On Digital Signatures in Ad Hoc Networks

D. Westhoff, *Member, IEEE* B. Lamparter, C. Paar, A. Weimerskirch

***Abstract*—Cryptographic primitives need to be carefully evaluated when being applied in components ensuring enhanced protection aims. This work analyses the performance of RSA and ECDSA based digital signature schemes in the context of multi-hop ad hoc networks. Our work shows that, contrary to a single wireless hop scenario with restricted clients in the wireless network and a powerful server in the fixed network, the choice of an appropriate signature candidate is not as obvious as for the aforementioned architecture. Our performance analysis considers three, partially interdependent axes, a) the required security level, b) the device restriction, and c) the protocol specification with the required security relationships and the assumed traffic flow. We take c) into account by analysing two charging protocols for wireless multi-hop scenarios. By generalizing our observations we carefully derive some recommendations to strengthen the security engineering process for dependable distributed systems over the wireless network.**

***Index Terms*—public key systems, security in ad hoc networks, digital signature, ECDSA, RSA.**

I. Introduction

Performance comparisons of asymmetric cryptographic schemes such as RSA [29] and ECDSA (*elliptic curve digital signature algorithm*) [14] are of considerable importance for the security engineering community. This is in particular true since technologies that are based on elliptic curves have reached the standardization bodies [6]. At the same time more and more powerful mobile handhelds (PDA, mobile phone) become available. Although their performance tends to be weak and restricted, even relatively expensive schemes like digital signatures are nowadays conceivable.

Arguing from that perspective the usage of digital signatures even in wireless multi-hop networks consisting of devices with restricted bandwidth, CPU power and storage capacity therefore is not unrealistic anymore. Of particular interest is an integration into protocols aiming at secure routing [26], [39] a technical support for accounting and charging [1], [16], [40], security aspects for peer-to-peer services, or giving incentives for cooperation [1], [2], [16], [25], [40]. Protocols from these classes can be characterized as follows:

- in its general case such protocols require the explicit involvement of more than two entities,
- the mandatory security level as well as the requirements to its durability are rather relaxed,
- each entity may be in different roles at different moments.

Although the currently available performance analyses [6], [18], [31], [37] have built up a proper understanding of a comparable security level of RSA and ECDSA, a protocol and application-related analysis [10], [37] is currently under-represented in the literature. However, we feel that a performance analysis of digital signatures, considering also the concrete protocol and its applications, adds additional value to the deployment of advanced security protocols with frequently changing entity roles. The latter is a conspicuous characteristic in ad hoc networks. Entities that may at one point of time be in the role of a client, whereas in the next moment they may be in the role of a service or network-provider, must be prepared for situations with sometimes contradicting requirements. Furthermore, in the general case, more than two entities are explicitely involved in the protocol. And finally, the considered protocol classes mostly require a rather weak durability of an appropriate security level. Where e.g. the long-term storage of digital documents with linkage to its originator(s) may require a durability of an appropriate security level for up to more than one decade, the required security level of protocols from the considered classes is in the order of several hours but not more than one day. Since these protocols only aim at the protection of information with an extreme short life experience, such a design decision causes no security lack.

One may argue that authentication schemes [27], [34], [35], [36] which are based on a symmetric message authentication code (MAC) are more suitable for the envisioned protocol class since they are in the range of 10^2 to 10^3 less computational intensive than asymmetric schemes. Unfortunately, a symmetric scheme does not support the protection aim *non-repudiation,* which is mandatory in all cases where technical solutions are linked to the jurisdiction

Manuscript received August, 2004. This work was partly funded by the German Federal Ministry of Education and Research BMB+F. The results are part of the project IPonAir – Next Generation Wireless Internet http://www.iponair.de/. A. Weimerskirch was partly funded by BRICS, Basic Research in Computer Science, funded by the Danish National Research Foundation.

B. Lamparter and D. Westhoff are with NEC Europe Ltd., Heidelberg D-69115 Germany ({bernd.lamparter, dirk.westhoff@ccrle.nec.de}).

C. Paar and A. Weimerskirch are with the Communication Security Group at University of Bochum, Bochum D-44780 Germany ({cpaar, weika}@crypto.rub.de).

in real-world. Another advantage of asymmetric schemes is *scalability*. To fully cover all bilateral security relationships of an ad hoc community with N nodes, $N(N-1)/2$ symmetric keys need to be established. By applying an asymmetric scheme, only N public/private key pairs need to be generated. Note that the scalability aspect is of essential importance for the considered protocol classes in ad hoc networks. Since generally more than two entities may be explicitly involved, in the general case also multiple entities need to authenticate an originator and/or a message it has sent. This receiving subset of verifying nodes frequently is unknown in advance and may change with each run of the protocol. Clearly a purely symmetric scheme based solution with a single key between the originator and all receivers does not overcome this problem. Each verifier knowing the key could easily masquerade as the originator of the message. In [4], Canetti et al. propose an '*asymmetric MAC*' scheme which is based on multiple symmetric keys and MACs for n receiving parties. Originally proposed for multicast traffic with a small set of colluding nodes, we feel that this approach does not scale well for protocols in ad hoc networks due to the nodes' mobility and the danger of a large set of compromised nodes resulting in high data and computational overhead.

Closely related to the scalability issue, the use of a public key scheme requires a security infrastructure. We argue that, depending on the ad hoc network type, which may be of a pure kind, or a stub access to the fixed network [15], [19], we either assume a threshold based certification authority [41], a self-organizing public-key management system [5] similar to Phil Zimmermann's PGP, or, in case of temporary connection to the fixed network, a PKI in the backbone.

The organization of this article is as follows. We take up discussion on related work in Section II. Section III compares our own speed optimized ECDSA reference implementation against an available RSA reference implementation. Section IV, as an example describes the impact of the different digital signature schemes on the protocol performance of two protocols for charging in stub ad hoc networks. In Section V, we carefully generalize our results for multi-hop ad hoc networks and present some design criterions for the usage of digital signatures for protocols from the introduced protocol classes. We conclude in Section VI.

II. Related Work

Lenstra and Verheul [18] present a recommendation of key sizes for asymmetric cryptosystems, RSA, and discrete logarithm based cryptosystems both over finite fields and over elliptic curves over prime fields. Based on Moore's law, they incorporate future changes of the available computational power and the arising hardware cost. Their model also takes into account future progress in cryptanalysis. Based on some hypotheses, e.g. that DES has been secure enough for commercial applications until 1982, the authors come up with an equation that predicts the computational load they consider to be infeasible until any concise year in the future. They present for each year until 2050 which minimum key size for which cryptosystem can be considered to be secure until that year. Table I shows a selection of their recommended RSA key sizes.

TABLE I
Lenstra and Verheul's Key Size Recommendation: Security computationally equivalent to that offered by DES in 1982.

Year	…	1986	…	2003	2005	2010	2015	2020	2025	…
key size (RSA)	…	513	…	744	810	990	1191	1416	1664	…

In an RSA Laboratories Technical Note from 1997 [31], Robshaw and Yin analyze cryptosystems based on RSA and on ECDSA. For ECDSA with a key size of 160 bits and 1024-bit RSA, which they found to achieve a comparable level of security, the authors compare the performance of both schemes in terms of storage requirements and computational speed. While its short key size leads to clear advantages for ECDSA with respect to storage requirements, their findings for the computational speed do not allow such a clear statement. According to their benchmarks, the RSA sign operation is about 7 times slower than the one of ECDSA, but the verify operation is more than 6 times faster. Robshaw and Yin fear that elliptic curve cryptography might still offer some yet undiscovered loopholes due to the sophisticated mathematical theory behind it.

