attacks under a given honest-reader assumption. In order to guarantee the statistical zero-knowledge property, we followed the equation $r = c \times s \times 2^{80}$ in our experiments as it has been advised by the authors [4].

An example of an hardware implementation that makes use of stored coupons is given by McLoone et al. [9].

Tag		**Reader**
$r \in_R \mathbb{Z}_{2^\rho}$		
$X \leftarrow rP$	$\xrightarrow{X}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_{2^t}$
$y = r - (s \times c)$	$\xrightarrow{y}$	
		If $yP + cZ = X$ then accept, else reject

Fig. 2. ECGPS identification scheme.

2.1 Comparing the Schnorr and GPS Identification Scheme

While GPS shares basic characteristics with the original Schnorr scheme, there exist some differences influencing the applicability for certain fields of applications. In the following section, we will describe the most influencing differences of both schemes.

The main difference of GPS compared to the Schnorr identification scheme is the elimination of the final reduction step modulus n in the challenge-response computation. This allows a faster authentication process and reduces the costs for a modular reduction unit in both software and hardware implementations.

Nevertheless, the costs for leaving the modular reduction step in the challenge-response computation are an increased size of the scalar r to be used for scalar multiplication with the common point P. In Girault et al. [4] the order of r is advised to be $c \times s \times 2^{80}$ where c is the order of the challenge c, s is the order of the secret key, and 2^{80} is an additional factor to compensate for the omitted reduction. Using a secret key s of 192 bit and a challenge c of 48 bit this adds up to $192 + 48 + 80 = 320$ bit for the scalar r used to create a witness point. Compared to the classical Schnorr identification scheme, which uses a scalar 192 bit for example, the GPS scheme needs a scalar of 320 bits to blind the secret key in the challenge-response calculation. While the time to create a witness does generally not come into account when using coupons (a coupon is composed of a pre-computed witness X and an associated commitment r) it is of major importance if tags need to recalculate coupons by themselves. This is the case we have analyzed in our work, indeed.

While the increased factor r influences the witness-creation time, it also has an impact on the challenge-response computation time. Following the advice mentioned above, the challenge response in the GPS scheme—denoted as $y = r - (c \times s)$—uses a commitment r of 320 bit, a challenge c of 48 bit, and a secret key s of 192 bit. While the multiplication of $c \times s$ is identical to the classical Schnorr scheme, the subtraction $r - c \times s$ gives a 320 bit result which has to be transmitted to the reader. The advantage of the GPS scheme of performing a much faster challenge-response computation, is however accompanied with a larger challenge-response size of 320 bit instead of 192 bit. This should not be neglected as it impacts memory consumption as well as required transmission bandwidth.

3 Experimental Setup

In this section, we present the setup used to evaluate the identification schemes of ECSchnorr and ECGPS. Our RFID-tag prototype is composed of a programmable ATMega128 microcontroller from Atmel, an antenna in the same shape as specified by the ISO 7810 standard, and an analog front-end that transforms the analog signals of the reader into the digital world of the microcontroller. The ATMega128 has an 8 bit RISC Harvard architecture and is clocked at 13.56 MHz. It provides 4 kB of RAM, 32 general purpose registers, and 128 kB of Flash memory. Both identification schemes have been implemented in ANSI C using the Crossworks IDE for AVR from Rowley Associates [11]. The program code was merged with an already existing framework that enables the tag prototype to communicate with a standard RFID reader according to ISO/IEC 14443-A [7]. As an RFID reader, we used an HF reader (TAGscan) from TAGnology [13]. The reader is controlled by a PC over a serial connection. We

used Matlab to control the communication process. The embedded Maple engine of Matlab was used to perform the elliptic-curve operations by using arithmetic with long integer numbers.

