RFID AUTHENTICATION PROTOCOLS BASED ON ELLIPTIC CURVES
A Top-Down Evaluation Survey

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

Keywords: RFID, Elliptic Curve Cryptography, Authentication, Information Security

Abstract: Authentication of RFID tags is a challenging task due to the resource-constrained environment they are operating in. In this article, we present a top-down evaluation survey for RFID-tag authentication based on elliptic curves. First, we describe a general model to characterize different state-of-the-art public-key techniques that provide entity and message authentication. Second, we present practical results of evaluations of elliptic-curve based identification and signature schemes. We analyzed and compared the ECSchnorr, ECOkamoto, and ECGPS protocol with respect to their computational complexity, storage requirement, and communication overhead. Furthermore, we examine different certificate-management solutions in RFID applications and give size estimations from simulated scenarios. Our studies have led us to the result that elliptic-curve based identification schemes and signature schemes have nearly the same complexity. ECOkamoto provides more enhanced security features while ECGPS has been designed for efficient "on-the-fly" authentication using offline pre-computations. ECSchnorr might be preferred if primitive computation is performed online during tag authentication.

1 INTRODUCTION

Radio Frequency Identification (RFID) is an emerging technology that has been already applied to existing applications such as supply-chain management and inventory control. An RFID tag is composed of a tiny microchip that is attached to an antenna. While these tags are stuck onto objects, they can be identified and tracked by readers over an electromagnetic field. RFID tags facilitate the process and the management of products during their life cycle. In the last few years, many RFID devices have also been applied to new applications where security plays a major role. These applications are, for example, electronic payment, medical care, and access control. RFID tags are very useful in this context by authenticating products and also to thwart product counterfeiting.

RFID tags operate in a constraint environment and are typically deployed in a large scale. It is evident that such tags require appropriate protocols that are extremely light-weight in terms of power, area, and costs. Large effort has been made by the cryptographic community to develop protocols that are suitable for RFID devices. One promising approach is to use elliptic-curve cryptography (ECC). Elliptic curves have proven to be applicable for resource-constrained devices like RFID tags. For the same security level, they need much smaller key sizes as opposed to other asymmetric techniques like RSA. However, there exist many proposals for authentication protocols that mainly differ in the used schemes and primitives. The question of which kind of protocol becomes more appropriate for a given application keeps therefore often not straightforward to answer.

In this article, we focus on the evaluation of different authentication protocols that are based on elliptic curves. First, we define and describe a general model for state-of-the-art public-key techniques which provide authentication as a cryptographic service. Second, we present practical results of performance measurements of authentication protocols that have been especially designed for low-resource devices. We analyze both identification and signature schemes of the elliptic-curve variants of the Schnorr (Schnorr, 1990), Okamoto (Okamoto, 1993), and GPS (Girault et al., 2006) protocol. Furthermore, we analyze three different scenarios for efficient certificate management in RFID applications. Our investigations show that the described identification schemes and signature schemes have nearly the same complexity in terms of memory usage, computational effort, and communication. Point multiplication is the most complex operation while additional hash computations have a minor impact on the overall performance. The right choice for a protocol depends on several issues such as security (ECOkamoto), online computation of the scalar multiplication (ECSchnorr), or pre-computed coupons offering "on-the-fly" authentication (ECGPS).

The remainder of this article is structured as follows. Section 2 gives related works on that topic. In Section 3, different public-key techniques are described providing entity and message authentication. Section 4 presents authentication protocols that have been designed to be suitable for low-resource devices. Section 5 discusses identification and signature schemes on RFID tags. In Section 6, we describe our evaluation framework and discuss the used performance metric. Results are given in Section 7. The conclusion are drawn in Section 8.

2 Related Work

There exist many articles that focus on elliptic-curve cryptography and low-resource implementations on RFID devices. Large effort in that context has been made by L. Batina et al. (Batina et al., 2006). They have analyzed hardware implemenations of the Schnorr and Okamoto protocol in the case of elliptic curves over $\mathbf{F}_{2^m}$. Girault et al. (Girault et al., 2007) presented hardware results of the GPS authentication protocol based on an FPGA prototype implementation. Like the work of Girault, M. McLoone et al. (McLoone and Robshaw, 2007) assumed the use of so-called *coupons*, which represent (hashed) pre-computed commitments, and implemented only the response calculation of the identification scheme without performing the scalar multiplication. Low-resource implementations of elliptic-curve processors in general have been investigated by J. Wolkerstorfer (Wolkerstorfer, 2005), R. Schroeppel et al. (Schroeppel et al., 2003), G. Gaubatz et al. (Gaubatz et al., 2005), and Kumar et al. (Kumar and Paar, 2006).