Another article [37] by Wiener contains a comparison between 1024-bit RSA and 168-bit ECDSA. The verification times of RSA are found to be more than 30 times faster than those of ECDSA. The signature generation is measured to be around 8 times slower. The author points out that the optimal choice of a signature scheme depends on the particular application. He comes to the conclusion that RSA is well suited, for example, for certificate-based systems that require only few signature generations but thousands of signature verifications. However, in wireless communication scenarios Wiener favors ECDSA as public-key algorithm. The short key size and low signature overhead save transmission bandwidth and lead to smaller silicon implementations.

Performance analyses for specific protocol applications can be found in [10]. Gupta, Gupta and Stebila added ECC support to OpenSSL 0.9.6b and obtained the execution times for ECDSA and RSA signature operations on a PDA with 200Mhz StrongARM CPU and a server with 450MHz UltraSPARC II processor. According to their observations, 1024-bit RSA on the server is about 8 times faster for signature verification, but approximately 5 times slower for signature generation than 163-bit ECDSA. On the PDA, they found verify to be 4 times faster and sign to be 8 times slower. The authors also analyze the performance influence of ECDSA and RSA on the SSL protocol. Their measures are the Handshake Crypto Latency, which is essentially the sum

of the times the client and the server spend doing public key operations, and the Server Crypto Throughput, which is the rate at which the server can perform the cryptographic operations. For a security level of 1024-bit RSA, ECC performed more than five times better in terms of Server Crypto Throughput. However, in terms of Handshake Crypto Latency, the performance depends on the underlying scenario. For a PDA talking to a server, RSA outperforms ECC, while for PDA talking to PDA or server talking to server, ECC is nearly twice as fast as RSA. The authors see a performance advantage for ECC at higher levels of security.

Another performance evaluation for a specific protocol was done by Zhong, Chen and Yang in [40]. For a Laptop at 866 MHz, the authors compare the usage of digital signatures based on ECNR over *GP(p)* 168-bit [11] and digital signatures based on 1024-bit RSA for the protocol 'Sprite'. The protocol supports a credit-system for ad hoc nodes and fits exactly into the protocol class considered in this paper. The authors point out that for minimizing delays in the forwarding phase over several involved intermediate nodes RSA would be a better choice than an elliptic curve based implementation. Using RSA forwarding over eight nodes takes 0.3 ms compared to 13.2 ms with ECNR. Zhong, Chen and Yang argue in the right direction. Nevertheless, we feel that a proper performance analysis of digital signatures for multi-hop ad hoc networks needs to be evaluated in more detail.

III. Reference Implementation

Naturally, the choice of the destination platform has a strong impact on the absolute and even on the relative execution times of RSA- and ECDSA-based digital signature schemes. We thus understand the values below as guiding values.

Our destination platform is a Personal Digital Assistant (PDA) Sharp Zaurus SL-5500G, equipped with the operating system Linux 2.4 and an Intel SA-1110 StrongARM CPU (206MHz). We have chosen as RSA reference implementation the version from OpenSSL (Developer Snapshot 20021202) [24]. Since the ECC candidate from OpenSSL is on average on factor 5 slower than our own speed optimized ECDSA implementation [28] we used the latter for performance comparison. The binaries for the Zaurus were generated with the gcc 2.95.2 compiler from the Embedix Tool Chain Packet.

We have implemented ECDSA for general elliptic curves over the binary field $GF(2^m)$ as well as for Koblitz curves. We implemented several different point multiplication methods and determined the optimal choice for our application and target platform. Using the *Fixed-base Comb* method [20] for scalar multiplication with a fixed base point of the curve and the *Montgomery* method [22] for scalar multiplication with arbitrary points, a signature generation on a StrongARM CPU clocked at 206 MHz takes 5.7 ms and a signature verification takes 17.9 ms for 163-bit ECDSA. In case of Koblitz curves, we use the window TNAF method [12], which enables us to verify a signature on a 163-bit Koblitz curve in 11.7 ms. This is a significant speed up compared to the current ECDSA implementation of the OpenSSL project. The RSA keys and signatures were generated by using the OpenSSL crypto library. For signature generation, the same public exponent $e=2^{16}+1$ was used like in the genrsa tool from OpenSSL.

A. Execution Times

It is our concern to compare the execution times of both signature schemes, which use different mathematical problems to achieve security. The resulting security of the scheme RSA is based on the *factorization problem for large numbers,* whereas the ECC-based schemes are based on the *elliptic curve discrete logarithm problem.* Thus, it is decisive to agree which order of an elliptic curve is comparable with which RSA modulus. We consider both as key size and refer to their corresponding security levels to those proposed by the *standards for efficient cryptography group* (*SECG*) [6]. For example, their weakest proposed security level is a 113-bit ECDSA-based key and its comparable 512-bit RSA-based key. For the highest proposed security level, SECG considers a 233-bit ECDSA-based key size to be correspondent to a 2240-bit RSA-based one.

TABLE II
Execution Times for Signature Operations Based on ECDSA and RSA on a Sharp Zaurus SL-5500G.

Security level [bit]		Time for signature generation [ms]			Time for signature verification [ms]		
ECC	RSA	ECC	RSA	Ratio	ECC	RSA	Ratio
113	512	2.8	13.7	4.9	7.5	1.3	5.7
131	704	3.8	32.4	8.5	11.5	2.5	4.6
163	1024	5.7	78.0	13.6	17.9	4.3	4.1
193	1536	7.6	251.9	33.0	26.0	9.7	2.6
233	2240	10.1	731.8	72.0	37.3	20.4	1.8

Table II documents the execution times of our platform. The ECC based execution times refer to our ECDSA implementation for general elliptic curves. We separate between signature generation and signature verification. Like stated in previous work, an ECDSA-based signature generation is much faster than the corresponding RSA operation. Vice versa, the RSA-based signature verification clearly beats the equivalent ECDSA-based verification. For both cases, the relative gain depends on the targeted security level. A 113-bit ECDSA-based signature generation is 4.9 times faster than its RSA competitor. But this ratio increases with a rising security level to a factor of 72 for a 233-bit security level. In contrast, a 113-bit ECDSA signature verification is 5.7 slower than its comparable RSA-based operation. This ratio shrinks to a factor of 1.8 for an increasing ECDSA-security level of up to a 233 bit. Finally note, that the time complexity for both signature schemes for

signature verification and signature generation is $O(|key_{ECC}|^3)$, whereas $|key_{ECC}|$ is the size of an ECDSA key. Nevertheless, due to the limited value space of the listed key sizes the impact of constants cannot be neglected.

A hardware realization leads to a dramatic improvement of the presented execution times. However, the current size of such hardware components and their production cost are too high to be a standard part of PDAs or mobile phones.

B. Data Overhead

The additional required storage space and transport volume of the competing signature schemes result from their certificate- and signature lengths. Again, we cannot properly

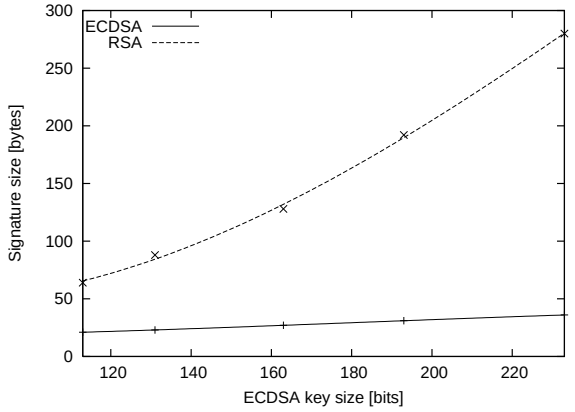

Fig.1. Comparison of RSA- vs. ECDSA-based signature lengths.