4 Implementation of the ECSchnorr and ECGPS Identification Schemes on an RFID-Tag Prototype

The point multiplication is one of the most time and resource consuming operation of elliptic-curve based implementations. Especially software implementations suffer from long computation times which are often not available in practice. The common RFID ISO/IEC 14443-A standard specifies the physical characteristics, the power and signal interfaces, and the initialization and anti-collision sequences (to handle multiple devices in the field of a reader) of tags. This standard defines also the frame-delay timings between the reader and the tags, i.e. the time between the end of the last reader challenge and the first modulation of the tag response. The smallest frame delay from the reader to the tag is specified to about 86 μsec. In contrast, the ISO/IEC 15693 standard, which is one of the most prominent standards for RFID vicinity cards, specifies this time to 320 μsec. Within these timings, the tag should be able to compute the response data in order to send the answer subsequently.

In our experiments, we followed the approach of [5] and precomputed one single coupon during the idle time of the tag prototype. The coupon (witness X and the commitment r) is computed when the tag is powered up and is recalculated directly after one successful authentication process. This mechanism allows us to store only one coupon in the non-volatile memory of the prototype and to perform a fast authentication on demand of the reader.

The authentication process of the RFID-tag prototype (see Figure 3) is triggered by a proprietary ISO/IEC 14443-A *AUTHENTICATE* command sent by the RFID-reader application. The tag answers by immediately sending the precomputed witness X to the RFID reader. The RFID-reader application stores the witness and challenges the tag with a challenge c using the *SEND_CHALLENGE* command. Following the identification scheme, the tag computes a challenge-response and sends it back to the reader which validates the authentication.

In order to comply with the restricted timings of the ISO/IEC 14443-A standard, we followed an *answer-on-demand* approach. The reader sends a *SEND_CHALLENGE* command to the tag which initiates the challenge-response computation on the tag. After a certain amount of time given by the time required for the tag to compute the challenge response, the reader sends a *GET_RESPONSE* to the tag. The tag responds by commiting the challenge response. Using this approach, we have been able to send an answer within the standardized frame-delay timings. Note that this approach can be omitted if the calculation of the challenge response can be performed within the minimum time specified by the respective standard. Section 5 discusses the computational complexity of the challenge-response calculation in software and estimates the costs for an hardware implementation to be performed within the given frame-delay timings.

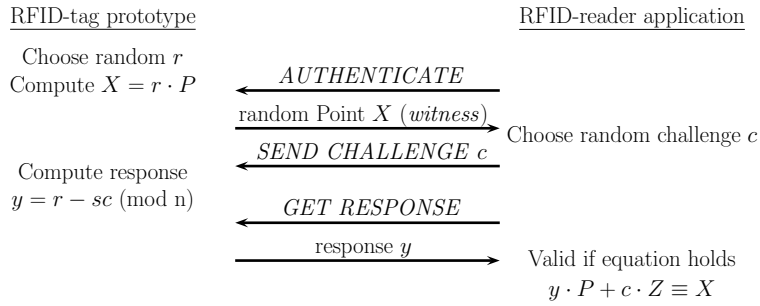

Fig. 3. Sequence of RFID transmissions of the implemented ECSchnorr scheme.

In order to communicate with the tag prototype and to perform the Schnorr and GPS authentication, proprietary commands have been defined. One example of a proprietary command that is sent by the reader is shown in the following.

$$\underbrace{t \quad \underbrace{07}_{Payload\ length} \quad \underbrace{03}_{Flags} \quad \underbrace{78}_{Command\ ID} \quad \underbrace{E02341BD921A}_{Payload}}_{ISO14443A\ Message}$$

The given example describes the construction of a proprietary reader command that sends the *SEND_CHALLENGE* command. The preamble is represented by the letter "t" followed by the byte length of the transmitted payload and an option flag that is used to enable the calculation and verification of the Cyclic Redundancy Check (CRC). The payload includes the proprietary command ID "78" and the following 48 bit challenge "E02341BD921A".