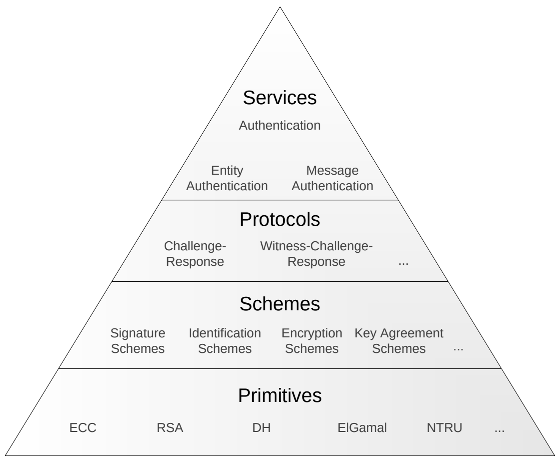

Figure 1: Different public-key techniques lead to the cryptographic service of authentication.

3 Public-Key Techniques

There exist many applications nowadays that have to provide security services such as authentication, confidentiality, non-repudiation, or data integrity. Although, these services can be provided by using symmetric cryptography, public-key techniques have often been preferred in practice to tackle the problem of key distribution. However, cryptographic services can not only be achieved by a specific public-key technique. In contrast, there exist various public-key techniques which lead to the same cryptographic service. Authentication, for example, can be achieved by various cryptographic protocols that are based on different schemes and primitives. Both digital-signature schemes and zero-knowledge proof-of-knowledge schemes can be used in protocols to follow the same security goal of entity authentication. Although many of these protocols have been standardized in the last decade, less effort has been made to uniformly classify these protocols according to their underlying public-key technique. Hence, we first describe a general model to characterize different state-of-the-art public-key techniques. The model separates all techniques into four levels: services, protocols, schemes, and primitives. Figure 1 shows the natural structure of this model as an instance of authentication.

3.1 Services

Authentication has become one of the most important services of cryptographic applications. It is the process where someone or something is confirmed to be authentic and that a verifier is assured of the identity of a prover. The authentication service can be separated into two main types: entity authentication and message authentication. Entity authentication is referred to as a real-time service where both parties the prover and the verifier actively participate in a communication. Thus, the identity of the prover is guaranteed in a timeliness fashion. In contrast, message authentication refers to the service where a verifier gets assurance of a dedicated message that was generated by the prover. The difference between entity and message authentication is therefore as follows. In the first case, an entity or person is authenticated exactly during an actual communication protocol. The proof of authentication is here not transferable. In the second case, a prover digitally signs a message which can be verified even after the actual communication process. This is useful when one party is not active in the communication, which may happen in Internet applications, for instance. However, the latter case implicitly provides two additional cryptographic services that are data integrity and non-repudiation of the signed message which makes the use of digital signatures more attractive in practice.

3.2 Protocols

Protocols are typically used to provide one or more cryptographic services. They define a sequence of steps for two or more entities that would like to achieve these services. Such protocols differ in several properties. First, they can be built upon different schemes. Entity authentication protocols, for example, can be based upon encryption schemes, signature schemes, or identification schemes. Protocols that provide message authentication can be constructed by encryption schemes or signature schemes. Second, protocols may differ in the number of message passes between the involved entities. The number of message passes of a protocol actually depends on the required message passes of the underlying schemes and also on the protocol specific passes that are needed to achieve a certain service. Challenge-response protocols usually need two message passes but there exist also other protocols that need more or even less message passes than the sum of message passes of the individual schemes they involve. Third, there are protocols that provide services for all or a subset of involved parties. In an unilateral authentication protocol, for instance, only one entity is authenticated, whereas in a mutual authentication process both parties get assured to be authentic (Menezes et al., 1997).

3.3 Schemes

Schemes are the basic building blocks of cryptographic protocols. They provide a set of cryptographic operations and methods that are typically combined within a protocol in order to achieve a certain security service. One of the most commonly used schemes are encryption schemes, signature schemes, identification schemes, and key-agreement schemes. Encryption schemes provide encryption and decryption routines and make use of encryption primitives such as ElGamal or RSA. Signature schemes provide a signature generation and signature verification operation. These schemes use additional cryptographic functionalities such as hash and redundancy functions. They are based on signature primitives like the (EC)DSA. Identification schemes, in contrast, offer a prove and verify operation. They typically provide additional functionalities such as random numbers that are needed to compute the challenges, for instance. Key-agreement schemes offer methods to agree on a common session key and are based on primitives such as DH (Diffie and Hellman, 1976) and MQV (Law et al., 2003). In addition to the described operations, schemes also offer methods for key management such as the generation and verification of public keys.

3.4 Primitives

The lowest level of our model are cryptographic primitives. Primitives are algorithms that rely on mathematical hard problems. The intractability of these problems are typically exploited to provide the security of a cryptographic protocol. The problem of factoring large integers (e.g. RSA), solving discrete logarithms (e.g. ElGamal, DSA, DH), or solving elliptic curve discrete logarithms (e.g. ECDSA, ECDH) are the most prominent mathematical problems nowadays. However, there are also other problems known where protocols and schemes are based upon. Such protocols are, for instance, the NTRU cryptosystem which is based on the Shortest Vector Problem in a Lattice or the Goldwasser-Micali cryptosystem that is based on the Quadratic Residuosity Problem.