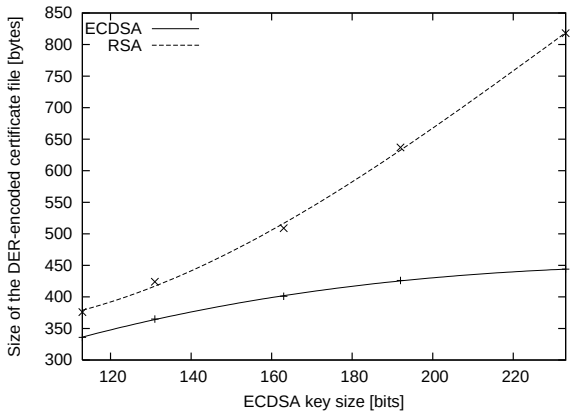

Fig.2. Comparison of RSA- vs. ECDSA-based certificate lengths.

analyze this overhead without taking the targeted security level into account. In Figs. 1 and 2 we illustrate this interrelation.

Obviously, both, the signature- and the certificate length are more favorable, since smaller, for an ECDSA based signature scheme. This suits for all the considered security levels. With an increasing targeted security level this advantage compared to RSA-based signature schemes dramatically increases to a ratio of up to 1/3 for the certificate and 1/6 for the signature. In this connection the certificate length tends to be more important with respect to some storage restriction on the mobile device. In contrast, the length of the signature for both signature schemes has more impact on the net throughput over the wireless link. Compared to the relatively rare broadcast of a certificate, the latter results from a by far more frequent transmission of digital signatures.

Not considering some additional coding information, which are about 6 byte, a RSA-based digital signature is of length $|key_{RSA}|$, whereas an ECC-based signature is of length $2 \cdot |key_{ECC}|$. By applying the concept of *point compression* [13], we can even reduce the latter to the size $|key_{ECC}|+1$. This means that a 1024-bit RSA-based digital signature is about 128 byte, or, including coding information 134 byte. Its corresponding 163-bit ECDSA representative needs 21 byte resp. 27 byte with coding information.

After these preliminary considerations we are prepared for a protocol and application sensitive performance analysis of digital signatures in wireless multi-hop ad hoc networks.

IV. Case Study: Sprite and SCP

Two protocols are known from the literature, which use digital signatures to support charging and billing in multi-hop ad hoc networks. Both, namely the *Simple Cheat-Proof protocol* (*Sprite*) [40], and the *Secured Charging Protocol* (*SCP*) [16] present technical solutions for a similar business model: Each node that forwards foreign data is positively charged and gets some monetary reward, whereas the sender, and for the latter case also the final destination, are negatively charged by an ISP responsible for the (temporary) connection of the ad hoc cloud to the fixed network. Note that in cases where the sender and the final destination belong to the same ad hoc cloud, in particular a decentral traffic forwarding remains possible. More precisely, the traffic does not need to be diverted over the access point to the fixed network. The aforementioned business model becomes only reasonable with its unforgeable technical realization[1] since only then the participators of the ad hoc community indeed get incentives to co-operate. In particular, the forwarding of data over power restricted devices with potentially selfish but battery saving user behavior seems to be more reasonable in the presence of protocols like *Sprite* or *SCP*.

A third charging and rewarding protocol for packet forwarding has been proposed in [1] by Salem et al. They propose forwarding all traffic in a multi-hop manner over an access point to the fixed network, even in cases when the sender and the receiver belong to the same ad hoc cloud. Due to this restriction for traffic forwarding, for this approach the scalability requirements for the security associations relax. Only the access point (and not each mobile node) needs to establish security associations with each mobile node. As a result, they choose a symmetric MAC scheme for the authentication.

In contrast, due to the support of a decentral routing policy, *Sprite* and *SCP* use digital signatures to still be scalable.

[1] Sprite and SCP have been formally validated either by proofs of theorems or by BAN-Logic.

Here, the coverage of all bilateral security relationships of nodes from the ad hoc community is mandatory. We do not present the detailed protocol specification of both protocols and thus refer to [40] and [16]. Instead, we just list the expensive operations, which for both protocols are

1) *signature generations* per packet or per bundle of packets at the sending mobile node, and their
2) *signature verifications* at each forwarding intermediate node.

The signature generations at the sending mobile node (*S)* serve to recognize the originator of a data packet at intermediate nodes (N_i ,*i=1,…,n)* on the routing path. In a fully open system with mobile nodes from various administrative domains, it is impossible for the verifying intermediate nodes to doubtlessly infer to the identity of the originator. Instead, intermediate nodes check by verifying the received signatures that the originator of the packets has properly registered at the ISP. Since intermediate nodes only get rewards from the ISP for forwarding traffic from a legitimated sending mobile node, it is in each intermediate node's interest to verify from whom the data stem. Thus, we need signature verifications at the intermediate nodes. A keyed hash chain generated by all involved intermediate nodes, and, in the case of *SCP*, a confirmation from the final destination *D* to the last forwarding intermediate node N_n about the received amount of data are additional schemes that make the ISP re-engineer the involved parties and the amount of traffic received at *D*. Since a keyed hash chain is a construct that is based on a nested appliance of multiple hash-based MACs it is a very fast cryptographic construct that needs not to be considered in the following performance analysis. Instead, we restrict ourselves just taking into account the necessary digital signature operations. Nevertheless, it needs to be stated that, with the usage of the keyed hash chain, no additional scalability problem arises, since all used symmetric keys have only to be agreed between the access point and each mobile node.

A. Required Level of Security

A typical application scenario for the described charging and rewarding protocols is the provision of wireless multi-hop Internet access at public places, for instance a shopping mall, an airport terminal or a railway station. Such places have in common that the users do not stay longer than several hours at that place. Consequently, most user certificates, private and public keys only need to be valid for a short time, e.g. 24 hours. In case of a longer stay, one could simply enforce a certificate renew every 24 hours to provide fresh certificates.

The objective of someone who attacks such a secure charging protocol is certainly to get uncharged network access or to get rewards without forwarding foreign data. However, the amount of money involved can be expected to be rather low. Today's (June 2003) prices for wireless network range from 3.75€ per hour at the Munich airport to 9€ per hour at the CeBIT. Thus, the financial harm of successfully breaking one pair of keys is less than 200€. Consequently, one can expect that an attacker will not spend a large amount of money to break a key. Note that breaking a key of the proposed charging protocols can only be used to compromise the node that owns the key. No secret information will be revealed; the attacker may forge only the secure charging protocol signatures of the compromised node.

Using the results of DES Challenge III [32] as a basis, one can determine the key size for our purposes according to the RSA key size recommendations of Lenstra and Verheul [18]. In the DES Challenge III launched in January 1999 by RSA Security, Inc., a message encrypted with 56-bit DES algorithm has been broken in 22 hours and 15 minutes. As motivated above, keys that can be broken in approximately this amount of time still provide enough security for our purposes. Lenstra and Verheul offer equations to adapt their recommendations to this level of security. However, we feel that their recommendations are too careful. So far, the largest RSA Challenge Number that has been factored in RSA Security's Factoring Challenge [31] is a 512-bit number. The factoring was finished in August 1999, took 3.7 month and the CPU-effort is estimated to approximately 8000 MIPS years.