Because the implementation on the tag prototype represents long integers with LSB on array indices 0 and MSB on arbitrary indices to make the solution more portable if other elliptic curves with bigger parameters are desired, operands need to be reversed. To keep the tag from reversing operands, the RFID-reader application takes care of reversing these operands.

5 Evaluation Results

In this section, the results obtained from the software implementation described in the previous section are discussed in terms of required memory usage, computational complexity, and communication bandwidth.

5.1 Static Memory Usage

The code sizes of finite-field operations, elliptic-curve operations, and the authentication-protocol process of our implementation on the RFID-tag prototype are given in Table 1. The implementation of ECSchnorr comprising the code for projective coordinates and the Montgomery Ladder multiplication uses 8.028 kB

of Flash memory and 1.856 kB of RAM while the ECGPS implementation uses 7.750 kB of Flash memory and 1.887 kB of RAM (the memory usage of the required ISO/IEC 14443-A interface are not included in the table).

Project Item	ECSchnorr		ECGPS	
	Code [bytes]	Data [bytes]	Code [bytes]	Data [bytes]
ec_authentication.c	410	241	382	241
ec_arithmetic.c	1,824	289	1,824	289
finite_arithmetic.c	4,322	674	4,072	674
bitutils.c	320	24	320	24
main.c	1,152	628	1,152	659
Total memory usage	8,028	1,856	7,750	1,887

Table 1. Static and dynamic memory consumption of ECSchnorr and ECGPS on our RFID-tag prototype.

The given sizes show some differences between the Schnorr and GPS implementation. The main difference between the two implementations is a slightly higher dynamic memory usage of the ECGPS protocol. This is due to the increased variable length of the enlarged commitment r and the larger size of the response y. Furthermore, it shows that ECGPS needs less static memory which has its reason in the fact that the reduction operation can be left out as opposed to the ECSchnorr protocol. In view of hardware implementations, ECGPS would need an increased size in memory usage on the one hand while the reduction unit can be saved on the other hand. Note that our implementations are straight-forward implementations with no special attention towards code optimization.

5.2 Computational Complexity

This section describes evaluation results of tag authentication tests. The RFID-tag authentication is composed of two major tasks: the witness creation and the challenge-response computation. While GPS optimizes the second task and offers a considerably faster challenge-response computation, it aggravates the witness creation process because of the larger scalar value. In the following, our results for these tasks are described in a more detail.

Witness Creation As the point multiplication requires numerous calls to subsidiary elliptic-curve operations of point addition and point doubling, this operation clearly dominates the overall execution time. The most time-consuming underlying finite-field operation is the operation of inversion. In fact, eliminating the inversion operation by choosing a projective-coordinate representation to avoid costly divisions is a major time gain. The instruction cycles of point addition and point doubling as well as the Montgomery Ladder multiplication

using projective coordinates are listed in Table 4. The results for point multiplication shows an elementary difference for the operation of point multiplication between ECSchnorr and ECGPS.

Operation	**ECSchnorr**	**ECGPS**
Single Point Addition [cycles]	561,863	561,863
Single Point Doubling [cycles]	560,604	560,604
Point Multiplication [cycles]	217,020,276	358,839,445
@13.56 MHz [seconds]	∼16.0	∼26.5
@1 MHz [seconds]	∼217.0	∼358.1

Table 2. Instruction cycles of elliptic-curve operations using the base point of NIST p-192 for addition and doubling. Addition performed as $P + 2P$, doubling performed as $2P$, and multiplication was performed using point P and a random scalar r of two different dimensions (192 bit for ECSchnorr and 320 bit for ECGPS).

The witness creation (coupon recalculation) of the ECGPS schemes consumes significantly more time in our implementation than the ECSchnorr scheme. The ECGPS requires a point multiplication using a 320 bit scalar while ECSchnorr uses a 192 bit scalar factor. The given computation time closely matches the estimated cycle counts of $k_{192\,bit}P \cdot \frac{320}{192} \approx 357 \cdot 10^6$ cycles giving a time factor of $\times 1.6$ for the ECGPS scheme in our implementation.