4 Authentication Protocols

The fundamental security goal of authentication protocols is the resistance against impersonation through both passive and active attacks. Passive attacks extract information by passively monitoring (eavesdropping) multiple protocol executions. In active attacks, an adversary plays the role of the verifier and extracts information by actively interacting with the prover. In the following, we focus on authentication protocols that are at least provably secure against passive attacks. They are constructed of different types of schemes but they use the same cryptographic primitive which is based on the intractability of the elliptic curve discrete logarithm problem (ECDLP). First, we describe protocols providing entity authentication. These protocols are composed of encryption schemes, signature schemes, or (zero-knowledge) identification schemes. Second, we describe message authentication protocols that are constructed using signature schemes. All protocols involve two parties (a reader and a tag) and provide unilateral authentication.

The following notation is used throughout the paper. Common parameters to all entities are: the underlying finite field $\mathbf{F}_q$, the elliptic curve parameters a and b, the curve point P with order n, and the public key Q. Finite elliptic-curve points are upper case such as P and Q. x is the x-coordinate of P and $\bar{x}$ is the integer representation of the binary representation of x. The private key is denoted by s, the protocol challenge is denoted by c, and the response is denoted by y. All used random numbers (ephemeral keys) are referred to r and the security parameter t defines the number of bits for the challenge. The variable e represents the output of the one-way hash function h.

4.1 Entity Authentication based on ECC Encryption Schemes

One way to reach entity authentication is to demonstrate the knowledge of a private key through the decryption of a ciphertext. In view of ECC, this can be achieved by applying the ECC-based variant of the ElGamal encryption scheme (ElGamal, 1984), for instance. In Figure 2, the principle of an entity-authentication protocol is shown that is based on an encryption scheme. First, the reader generates a random number r and encrypts r together with the identifier of the tag. Then it sends the result c, the identifier ID and the witness e of the random number r to the tag. The tag decrypts the message and verifies the obtained values. It sends the decrypted value y to the reader which accepts if it is equal to the previous generated random number r. Note that the random number is necessary to prevent replay attacks, the identifier avoids reflection attacks, and the witness $e = h(r)$ is used to preclude chosen-text attacks.

Although encryption-based entity-authentication protocols are semantically secure and provide se-

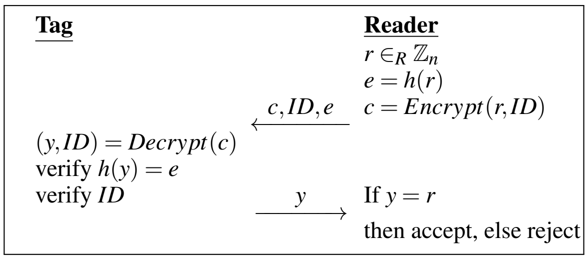

Figure 2: Entity authentication based on an encryption scheme.

curity against passive and active attacks, they lack several basic needs which limit the use in practical applications. First, they rely on encryption algorithms which would not be licensed for an export to external countries. Second, they often make use of additional functionalities such as hash functions and timestamps. These building blocks are usually considered to be difficult to construct especially on resource-constrained devices like RFID tags.

4.2 Entity Authentication based on ECC Signature Schemes

Another way to obtain entity authentication is to demonstrate the knowledge of the private key through the signing of a challenge. Figure 3 shows an authentication protocol according to ISO/IEC 9798-3 (International Organisation for Standardization (ISO), 1993). The protocol is structured as follows. First, the reader generates a random challenge c_1 and sends it to the tag. The tag now digitally signs the challenge and sends the signature y together with the identifier of the reader and the public-key certificate of the tag to the reader. The reader verifies both the certificate of the tag and the signature and it accepts the tag if the verification succeeded.

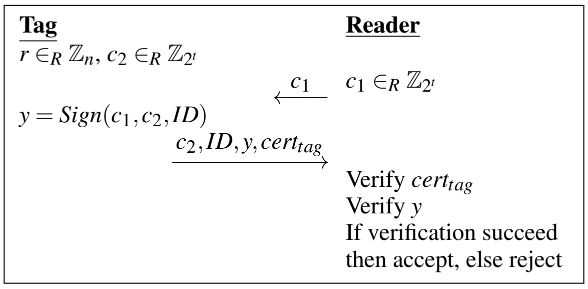

Figure 3: Entity authentication based on a signature scheme.

Entity-authentication protocols that are based on signature schemes typically rely on a one-way hash function to prevent existential-forgery attacks, i.e. to find another challenge that results in the same signature.