TABLE III

VARYING KEY SIZES WITH ESTIMATED YEAR UNTIL SUFFICIENT SECURITY FOR SPRITE OR SCP IS PROVIDED.

Year	RSA key size	ECDSA key size
1999	512	113
2006	704	131
2015	1024	163
2026	1536	193
2039	2240	233

We feel that this margin of security is still sufficient for our particular applications. Lenstra and Verheul recommend a minimum key size of 513 bits for RSA in the year 1986, so we obtained the values in Table III by taking their recommendations for the year 1986+x as guide value for our implementation in the year 1999+x. To our opinion this is still a correct interpretation, simply with a different margin of security. Note, that we do not claim that breaking cryptosystems that use the key sizes in Table III is infeasible. However, with respect to the considered protocol classes, breaking such cryptosystems in less than 24 hours requires an amount of computational resources that is disproportionate to the financial gain an attacker might get.

Now that we have a guide for RSA key sizes, we can again use the SECG standard document [5] to derive corresponding ECDSA key sizes. Table III contains different key sizes listed in the standard document and the year until which they prospectively provide sufficient security for the digital signature within our considered protocol class. These key

sizes are based on the assumptions made in [18]. In particular, we cannot fully anticipate the effects of future progress in cryptanalysis.

In November 2002, Certicom announced that the ECCp-109 challenge has been solved using a large network of 10,000 computers within 549 days [7]. This cryptosystem is considerably weaker than an elliptic curve cryptosystem over a 113-bit binary field. Hence, the level of security that the key sizes in Table III provide is probably still greater than necessary for the envisioned applications.

B. Execution Times of Sprite and SCP

Whenever involved in a communication process, each mobile device of the ad hoc cloud may be either in the role of the sending mobile node, the final destination, or an intermediate node. It may even be the case that a mobile device is in more than one role simultaneously, e.g. it may be the sending node for one communication process and the

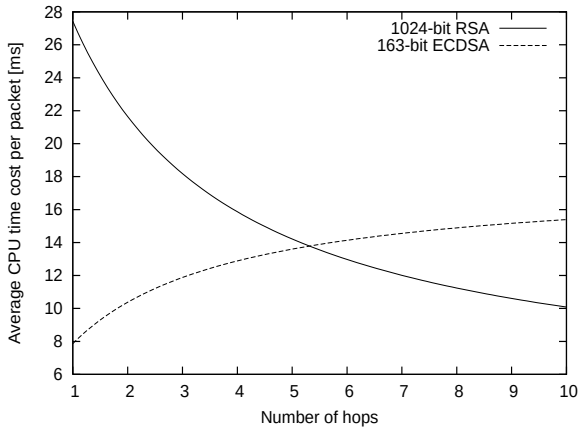

Fig.3. Impact of the network topology on the protocol performance with the usage of RSA- and ECC-based digital signatures.

forwarding intermediate node for another. For an optimal solution related to the execution times on each mobile device, we need to minimize the total time of execution times for the different operations according to their probabilities of appearance. Eq. 1 serves as a metric to find such a protocol specific optimal solution and execution time related optimal solution. Since we are not interested in the absolute execution times but in relative execution times, for a proper metric that helps deciding which signature candidate is best suited, we can neglect the influence of multiple simultaneous communication processes without the loss of generality:

$$t(X)_{CPU} = P(X=S)t_g + \sum_{i=1}^{n} P(X=N_i)t_v + P(X=D)t_v \quad (1)$$

On average, the probability $P(.)$ at which a device X is either in role of S, N_i $(i=1,\ldots,n)$, or in role of D is

$$P(X=S)=P(X=N_i)=P(X=D)=1/(n+2), \quad (2)$$

where n is the number of involved intermediate nodes of a single communication process. We denote the execution times

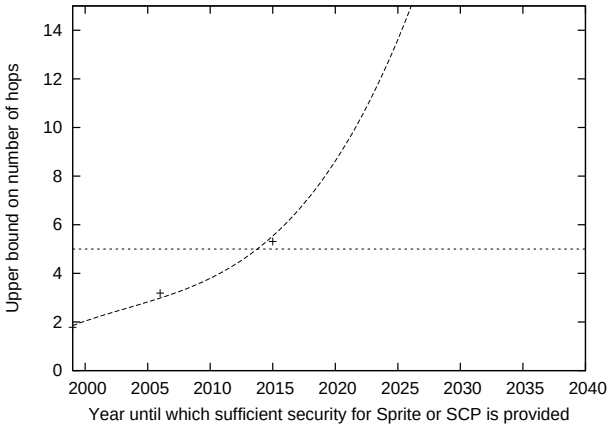

Fig.4. Choice of a digital signature with respect to the average route length and the striven security durability.

for signature generation and signature verification with t_g res. t_v. Using the probabilities of Eq. 2, Eq. 1 results in

$$t(X)_{CPU} = (t_g+(n+1)t_v) / (n+2). \quad (3)$$

Obviously, besides the execution times for signatures also the assumed average route length (indicated by n) is another parameter that has impact on the resulting total execution time $t_{CPU}(X)$ on a mobile device. We illustrate this relation in Fig. 3 for various average route lengths by using the execution times of our reference implementations. For illustration we have chosen a 163-bit ECDSA security level with *5.7ms* (ECDSA) versus *78.0ms* (RSA) for a signature generation and *17.9ms* (ECDSA) versus *4.3ms* (RSA) for a signature verification. With such a choice we expect the charging protocols to be secure at minimum for the duration of one day until the year 2015.

Fig. 3 documents that for network topologies with a relatively small average number of required intermediate nodes, the ECC-based digital signature is less CPU wasting than its comparable RSA competitor. However, if more than five intermediate nodes need to be involved, this statement changes in favor of an RSA-based solution. In such cases, the increasing probability of a signature verification becomes the contributing factor on the expected CPU load of the mobile device.

Apparently this trade-off of up to five involved intermediate nodes will change with a varying striven security durability. We illustrate this fact in Fig. 4 by again using Eq. 1, but this time with varying t_g and t_v for the execution times in Table 1. The area below the curve indicates situations, where ECDSA is the better choice for protocols like *Sprite* or *SCP* according to our metric. Vice versa, for a network topology with an average route length of six intermediate nodes and a more relaxed security level of up to 163-bit ECDSA, a RSA- based signature turns out to be the better choice. Nevertheless, for a higher security level with a security durability beyond the year 2015, one should prefer an ECC-based solution. This is in particular true since analytical

work on the general value of co-operation based approaches has shown that most impact on the overall end-to-end throughput can be achieved for ad hoc networks with small average route length below six intermediate nodes [17]. Finally, the intersection of the curve from Fig. 4 with the horizontal even refers to the security level chosen in Fig. 3.