Challenge-Response Computation Challenge-response computation in the case of ECSchnorr mainly depends on the reduction of the product $c \times s$ being subtracted from commitment r. The modular multiplication step $c \times s$ is a 48×192 bit multiplication requiring a reduction modulus n from 240 bit to 192 bit. In our software implementation, we applied the Barrett reduction algorithm to perform the modular reduction of the challenge response.

Table 3 illustrates the cycle costs for the suboperations of the overall challenge-response computation for the ECSchnorr implementation. The multiplication was performed as 192×192 bit multiplication, the subtraction and addition were performed both on 192 bit operands, and the reduction was performed on the product of a 192×48 bit multiplication. Given the cycle counts for each sub-operation, the challenge-response computation time for ECSchnorr can be given as

$$MUL_{192\times48} + REDUCTION_{240\rightarrow192} + MOD\_SUB_{192-192} = 145,426\,\text{cycles}.$$

Given a clock frequency of 13.56 MHz, this computation would take a time of 10.724 msec for our software implementation. In the following, we estimate the costs of a hardware implementation to perform the challenge-response computation. The modular addition and subtraction are estimated to 26 cycle counts

and the modular reduction (using Montgomery-reduction method) to 936 cycles. These estimated cycle counts are proven to be realistic in existing RFID hardware implementations, see for example the work of A. Auer [1] (page 72 and 73, Table 5.4 and 5.6). With these estimations we result in about 962 cycles which corresponds to 70.94 μsec at 13.56 MHz and about 320 μsec at 3 MHz. An hardware implementation of Schnorr would therefore perform the challenge-response computation within the standardized frame-delay timings of common RFID standards. It should be noted that the computational power required to perform such calculations is generally high and may not apply to average RFID-tags.

Operation [cycles]	**Software Implementation**	**Hardware Estimations**
Modular Addition	1,296	26
Modular Subtraction	1,218	26
Modular Multiplication	144,330	936

Table 3. Measured and estimated cycle counts for modular arithmetic of the challenge-response computation in software and hardware.

5.3 Communication Bandwidth

Next, we present results of performed communication bandwidth evaluation of the Schnorr and GPS identification schemes. We analyzed the time for byte transmissions between the tag prototype and the reader for one single authentication. In particular, we have measured the timings for initialization and anticollision, witness transmission, challenge transmission, and challenge-response transmission. All results are based on the ISO/IEC 14443-A RFID standard for proximity cards.

The results are given in Table 4. The time for the tag initialization and anticollision according to ISO/IEC 14443-A needs about 4.8 ms. This time has been obtained by measuring the time between the first request command (REQA) of the reader and the last sent SAK command of the tag prototype. After the initialization, the proprietary AUTHENTICATE command and the 192 bit of the witness are transmitted to the reader. The witness transmission takes about 2.3 ms. The transmission of the challenge (48 bit) needs about 0.6 ms. The transmission of the challenge response takes about 2.3 ms for the ECSchnorr scheme and 3.8 ms for the ECGPS scheme. The increased transmission time is due to the larger size of the challenge response. Note that the time between the sent data frames is not given in the table and has to be considered for a total authentication-time evaluation. However, the total payload of all transmissions of the proprietary commands is 432 bits for the ECSchnorr scheme and 560 bit for the ECGPS scheme. The time of one single authentication including the frame-delay timings between all sent data frames, takes about 13 ms to 15 ms in total.

Operation	ECSchnorr [ms]	ECGPS [ms]
Initialization and Anticollision	4.850	4.850
Witness transmission(192 bit)	2.311	2.311
Challenge transmission(48 bit)	0.578	0.578
Response transmission	2.311	3.851

Table 4. Transmission time for a single tag authentication based on ISO/IEC 14443-A communication.