In the following, we describe entity-authentication protocols that are based on identification schemes. These schemes have been especially adapted to work also without encryption and hash functions. They provide an interactive proof-of-knowledge and have become more and more important for RFID applications because of their small footprint and ability to pre-compute values to perform "on-the-fly" authentication (Girault et al., 2006).

4.3 Entity Authentication based on ECC Identification Schemes

We describe three entity-authentication protocols that are based on identification schemes. As opposed to the previous described challenge-response protocols, the given proofs are probabilistic rather than absolute. A verifier (reader) is convinced by a prover (tag) to be in possession of the private key. The described schemes consist of three communication passes that may be executed several times (sequential version) or only in a single round (parallel version).

The first protocol is given in Figure 4. It is an ECC-variant of the authentication protocol published by C. Schnorr (Schnorr, 1990). First, the tag generates a random number r and calculates the elliptic-curve point X as a witness. The tag sends the witness together with its certificate to the reader. Second, the reader verifies the certificate and generates a challenge c which it sends to the tag. The tag now calculates y and send it as a response back to the reader which accepts or rejects the tag. 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 thus secure against passive adversaries under the elliptic-curve discrete logarithm assumption but it is not secure against active attacks.

Tag		**Reader**
$r \in_R \mathbb{Z}_n$		
$X \leftarrow rP$	$\xrightarrow{X, cert_{tag}}$	verify $cert_{tag}$
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_{2^t}$
$y = r + (s \times c) \bmod n$	$\xrightarrow{y}$	
		If $yP + cQ = X$ then accept, else reject

Figure 4: ECSchnorr authentication protocol.

The second authentication protocol is due to T. Okamoto (Okamoto, 1993) shown in Figure 5. It is a variant of the prior described Schnorr protocol but provides additional security against active and concurrent attacks. First, the tag generates two random numbers r_1 and r_2. Using these two random numbers, it calculates the elliptic-curve point X and sends it together with the certificate to the reader. As a second step, the reader picks a challenge c and sends it to the tag. Then, the tag calculates two responses y_1 and y_2 and sends it back to the reader which verifies the authenticity of the tag. As opposed to the protocol of Schnorr, Okamoto is an interactive identification scheme that provides a witness-indistinguishable proof-of-knowledge.

Tag		**Reader**
$r_1, r_2 \in_R \mathbb{Z}_n$		
$X \leftarrow r_1P_1 + r_2P_2$	$\xrightarrow{X, cert_{tag}}$	verify $cert_{tag}$
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_{2^t}$
$y_1 = r_1 + (s_1 \times c) \bmod n$		
$y_2 = r_2 + (s_2 \times c) \bmod n$	$\xrightarrow{y_1, y_2}$	
		If $y_1P_2 + y_2P_2 + cQ = X$ then accept, else reject

Figure 5: ECOkamoto authentication protocol.

The third authentication protocol is an ECC-variant of an interactive identification protocol proposed by M. Girault, G. Poupard, and J. Stern in 2001. It is part of the European project NESSIE (Preneel et al., 2003) and has been standardized in the ISO/IEC 9798-5 standard (International Organisation for Standardization (ISO), 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 (Girault et al., 2006).

4.4 Message Authentication based on ECC Signature Schemes

Any of the previous described (interactive) identification schemes can be transformed into a (non-interactive) signature scheme using the transformation technique proposed by A. Fiat and A. Shamir (Fiat and Shamir, 1987). The non-

$$\begin{array}{lcl}
\underline{\textbf{Tag}} & & \underline{\textbf{Reader}} \\
r \in_R \mathbb{Z}_{2^p} & & \\
X \leftarrow rP & \xrightarrow{X, cert_{tag}} & \text{verify } cert_{tag} \\
 & \xleftarrow{c} & c \in_R \mathbb{Z}_{2^t} \\
y = r + (s \times c) & \xrightarrow{y} & \\
 & & \text{If } yP + cQ = X \\
 & & \text{then accept, else reject}
\end{array}$$

Figure 6: ECGPS authentication protocol.

interactive signature schemes can be constructed by replacing the challenge c by the outcome of a cryptographic hash-function h. The input of the hash function is the witness x concatenated with the message m that has to be signed $c = h(x,m)$. For the security of the signature scheme it is necessary to apply a hash function that is collision resistant, i.e. that two messages do not lead to the same hash value. Furthermore, the size of the hash function must be chosen not too small to prevent existentially unforgeable signatures under adaptive chosen-message attacks. Therefore, we applied a 160-bit SHA-1 hash function to all described protocols and transformed them into signature schemes.