C. Data Overhead of Sprite and SCP

Considering the time horizon at which civilian multi-hop ad hoc networks are envisioned to become reality, we expect that the considered protocol classes become relevant earliest in 5-8 years. Assuming Moore's Law the hardware will improve in speed by a factor of around 16 by that time. Further, one can argue that during this time scale suitable scaled hardware-based digital signature components for minimal prices will be available, but we believe that it will only be built into expensive high-end products. This means that by the time the execution times for protocols like *Sprite* and *SCP* will become close to be able to sign and verify each packet over a moderate number of restricted devices similar to the considered PDAs. This statement even holds for time critical traffic like *Voice over IP (VoIP)* [9] with a proposed end-to-end delay of not more than *50-70 ms*. Obviously, there still will be a need for reducing the execution time of message authentication to save battery power and to leave CPU power to other applications such as multi-media. In contrast, it turns out that the length of a digital signature is the bottleneck. Even if we use a relatively relaxed security level of 163-bit ECDSA, which is still acceptable for protocols like *Sprite* or *SCP* for remaining 5-7 years until the year 2015, this means that the digital signature length is 21 byte including coding information (compared to 70 byte when using RSA). Note that even for applications with both, a very relaxed required security level and a short security durability a further reduction of the key size is considered to be unacceptable. In particular, a 131-bit key which we expect to be secure until the year 2006 would not be appropriate for protocols we envision to become reality not earlier than in the year 2008.

The most accurate way of ensuring data origin authentication at various nodes is to digitally sign each packet. However, this might exhaust computational resources and add an extreme data overhead to each packet. There are two possible ways to reduce computational complexity and to shorten the data overhead by not loosing the aforementioned scalability advantage of public-key schemes. The first way improves the digital signature algorithm whereas the second one secures more data, i.e. more packets, with only one signature. Since we focused on standardized signature schemes here as trusted basis, we only consider the second approach. Candidates are described in [8], [30], [38] for the purpose of *multicast-stream authentication based on digital signatures*. It is clear that a scheme to authenticate multicast traffic can directly be applied to authenticate data packets in a serial fashion, since the latter case is a special kind of a multicast-stream. There are two approaches which both use symmetric methods to extend an asymmetric signature. The first one [8], [38] integrates hash values from several packets into one that is protected by a digital signature. Hence to protect l packets only one digital signature has to be generated by the sender and verified by the receiver and all intermediate nodes. To make this approach robust against burst packet losses, redundant hash information are spread over a sequence of packets. Even in case of packet loss over the wireless channel, the verifying node thus has enough information to start the verify operation. Hash values that require 10-20 bytes are attached to the packets, and not more than two hash values are attached to each packet, such that there is only little data overhead in each packet. Unfortunately, this approach does not take the jitter into account. For real-time conform traffic like *VoIP* it is mandatory to guarantee an almost continuous flow from the source to the destination. Since the schemes have to buffer a bundle of packets either at the source or at the destination, this unacceptably affects the jitter from application layer to application layer from the source to the destination. The second approach [30] prevents jitter and is thus applicable to real-time conversations. It uses the idea of one-time signatures that are only applicable once but perform extremely efficient. This is extended to k-time signatures that allow k signatures and still perform very efficient. Adapted to an ad-hoc network, the idea is to sign a k-time public key with the ordinary certified key. The receiver and all intermediate nodes can then check if this public-key was signed by a certified key. Now the sender uses his k-time secret key to sign the next k packets. Finally another k-time public-key is signed to authenticate the next k packets, and so on. As mentioned above, this scheme does not require buffering of packets such that packets can be signed and verified in real-time. However, the data overhead is immense. For each packet, at least 270 bytes are required for the signature of a 1024-bit RSA security level which is more than twice as much as an RSA signature (and more than 40 times as much as an ECDSA signature). It becomes clear that the first approach is suited to reduce data and computational overhead at the cost of jitter, whereas the second approach reduces the computational overhead only. The smaller the length of each packet, the more the traffic flow equates a data stream. Considering *VoIP,* the expected frame size is 200 bytes, e.g. using a G.711 Codec (rate=64kbit/s). Hence both approaches are not suited to real-time traffic. We feel that more research is needed here, and propose the following until a proper solution is available.

Our solution to make protocols like *Sprite* and *SCP* real-time responsive is much simpler yet more realistic: Depending on the frame size of the traffic type and the application type, the sending mobile node generates signatures for each packet, or only for a minor number of packets. It is clear to us that such a scheme where only a

minor number of packets is signed whereas most of the packets are unauthenticated only works for some application types. It does not work if message authentication is crucial such as a stock market ticker. It is also clear that an attacker can insert or forge faulty messages, or embed his own data for an illegal receiver. However, note that our approach works perfectly fine if the communication between the sender and receiver is encrypted, i.e., protected by a session key. The receiver will then immediately notice modified and inserted messages. Intermediate nodes still might forward forged packets with embedded data of an adversary which is charged at the cost of the sender though. In case of frame sizes near the maximum transport unit (MTU) of e.g. 1500 byte, an attacker could send arbitrary data free of charge. Fortunately, for such a packet length a 21 byte, ECDSA signature has no crucial impact on the net payload rate of the packet. According to such a frame size, the sending node can integrate signatures into each packet without constraining the application's functionality. Note that sender and receiver can intuitively agree on a ratio *l* of signed packets once they have agreed on the relevant frame size. The sender has to sign each *l-th* packet (where *l=1,..,k* depends on the frame size and the data type) and the receiver checks whether a received packet contains a signature or not. It stops the traffic in case of a

Fig.5. SCP demonstrator set-up with four Sharp Zaurus PDAs to measure round trip times, jitter and throughput.

negative signature check or if more than *l+s* of the received packets did not contain a signature, where *s* is a tolerance threshold to take into account lost packets and packets that do not arrive in order. Even in case of loosing a bundle of packets at latest the *l*-th received packet does again contain a signature. Since no buffering of a bundle of packets at source or destination is necessary, no additional jitter at source or destination is caused. Clearly, an attacker may send *l+s*-1 packets without being charged. Thus, to prevent the attacker from sending meaningful traffic or modifying crucial information, we propose *l=1* or *l=2* with *l=2* for traffic like VoIP or video streaming and *l=1* for any file transfer application or data that must not be altered. We believe that for such parameters an attack would be too much effort at little gain.

D. SCP Demonstrator

We set-up an *SCP* demonstrator with four PDAs and WLAN 802.11b to measure the round trip times and the throughput for the forwarding phase either in the presence or in the absence of *SCP*. In case of using *SCP* we varied *l=1* and *l=2*. *SCP* handles a round trip via two sessions s_1 and s_2 each responsible for one direction. Although we varied the distance in-between the PDAs from *50cm* to *10m*, interference between the nodes was permanent and a change in distance did not effect our measurements. We respectively sent 50 packets per session and per packet size. Fig. 5 illustrates the demonstrator set-up.

Related to the size of the packets, in absence of SCP, the average round trip time of a packet over altogether six hops varies between 9.6 ms for 64 byte packets to 21.4 ms for 1024 byte packets.[2] In presence of *SCP* and *l=1* these times increased to 92.3 ms for 64 byte packets to 105.3 ms for 1024 byte packets. By choosing *l=2* the average round trip times could be reduced to 61.5 ms for 64 byte packets to 75.9 ms for a 1024 byte packet size. More precisely, for a 128 byte packet size which is most suitable for VoIP traffic, the round trip time of 66.1 ms is a still acceptable end-to-end delay. Fig. 6 substantiates these values, whereas each line connects three marks indicating the average-, the minimum- and the maximum round-trip-time per packet sizes.