6 Conclusions

In this article, we present an evaluation of the elliptic-curve based variants of the Schnorr and the GPS identification scheme. We implemented the schemes on an RFID-tag prototype and analyzed the memory usage, computational complexity, and communication bandwidth. For all our experiments, we considered the coupon recalculation approach by precomputing the commitment and witness as a coupon which is used during the authentication process. The coupon is calculated using an elliptic-curve point multiplication. The results show that if the coupons are recalculated on the tag during the idle times of tags, ECSchnorr becomes faster due to the smaller size of the scalar of the witness creation. In addition, the modular reduction that is needed in the challenge-response computation of the ECSchnorr scheme can be performed within most prevalent RFID standards such as ISO/IEC 14443 and ISO/IEC 15693. If coupon recalculation using elliptic curves is implemented in practice, we therefore suggest the use of the Schnorr identification scheme. In particular, if tags implement elliptic-curve operations and if enough time for an authentication process is available, we further advise the use of standardized signature schemes like the Elliptic Curve Digital Signature Algorithm (ECDSA). Signature schemes make use of additional hash functions but provide enhanced services such as data integrity, non-repudiation, and message authentication.

Acknowledgements.

This work has been supported by the European Commission under the Sixth Framework Programme (Project Collaboration@Rural, Contract Number IST-FP6-034921) and by the Austrian Government through the research program FIT-IT Trust in IT Systems (Project CRYPTA, Project Number 820843).

References

1. A. Auer. Scaling Hardware for Electronic Signatures to a Minimum. Master's thesis, Graz University of Technology, 2008.

2. B. Calmels, S. Canard, M. Girault, and H. Sibert. Low-cost cryptography for privacy in RFID systems. In J. Domingo-Ferrer, J. Posegga, and D. Schreckling, editors, *Smart Card Research and Advanced Applications, 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tarragona, Spain, April 19-21, 2006, Proceedings*, volume 3928 of *Lecture Notes in Computer Science*, pages 237–251. Springer, April 2006.
3. M. Girault and D. Lefranc. Public Key Authentication with One (Online) Single Addition. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems – CHES 2004, 6th International Workshop, Cambridge, MA, USA, August 11-13, 2004, Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 413–427. Springer, August 2004.
4. M. Girault, G. Poupard, and J. Stern. On the fly authentication and signature schemes based on groups of unknown order. *Journal of Cryptology*, 19:463–487, 2006.
5. G. Hofferek and J. Wolkerstorfer. Coupon Recalculation for the GPS Authentication Scheme. In G. Grimaud and F.-X. Standaert, editors, *Proceedings of the Eight Smart Card Research and Advanced Application Conference, CARDIS '08, September 8-11, 2008, London, UK, Proceedings*, volume 5189 of *Lecture Notes in Computer Science*, pages 162–175. Springer, September 2008.
6. International Organisation for Standardization (ISO). ISO/IEC 9798 Part 5: Information technology – Security techniques – Entity authentication – Mechanisms using zero knowledge techniques, December 2004.
7. International Organization for Standardization (ISO). ISO/IEC 14443: Identification Cards - Contactless Integrated Circuit(s) Cards - Proximity Cards, 2000.
8. Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede. Elliptic-Curve-Based Security Processor for RFID. *IEEE Transactions on Computers*, 57(11):1514–1527, November 2008.
9. M. McLoone and M. J. B. Robshaw. Public Key Cryptography and RFID Tags. In M. Abe, editor, *Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings*, volume 4377 of *Lecture Notes in Computer Science*, pages 372–384. Springer, February 2007.
10. B. Preneel et al. NESSIE Security Report, D20, 2003. Available online at `http://www.nessie.eu.org`.
11. Rowley Crossworks IDE. Crossworks v1.4 and v2.0 for AVR. `http://www.rowley.co.uk`.
12. C.-P. Schnorr. Efficient Identification and Signatures for Smart Cards. In G. Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceeding*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1990.
13. TAGnology RFID GmbH. TAGscan 13.56 MHz Multi-ISO. `http://www.tagnology.com`.