5 Identification vs. Signature Schemes

There are two major authentication services which are entity authentication and message authentication. The kind of service that an RFID tag has to provide basically depends on the application. A typical scenario where authentication plays an important role, is protection against counterfeiting goods. There, RFID tags proof the origin of objects to which they are attached to. These tags are typically powered passively and are implicitly involved in an active communication to the reader. Such tags actually have to provide at least entity authentication. There is no need for RFID tags to provide additional cryptographic services such as data integrity or non-repudiation. This actually discourages the use of signature schemes in that scenario.

However, there exist also other scenarios where it becomes important for a tag to provide message-authentication capabilities. Suppose the case where the reader has to transfer the proof of the tag's origin or has to prove the authenticity of a tag even at some later instant of time. Digital signatures provide these services which can not be realized using identification schemes.

In the following, we focus on the evaluation of different identification and signature schemes. Each scheme provides specific services and tradeoffs among costs, security, and performance.

6 Protocol Evaluation

In order to evaluate the performance and efficiency of the described authentication protocols, we have developed a framework in Java. The framework is able to model an RFID system involving several components such as a tag, reader, air-interface, RFID-communication protocol, cryptographic protocol, and public-key management capability. Using this model, we have been able to simulate different RFID scenarios that apply different settings, protocols, and schemes. As a common cryptographic primitive, we have used a standardized 192-bit NIST elliptic curve (National Institute of Standards and Technology (NIST), 2000) defined over the finite field $GF(p)$. The main operation of the primitive is a scalar multiplication that adds k copies of a certain elliptic-curve point P together. For achieving this operation, we applied the Montgomery method using standard-projective coordinates. For the evaluation of different interactive identification schemes, we considered only the one round $l = 1$ (parallel) version. The size of the challenge is defined by the security parameter t. It has been chosen to be 48 bits. The probability of cheating the verifier is therefore $l/2^t = 1/2^{48}$.

For the signature schemes, we applied the SHA-1 hash function and used the same security parameter as used for the identification schemes. Furthermore, we simulated a simple public-key infrastructure where the public-key of an RFID tag is signed by a trusted third party. This certificate is stored in the memory of the tag and is transmitted during the tag authentication process.

6.1 Certificate Management

We considered three different scenarios for certificate management in an RFID application. All scenarios are based on the International Telecommunication Union (ITU-T) standard X.509 for Public-Key Infrastructure (PKI). Only version 1 certificates have been created to restrict the size of the certificates that will be stored in the non-volatile memory of RFID tags. In the first scenario, a standard X.509 certificate is generated. The structure of this certificate is shown in Figure 7. It contains attributes such as the version number, serial number of the certificate, issuer and subject identifier, validity, public-key data, and the signature of the certificate. The certificate is encoded using the

```
Certificate
  Version: 1
  Serial Number: 4660
  Signature Algorithm: ecdsaWithSHA1 (1.2.840.10045.4.1)
  Issuer: CN=TestCA
  Valid not before: Thu Feb 12 18:08:14 CET 2009
        not after: Tue Feb 12 18:08:14 CET 2019
  Subject: CN=14443A00,EMAIL=test@test.com
  SubjectPublicKeyInfo:
    Algorithm: ecPublicKey, NISTp192 (1.2.840.10045.2.1)
    SubjectPublicKey:
      03:32:00:04:62:B1:2D:60:69:0C:DC:F3:30:BA:BA:B6:
      E6:97:63:B4:71:F9:94:DD:70:2D:16:A5:63:BF:5E:C0:
      80:69:70:5F:FF:F6:5E:5C:A5:C0:D6:97:16:DF:CB:34:
      74:37:39:02
SignatureAlgorithm: ecdsa-with-SHA1 (1.2.840.10045.4.1)
Signature:
  30:35:02:18:1F:91:F5:89:8B:4F:C5:D3:47:D8:7C:F2:5D:8F:
  AE:53:6F:F7:39:3E:B2:D3:18:92:02:19:00:B4:F5:9A:F7:3B:
  13:80:48:B3:86:82:42:62:C8:23:57:7A:C5:A9:A6:B5:96:C2:
  D9
```

Figure 7: Structure of a generated standard X.509 certificate.

Distinguished Encoding Rules (DER) which results in a binary representation of the given Abstract Syntax Notation One (ASN.1).

The second scenario takes compressed certificates, which only store the x coordinate of the public key since the y coordinate can be easily reconstructed by solving the curve equation. This technique is known under point compression and is claimed by U.S. patent 6141420.

In the third scenario, we only stored the variable part of the certificate, which are the public key, the serial number of the certificate, and the signature. The rest of the certificate can be reconstructed by a reader which adds the remaining constant part of the certificate to obtain a valid X.509 certificate. The last scenario might be preferably used in RFID applications where less memory is an urgent requirement.

7 Results

In the following, we describe a protocol-performance metric of all described authentication protocols of Section 4. The protocols have been evaluated based on different criterias such as their security assumptions, storage requirements, computational effort, and communication overhead. First, we present results of analyzed authentication protocols that are based on identification schemes. Second, we focus on results obtained from the evaluation of authentication protocols which are based on signature schemes.