Decreasing the round trip times by choosing *l=2* instead of *l=1* comes at the cost of additional jitter caused at the intermediate nodes. Nevertheless to guarantee an acceptable

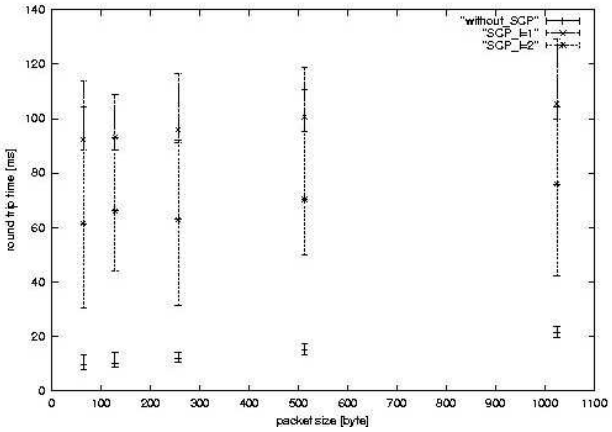

Fig.6. Round trip times and jitter on the demonstrator in absence or in presence of SCP with l=1 or l=2 and varying packet sizes.

end-to-end throughput this choice seems most promising for real-time responsive traffic.

Over WLAN 802.11b with a real throughput in the 4Mbps to 6Mbps range we measured for HTTP traffic a throughput of 194.2 Kbytes/s in absence of *SCP* and 52.3 Kbytes/s (*l=2*) resp. 44.1 Kbytes/s (*l=1*) in presence of *SCP*. Note, that due to the additional signing and verifying of acknowledgments, connection oriented TCP traffic is much more critical with

[2] Such values are generally higher than for one single session forwarding over five intermediate nodes. Obviously for our demonstrator set-up each node has to handle twice as much packets in parallel.

respect to throughput than audio streams or video streams over connectionless UDP traffic. Neither signing nor verifying the acknowledgments will heavily improve the above values.

To conclude, with our demonstrator environment and by applying the ECDSA reference implementation we did the validation that over a moderate number of intermediate nodes *SCP* is compatible to HTTP traffic, video-traffic and audio-traffic with still suitable performance characteristics.

V. Recommendations

It is our ambition to generalize our experiences according to digital signatures and to come up with careful recommendations for their usage in multi-hop ad hoc networks. We want to emphasize that the recommendations below do not only hold for the exemplary Sprite and SCP protocol, but for nearly every protocol regarding secure routing, accounting and charging, as well as cooperation based schemes.

A. Scalability of Security Relationships

With respect to whether choose a symmetric MAC based authentication or a digital signature based authentication we propose identifying the number of the required authentication relationships. Do we need *unilateral* authentications of x originators against y verifiers ($x \rightarrow y$ or $y \leftarrow x$) with $x,y \in \{1,...,N\}$? Or is it required to have *bilateral* authentications of x originators against y verifiers and vice versa ($x \leftrightarrow y$)? In case non-repudiation is not a required feature, we recommend to use MACs in the $N \rightarrow 1$ and $1 \leftrightarrow N$ cases, where N again is the number of nodes in the ad hoc cloud. With respect to the number of required keys, for all other authentication relationship scenarios ($1 \rightarrow N$, $N \rightarrow N$, $N \leftrightarrow N$, $M \rightarrow N$, $M \leftarrow N$, $M \leftrightarrow N$ with $1<M<N$) digital signatures clearly outperform MAC based solutions.

Table IV defends this statement. Note that even in a small ad hoc network with $N=100$ and $N \rightarrow N$ or $N \leftrightarrow N$ *100* public/private key pairs versus *4950* symmetric keys are required. Even in a $M \rightarrow N$ scenario with relatively small $M=5$ and again $N=100$, *5* public/private key pairs versus *485* symmetric keys are required.

B. Topology of the Network

In a 'conventional' architecture which is characterized by a single wireless hop, a restricted mobile client, and a powerful server in the fixed network, the digital signature generation at the client to authenticate towards the server is the only time-critical and CPU wasting operation. In contrast, in the context of multi-hop ad hoc networks the operations signature generation and signature verification become coequal operations. Both operations need to be executed by devices that tend to have lower power. For an appropriate choice of a digital signatur related to its minimum total execution time

TABLE IV
Number of Public/Private Key Pairs versus Number of Symmetric Keys for Various Unilateral or Bilateral Authentication Relationships.

	1→N	1←N	1↔N	N→N	1↔N	M→N	M←N	M↔N
Sig.	1	N-1	N-1	N	N	M	N	N
MAC	N-1	N-1	N-1	N(N-1)/2	N(N-1)/2	$\sum^{M}_{I=1}$N-I	$\sum^{M}_{I=1}$N-I	$\sum^{M}_{I=1}$N-I

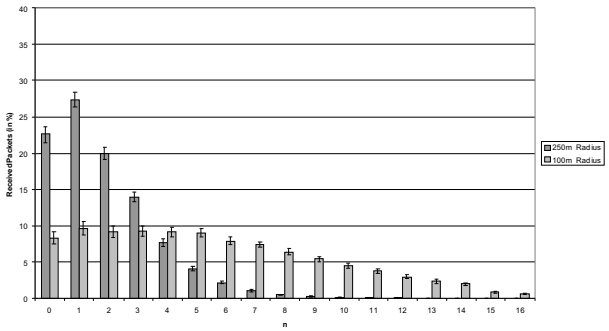

Fig.7. Traffic distribution per route length, e.g. with 100m vs. 250m transmission radius.

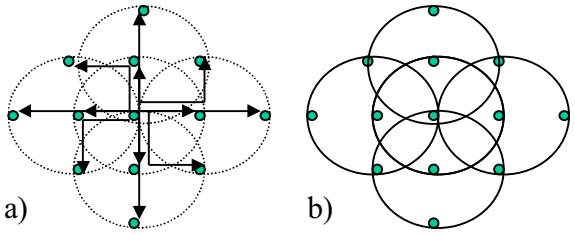

Fig. 8. a) Unicast of N symmetric key agreement information and b) broad-cast of one certificate to N nodes in a topology with C=4 and l=2.

$t_{CPU}(X)$ with $X \in \{S,D,N_i | i=1,...,n\}$, we propose considering the occuring traffic within the ad hoc network. More precisely, the distribution of the whole network traffic with respect to the required number of intermediate nodes to reach the final destinations needs to be considered.

TABLE V
NUMBER OF LOCAL BROADCASTS FOR A NEWCOMER'S CERTIFICATE TO REACH EACH NODE.

N	4	12	24	40	60	84	122	144
L	1	2	3	4	5	6	7	8
$\#_B$	1	5	9	17	25	37	47	62

To estimate the traffic distribution for two exemplary network topologies, we used the ns-2 simulator version 2.1b9 [23]. Nodes move according to the random waypoint model. Initially, 100 nodes are uniformly distributed in a *1000m·1000m* area. Subsequently, nodes choose new locations and move with a uniform velocity of *1-2 m/s* until they reach their destinations. The process is repeated unless the simulation time has elapsed. In our scenario, eight concurrent connections send five packets with 512 bytes per second where the average duration of a connection is 30 *sec*. The routing protocol is AODV. At the end of the connection a new source destination pair is immediately chosen. To achieve convincing data, every scenario was simulated 120 times with different source destination pairs. To show that the calculated values are reasonable approximations of the mean values the 95% confidence interval is depicted in Fig. 7.