7.1 Authentication Protocols based on Identification Schemes

The performance metric of the described authentication protocols is given in Table 7.1. All protocols provide an interactive proof-of-knowledge and are secure against impersonation under passive attacks. ECOkamoto also provides provably security against active and concurrent attacks. The protocols are not secure against reset attacks as shown by Cannetti et al. (Canetti et al., 2000) but can be protected as shown by Bellare et al. (Bellare et al., 2001).

Table 1: Authentication Protocols based on Identification Schemes

	Schnorr	**Okamoto**	**GPS**
Zero knowledge	honest-verifier	witness-indist.	honest-verifier
Crypt. Service			
Entity auth.	Yes	Yes	Yes
Message auth.	No	No	No
Security against			
Passive attacks	Yes	Yes	Yes
Active attacks	No	Yes	No
Concurrent attacks	No	Yes	No
Reset attacks	No	No	No
Memory [byte]			
Private key	24	48	24
(*Certificate 1*	*268*	*292*	*268*)
(*Certificate 2*	*243*	*267*	*243*)
Certificate 3	76	100	76
Total	**100**	**148**	**100**
Computation			
Size of scalar	24	48	40
#Additions	771	1,544	1,283
#Subtractions	769	1,536	1,281
#Multiplications	3,271	6,542	5,447
#Squarings	962	1,924	1,602
#Inversions	2	4	2
#Hash comp.	0	0	0
Total Operations	**5,775**	**11,550**	**9,615**
Estim. Cycle Count	**993,432**	**1,986,864**	**1,630,872**

The memory requirement of the protocols has been characterized by considering the storage of the private key and the certificate of the public key. The private key needs 24 bytes for all protocols except for ECOkamoto, which needs 48 bytes for one additional private key. The size of the certificate depends on the scenario described in the previous section. For a standard X.509 certificate, as given in Scenario 1, we obtained a size of 268 bytes for ECSchnorr and ECGPS, and 292 bytes for ECOkamoto due to the storing of an additional public key. (see Figure 7 for the structure and content of the certificate). For the second sce-

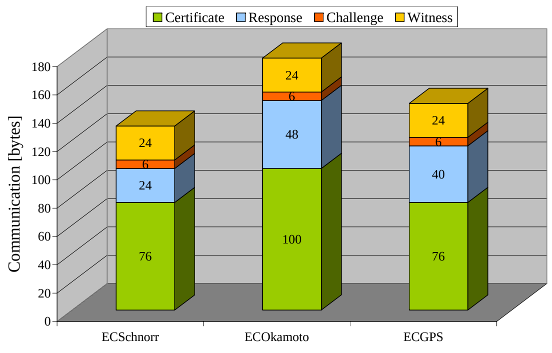

Figure 8: Communication overhead of evaluated identification schemes.

nario, we applied a point compression technique and yielded a certificate size of 243 bytes and 267 bytes, respectively. In the third scenario, only the variable part of the certificate is stored that is the public-key (24 bytes for the x-coordinate of one public key), the signature (48 bytes), and the serial number (4 bytes). Thus, we obtained a certificate size of 76 bytes and 100 bytes, respectively. By taking the third scenario as a reference, we got a total memory usage of 100 bytes for both ECSchnorr and ECGPS and 148 bytes for the ECOkamoto protocol.

The computational complexity of the protocols has been evaluated as follows. First, we have determined the elemental number of finite-field operations by counting the appropriate simulation-method invocations. ECSchnorr needs 5,775 operations, ECOkamoto needs 11,550 operations, and ECGPS needs 9,615 operations. Second, we empirically weighted the operations due to their computational complexity. We estimated the cycle counts for the finite-field operations by assuming 45 cycle counts for modular addition and subtraction, 220 cycle counts for modular multiplication, 176 (=220*0.8) cycle counts for squaring, and 17,600 (=220*80) for modular inversion. By summing up the weighted values, we obtained a rough estimation for the overall computational efficiency of the analyzed protocols. The results are presented in the bottom of Table 7.1. It shows that in our experiment ECSchnorr is the most efficient protocol resulting in a total cycle count of 993,432 followed by ECGPS with 1,630,872 and ECOkamoto with 1,986,864.

The evaluation result of the protocol-communication overhead is shown in Figure 8. It shows that the ECSchnorr protocol needs the smallest amount of communication bytes which is 130 bytes. ECGPS needs 146 bytes because of the increased size of the response. As opposed to ECSchnorr, no modulo reduction of the response value is performed in ECGPS. However, this allows faster response calculations but needs more bytes to be transferred over the air interface. The highest obtained value is due to ECOkamoto which needs an extra amount for the response and also for the certificate. The total number of bytes for ECOkamoto is 178 bytes.

Although a comparison of the described protocols becomes largely fair at some level, it has to be noted that there exist many variants and improvements for their implementations in practice. One technique that we have used in our evaluation is to reduce the memory usage by only storing the x-coordinate of the public keys (point compression). Another common technique is to hash the witness to reduce the communication overhead which becomes rather interesting in scenarios where the commitments are pre-computed to be used as *coupons*. The challenge can also be chosen with a low-hamming weight so that the final multiplication in the response calculation can be omitted. This offers speed advantages for scenarios that allow ”on-the-fly” authentication. However, in order to compare different protocols, it is necessary to provide common parameters and setting for all protocols to become a fair evaluation in general. Thus, it is convenient to deal with relative values rather than absolute ones.

7.2 Authentication Protocols based on Signature Schemes

Next, we analyze authentication protocols which are based on different signature schemes. Therefore, we transformed the ECSchnorr, ECOkamoto, and ECGPS identification schemes into signature schemes using the technique proposed by Fiat-Shamir (Fiat and Shamir, 1987). All schemes are then used in a challenge-response protocol according to ISO/IEC 9798-3 (International Organisation for Standardization (ISO), 1993).

The performance metric of the resulting protocols is shown in Table 7.2. All protocols provide security against passive and active attacks and provide security against existential-forgery attacks.

The memory usage of the signature schemes is the same as for the identification schemes. We considered the storing of the private key and the certificate holding the public key and the signature. Also the computational costs are very similar to that of the evaluated identification schemes. The number of finite-field operations keeps constant but an additional hash calculation is needed to compute the signature. For all schemes, the input size of the hash functions is 30

Table 2: Authentication Protocols based on Signature Schemes

	Schnorr	**Okamoto**	**GPS**
Crypt. Service			
Entity auth.	Yes	Yes	Yes
Message auth.	Yes	Yes	Yes
Security against			
Passive attacks	Yes	Yes	Yes
Active attacks	Yes	Yes	Yes
existent. forgery	Yes	Yes	Yes
Memory [byte]	**100**	**148**	**100**
Computation [byte]			
Hash-input size	30	30	30
#FF Operations	5,775	11,550	9,615
#Hash comp.	1	1	1
Total Operations	**5,793**	**11,568**	**9,633**
Estim. Cycle Count	**997,392**	**1,990,824**	**1,634,832**

bytes. We estimated the costs for that hash calculation by assuming 4,000 clock cycles for the computation. This corresponds to approximately 18 additional finite-field multiplications. The overall complexity of the evaluated signature schemes are therefore as follows. ECSchnorr results in 5,793 finite-field operations with a cycle count of 997,392 followed by ECGPS with 9,633 operations and 1,634,832 cycles, and ECOkamoto with 11,568 operations and a cycle count of 1,990,824.

The communication of the described protocols is composed of the challenge (message), which is sent by the verifier, and the response (signature and certificate) that is sent by the prover. The results of our investigations are shown in Figure 9. ECSchnorr provides the smallest amount of communication bytes which is 126 bytes. ECGPS follows with 146 bytes and ECOkamoto needs to transmit 222 bytes in our experiment.

8 Conclusion

In this article, we presented a top-down evaluation survey of different public-key techniques to achieve RFID-tag authentication. First, we described a general model to characterize different authentication protocols and schemes. Second, we provided practical results of evaluations for elliptic-curve based variants of schemes that have been especially designed for low-resource devices. We analyzed both identification schemes and signature schemes of ECSchnorr, ECOkamoto, and ECGPS. Our evaluation has led us to the result that elliptic-curve based identification schemes and signature schemes have nearly

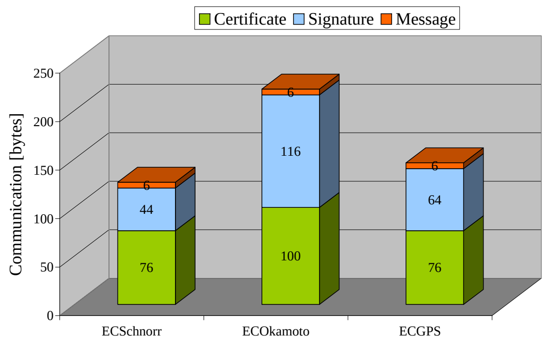

Figure 9: Communication overhead of evaluated signature schemes.

the same complexity in terms of memory usage, computational complexity, and communication bandwidth. For both types of schemes, the most resource-consuming operation is the elliptic-curve point multiplication. The hash computation performed by the signature schemes does not has a large impact on the overall performance, even though that signature schemes provide additional cryptographic services such as non-repudiation, data integrity, and a transferable proof-of-origin. In addition to that, our experiments have shown that ECSchnorr is more efficient in both computational effort and communication bandwidth than ECOkamoto and ECGPS. This is the case when the scalar multiplication is performed online by the tag during the authentication process. If the scalar multiplication is performed in an offline way and the tag uses pre-computed coupons, for example, ECGPS provides enhanced performance due to the lack of the modulo reduction during response calculation. The performance of ECOkamoto, in contrast, is higher than ECSchnorr and ECGPS due to the need of storing, computation, and communication transfer of an additional public-key pair. We conclude that each protocol provides different tradeoffs among the cryptographic service, storage, computational effort, and communication bandwidth. ECOkamoto provides additional security features while ECGPS has been designed for efficient ”on-the-fly” authentication. ECSchnorr might be preferred if the primitive computation is done online by the tag itself.