As can be derived from Fig. 7, in case of 250m transmission radius, about 90% of all successfully received traffic goes over less than five nodes. For such a network topology and for protocols with similar security requirements and distribution of the digital signatur operations than *Sprite* or *SCP*, ECC-based signatures are preferable. In contrast, for a transmission radius of *100m* only about 41.5% of the arising network traffic is forwarded over less than five intermediate nodes. For such a topology, RSA turns out to be the faster digital signatur choice. Summing it up: For the appropriate choice of a digital signature scheme related to its total execution times, we need to understand the traffic distribution per route length within the ad hoc network.

C. Membership Dynamics

With each newcomer to the ad hoc network, a certificate needs to be broadcasted to the already participating actual nodes of the network. Fig. 2 illustrates that, independent of the security level or the security durability, the certificate length of ECDSA-based certificates is considerably smaller than an RSA-based one. Each newcomer obviously increases this advantage. Consequently, the higher the membership dynamics of the ad hoc network, the more favourable are ECDSA-based digital signatures. With respect to the short lifetime of a certificate we propose to handle a certificate revocation only implicitly. More precisely, each certificate is implicitly revoked after its expiration time. We expect a certificate to be valid for not longer than one day and thus no strong synchronization of the nodes' clocks is necessary.

With respect to a comparison to a MAC-based authentication, the actual number of nodes N of the ad hoc network as well as the connectivity of the network need to be taken into account. For instance, for agreeing on symmetric keys with a comparable security level to a 163-bit ECDSA by using the Diffie-Hellmann key agreement protocol [21], one needs $|d|=28$ bytes per Diffie-Hellmann message per security relationship (including 6 bytes for coding information). We consider an $N \leftrightarrow N$ scenario with $N=\sum_{i=1}^{l} i \cdot C$ actual members of the ad hoc network, where C is the connectivity of the network in term of direct neighbors for each node and l indicates the longest path from the newcomer to each node of the ad hoc cloud. Further, for simplicity we assume that nodes are static and uniformly distributed over a square area. The newcomer shall be located in the centre of the area.

In case of using MACs the newcomer has to initiate *unicast* traffic to each node of the cloud. The total number of hops over the wireless to reach all other nodes of the cloud can be denoted as $\sum_{i=1}^{l} i^2 \cdot C$. Fig. 8.a illustrates this for the case $C=4$ and $l=2$. Since the Diffie-Hellman key agreement protocol needs two messages to agree on a symmetric key, the costs for a MAC based solution in terms of additional data overhead over the wireless can be noted as follows:

$$Cost_{MAC}=\sum_{i=1}^{l} 2i^2 \cdot C|d| \quad (4)$$

In contrast, with a digital signature based approach, the newcomer *broadcasts* its certificate over the wireless, or, more precisely, by applying local broadcasts, the certificate is destined to all actual nodes of the cloud (Fig. 8.b). It turns out that, with respect to the considered topology, we cannot derive a cost function $Cost_{Sig}$ similar to Eq. 4. Instead, in Table V we list for different N and l with $C=4$ the resulting number of necessary local broadcasts $\#_B$ for a certificate. By

TABLE VI
PROPOSED RANGE OF USE FOR DIFFERENT DIGITAL SIGNATURE SCHEMES.

	RSA	ECDSA
network topology	medium to large average route length	small average route length
membership dynamics	static, weak	static, weak, high
majoritarian operation	signature verification	signature generation
security level	Relaxed	relaxed to strong
security durability	short to middle	short to long
traffic type	asynchronous, non time-restrictive	even synchronous, time-restrictive

computing

$$\sum_{i=1}^{l} 2i^2 \cdot C|d| \leq \#_B|c| \quad (5)$$

one can according to our network topology derive from what number of nodes in the ad hoc network digital signatures scale better than a MAC based authentication. Applying the certificate length illustrated in Fig. 2, 163-bit ECDSA signatures with a certificate size about |c|=380 bytes scale better than a MAC-based solution for networks greater than 40 nodes (and *l=4)*. For the same security level, RSA-based signatures with a certificate size of about |c|=500 bytes are even favourable for networks greater 84 nodes and *l=6*. Although we are aware of the fact that the chosen topology and the connectivity of the network have strong impact on the results, we feel that these values indicate that, with respect to the membership dynamics, digital signatures scale better for a wide range of network sizes than MAC-based solutions.

D. Security Level and Security Durability

Civilian ad hoc networks will become reality at the earliest in the years 2008 to 2010. Taking this observation into account, we can derive from Table III, that even for a rather relaxed targeted security level, a key size smaller than a comparable 163-bit ECDSA key is not acceptable. For instance, a 131-bit ECDSA key is expected to be sufficient for protocols of the considered classes for not longer than the year 2006. Thus, for protocols aiming at secure routing, the technical support for accounting and charging, security aspects for peer-to-peer services or cooperation in ad hoc networks, we recommend key sizes not smaller 163-bit ECDSA. Vice versa, since this key size is expected to provide a sufficient security level until 2015, one should not use longer keys.

E. Type of Traffic

For some particular applications the length of a digital signature becomes the bottleneck. In contrast to the execution times of the signature operations, we cannot significantly reduce the length of a signature while at the same time guaranteeing at least a minimal level of security. This observation needs to be taken into account when designing secure protocols suitable for real-time responsive traffic or for traffic with a high net throughput. Thus, we recommend for asynchronous, non time-restrictive traffic digital signatures per packet (except the acknowledgments). For synchronous and time-restrictive traffic we recommend the usage of multicast stream authentication based on digital sigatures or our simple yet realistic stream authentication proposal. In Table VI we summarize our recommendations.

VI. Conclusion

We considered the usage of RSA-based versus ECDSA-based digital signatures in ad hoc networks. Compared to an alternative MAC-based solution, we highlighted the benefit of digital signatures with respect to scalability and membership dynamics. It turns out that related to the required number of keys for various authentication relationship scenarios digital signatures clearly outperform a MAC-based solution. With respect to the membership dynamics of the ad hoc network and the certificate size, ECDSA-based certificates always perform better than RSA-based certificates. Compared to a MAC-based solution, assuming a 163-bit ECDSA security level in a network of more than 40 nodes, ECDSA performs better in terms of data overhead with respect to newcomers. For an RSA-based signature, this value doubles. Considering the total execution times of digital signature operations on each node for protocols similar to the requirements of Sprite or SCP, we recommend to take the traffic distribution of the ad hoc network per number of intermediate nodes into account. For networks with a small average route length and protocols with the signature verification as the most frequent operation, we recommend the usage of ECDSA-based signatures. In case of a longer average route length and assuming execution times of our reference implementation with a 163-bit ECDSA security level for a PDA clocked at 206MHz, RSA-based signatures turn out to be faster. We consider such a key size to provide a sufficient level of security even in the year 2008 which is the earliest we expect civilian ad hoc networks to become reality. Finally, with respect to the supported type of traffic and considering the unacceptable great signature length for real-time responsive traffic, we recommend using a stream authentication concept based on digital signatures. In contrast, for any file transfer application this is not necessary.

Acknowledgment

The authors would like to thank Krishna Paul for her useful feedback. We thank Ingo Riedel, Joao Paulo Barraca and Joao Francisco de Lima Lobo Girao for the implementation and demonstrator set-up of SCP. Further, we thank Marc Plaggemeier for his effort in doing the simulation.

References

[1] N. Ben Salem, L. Buttyan, J.-P. Hubaux, M. Jakobsson, "A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks", *ACM/SIGMOBILE 4th International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*,Jun. 2003, Annapolis, MD, USA, 2003.