Acknowledgment

The author would like to thank Manfred Aigner, Johannes Wolkerstorfer, and Martin Feldhofer for their valuable inputs and helpful discussions. The research described in this paper has been supported by the European Commission funded project *Collaboration at Rural* under grant number 034921 (Project *C@R*) and the Austrian government funded project *CRYPTA* established under the *Trust in IT-Systems* program FIT-IT.

REFERENCES

Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., and Verbauwhede, I. (2006). Public-Key Cryptography for RFID-Tags. In *Workshop on RFID Security 2006 (RFIDSec06), July 12-14, Graz, Austria*.

Bellare, M., Fischlin, M., Goldwasser, S., and Micali, S. (2001). Identification Protocols Secure Against Reset Attacks. In *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, pages 495–511, London, UK. Springer-Verlag.

Canetti, R., Goldreich, O., Goldwasser, S., and Micali, S. (2000). Resettable zero-knowledge. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 235–244, New York, NY, USA. ACM.

Diffie, W. and Hellman, M. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654.

ElGamal, T. (1984). A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Advances in Cryptology - CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer.

Fiat, A. and Shamir, A. (1987). How to prove yourself: Practical solutions to identification and signature problems. In *In Advances in Cryptology - Crypto 86*, volume 263, pages 186–194. Lecture Notes in Computer Science, Springer.

Gaubatz, G., Kaps, J.-P., Ozturk, E., and Sunar, B. (2005). State of the art in ultra-low power public key cryptography for wireless sensor networks. *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 146–150.

Girault, M., Juniot, L., and Robshaw, M. (2007). The feasibility of On-the-Tag Public Key Cryptography. In Munilla, J., Peinado, A., and Rijmen, V., editors, *Workshop on RFID Security 2007 (RFIDSec07), July 11-13, Malaga, Spain*, pages 77–86.

Girault, M., Poupard, G., and Stern, J. (2006). On the fly authentication and signature schemes based on groups of unknown order. *Journal of Cryptology*, 19:463–487.

International Organisation for Standardization (ISO) (1993). Information Technology - Security Techniques - Entity authentication mechanisms - Part 3: Entity authentication using a public key algorithm.

International Organisation for Standardization (ISO) (2004). ISO/IEC 9798 Part 5: Information technology – Security techniques – Entity authentication – Mechanisms using zero knowledge techniques.

Kumar, S. S. and Paar, C. (2006). Are standards compliant Elliptic Curve Cryptosystems feasible on RFID? In *Workshop on RFID Security 2006 (RFIDSec06), July 12-14, Graz, Austria*.

Law, L., Menezes, A., Qu, M., Solinas, J., and Vanstone, S. (2003). An efficient protocol for authenticated key agreement. volume 28, pages 119–134, Norwell, MA, USA. Designs, Codes and Cryptography, Kluwer Academic Publishers.

McLoone, M. and Robshaw, M. J. B. (2007). Public Key Cryptography and RFID Tags. In Abe, M., 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.

Menezes, A. J., van Oorschot, P. C., and Vanstone, S. A. (1997). *Handbook of Applied Cryptography*. Series on Discrete Mathematics and its Applications. CRC Press. ISBN 0-8493-8523-7, Available online at http://www.cacr.math.uwaterloo.ca/hac/.

National Institute of Standards and Technology (NIST) (2000). FIPS-186-2: Digital Signature Standard (DSS). Available online at http://www.itl.nist.gov/fipspubs/.

Okamoto, T. (1993). Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In Brickell, E. F., editor, *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer.

Preneel, B. et al. (2003). NESSIE Security Report, D20. Available online at http://www.nessie.eu.org.

Schnorr, C.-P. (1990). Efficient Identification and Signatures for Smart Cards. In Brassard, G., 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.

Schroeppel, R., Beaver, C., Gonzales, R., Miller, R., and Draelos, T. (2003). A Low-Power Design for an Elliptic Curve Digital Signature Chip. In Jr., B. S. K., Çetin Kaya Koç, and Paar, C., editors, *Cryptographic Hardware and Embedded Systems – CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 366–380. Springer.

Wolkerstorfer, J. (2005). Is Elliptic-Curve Cryptography Suitable for Small Devices? In *Workshop on RFID and Lightweight Crypto, July 13-15, 2005, Graz, Austria*, pages 78–91.