[2] S. Buchegger, J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation of Nodes: Fairness In Dynamic Ad-hoc Networks)", *ACM/SIGMOBILE Third International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Jun. 2002, Lausanne, Switzerland, 2002.

[3] R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. "Adaptive Security for Threshold Cryptosystems." *Advances in Cryptology - Crypto'99*, LNCS 1666, 1999, Springer-Verlag pp. 98-115.

[4] R. Canetti, J. Garay, G. Itkis, D. Miccianicio, M. Naor, B. Pinkas, "Multicast Security: A Taxonomie and Some Efficient Constructions", *Proceedings of IEEE INFOCOM '99,* New York, USA, Mar. 1999.

[5] S. Capcun, L. Buttyan, J.-P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks", *IEEE Transactions on Mobile Computing*, pp. 52-64, Jan.-Mar. 2003.

[6] Certicom Webside http://www.secg.org/, secg – standards for efficient cryptography group.

[7] Certicom Corp. Certicom ecc challenge, since 1997. http://www.certicom.com/resources/ecc_chall//challenges.html.

[8] P. Golle, N. Modadugu, "Authentication for online Streams", *Proceedings of the Network and Distributed Security Symposium (NDSS)*, San Diego, California, February, 2001.

[9] B. Goode, , "Voice over Internet Protocol (VoIP)", In Proceedings of the IEEE, 90:1495-1517, Sep. 2002.

[10] V. Gupta, S. Gupta, D. Stebila, "Performance Analysis of Elliptic Curve Cryptography for SSL", *Proceedings of the 2002 ACM Workshop on Wireless Security*, Atlanta, Georgia, USA, 2002.

[11] I.P. Group, "IEEE P1363 standard", available at http://grouper.ieee.org/groups/1363/index.html.

[12] D. Hankerson, J.L. Hernandez, A. Menezes, "Software Implementation of Elliptic Curve Cryptography Over Binary Fields", *in Proceedings of Workshop on Cryptographic Hardware and Embedded Systems*, pp. 1-24, LNCS, Springer-Verlag, 2000.

[13] "IEEE P1363 Standard Specifications for Public-Key Cryptography", http://grouper.ieee.org/groups/1363/.

[14] D. Johnson, A. Menezes, S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", *A Certicom Whitepaper*, 2001.

[15] U. Jonsson, F. Alriksson, P. Johnson, T. Larsson, G. Q. Maguire Jr., "MIPMANET – Mobile IP for Mobile Ad Hoc Networks", *IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Boston, Massachusetts, USA, Aug. 2000.

[16] B. Lamparter, K. Paul, D. Westhoff, "Charging Support for Ad Hoc Stub Networks", *Elsevier Journal of Computer Communication, Special Issue on 'Internet Pricing and Charging: Algorithms, Technology and Applications',* Vol. 26, Issue 13, August 2003, pp. 1504-1514.

[17] B. Lamparter, M. Plaggemeier, D. Westhoff, "About the impact of Co-operation Approaches for Ad Hoc Networks", *ACM/SIGMOBILE Fourth International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Jun. 2003, Annapolis, Maryland, USA, 2003.

[18] A.K. Lenstra, E.R. Verheul, "Selecting cryptographic key sizes", *Journal of Cryptology: the journal of the International Association for Cryptologic Research 14 (4)* (2001) 255-293.

[19] Y.D. Lin, Y. Hsu, "Multihop cellular: A new Architecture for Wireless Communication", *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, Mar. 2000.

[20] C.H. Lim, P.J. Lee, "More flexible exponentiation with pre-computation", *Advances in Cryptography – Crypto'94*, LNCS 839, Springer-Verlag 1994, pp. 95-107.

[21] A. Menezes, P.C. v. Oorshot, S.A. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1965.

[22] P. Montgomery, "Speeding up the pollard and elliptic curve methods of factorisation", *Mathematics of Computation*, 48,1987, pp. 243-264.

[23] "The Network Simulator NS-2" , Available: http://www.ietf.org/html.charters/manet-charter.html.

[24] The openssl project, 1998-2000, Available: http://www.openssl.org.

[25] K. Paul, D. Westhoff, "Context Aware Detection of Selfish Node in DSR based Ad-hoc Networks", *Proceedings of IEEE GLOBECOM '02*, Taipei, Taiwan, November 2002.

[26] P. Papadimitrados, Z. Haas, "Secure Routing for Mobile Ad hoc Networks", *SCS Communication Networks and Distributed Systems Modelling and Simulation Conference (CNDS 2002)*, 2002.

[27] A. Perrig, R. Canetti, J.D. Tygar, D. Song, "Efficient and Secure Source Authentication for Multicast", *Network and Distributed System Security (NDSS) Symposium*, San Diego, California, February 2001.

[28] I. Riedel, "Security in Ad-hoc Networks: Protocols and Elliptic Curve Cryptography on an Embedded Platform", *diploma thesis*, March 2003.

[29] R. Rivest, A. Shamir, L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems", *Communications of ACM 21*, 2, 1978, pp. 120-126.

[30] P. Rohatgi, "A compact and fast hybrid signature scheme for multicast packet", *6th ACM Conference on Computer and Communication Security*, November 1999. pp. 93-100.

[31] RSA Laboratories, "Factoring Challenge", Available: http://www.rsasecurity.com/resources/ecc_chall/challenge.html.

[32] RSA Data Security. DES Challenge III, Jan. 1999, Available: http://www.rsasecurity.com/rsalabs/challenges/des3/.

[33] P.F. Syverson, D.M. Goldschlag, M.G. Reed, "Anonymous Connections and Onion Routing", *IEEE Journal on Selected Areas in Communication – Special Issue on Copyright and Privacy Protection*, 1998.

[34] A. Weimerskirch, D. Westhoff, „Zero Common Knowledge Authentication for Pervasive Networks", *Selected Areas in Cryptography (SAC'03),*Springer-Verlag LNCS 3006, Ottawa, Ontario, CA, August 2003.

[35] A. Weimerskirch, D. Westhoff, „Identity Certified Zero-Common Knowledge Authentication", *ACM Workshop on Security of Ad Hoc and Sensor Networks in conjunction with the Tenth ACM SIGSAC Conference on Computer and Communications Security, ACM SASN'03*, Oktober 2003.

[36] A. Weimerskirch, D. Westhoff, S. Lucks, E. Zenner, "Efficient Pairwise Authentication Protocols for Sensor Networks: Theory and Performance Analysis", *IEEE Press: Sensor Network Operations*, Editors: Jennifer Carruth, Thomas F. La Porta, IEEE Press Monograph, September 2004.

[37] M.J. Wiener, "Performance comparison of public-key cryptosystems", *CryptoBytes, Technical Newsletter of RSA Laboratories 4 (1) 1998* 1, 3-5.

[38] C.K. Wong, S. Lam, "Digital Signatures for Flow and Multicasts", *Proceedings of IEEE ICNP'98*, Austin, TX, October 1998.

[39] M.G. Zapata, "Secure Ad hoc On-Demand Distance Vector (SAODV) Routing", Internet Draft, draft-guerrero-manet-saodv-00.txt, Aug. 2001.

[40] S. Zhong, Y. R. Yang, J. Chen, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-hoc Networks", *Proceedings of IEEE INFOCOM '03,* San Francisco, USA, March 2003.

[41] L. Zhou, Z.J. Haas, "Securing Ad Hoc Networks", IEEE Network Magazine, vol 3, Nov./Dec. 1999.