DSA and ECDSA-based Multi-Signature Schemes

Hakim Khali, MIEEE and Ahcene Farah, SIEEE

Faculty of Computer Science & Computer Engineering
Ajman University of Science & Technology, PO.BOX 346, Ajman, UAE

Summary
This paper presents two new multi-signature schemes which aim at providing data authenticity, integrity, and non-repudiation. The proposed signing/verifying schemes are extensions of standardized algorithms, such as DSA (Digital Signature Algorithm) and ECDSA (Elliptic Curve DSA) algorithms. These schemes are faster than repeated individual signature (RDSA) using DSA or ECDSA (RECDSA) to generate a multi-signature. The final multi-signature of a message can be verified individually for each signer or collectively for a subgroup or entire group as well. Moreover, these schemes can also be used for group membership authentication. Finally, the proposed schemes can be used in E-commerce and E-government application. The security of the proposed schemes corresponds to the security of DSA and ECDSA algorithms respectively.

Key words: *Multi-signature, DSA, ECDSA*

1. Introduction

A digital signature is a bit pattern that depends on the message being signed, to prove the source of the data and protect against forgery. Digital signatures are dependent on public-key cryptography algorithms for their operation. Public-key cryptography relies on the availability of a pair of different but related keys: a private signing key *x* and a public verification key *y*. Several public-key cryptosystems [1]-[4], such as RSA [2], ElGamel [3][22], DSA or ECDSA's schemes [4][22], provide the necessary tools for accomplishing private and authenticated communication, and for performing secure and authenticated transactions over the internet as well as other open networks. Electronic layer-style organizations require the need to replace handwritten signatures by electronic ones. Legal electronic documents (contracts, cheque, etc) may require the signature of one party or several ones, which lead to multi-signatures [8-19]. Some multi-signature schemes do not require a pre-defined signing order. Such schemes are called Multi-signature schemes with undistinguished signing authorities. Other schemes require a pre-defined signing order and are called Multi-signature schemes with distinguished authorities. The signing process can be serial by repeating the selected scheme many times or parallel [15]. Finally, some signers can delegate proxies to sign a document on their behalf. This approach leads to Multi-signatures with proxy [21].

For most of the multi-signature schemes described in the literature, the following observations need to be mentioned:

- There are no official approved standards defining multi-signatures. Most of the schemes are specific for some E-applications [15].
- Several published multi-signature schemes have been proven un-secure and require additional modifications to reach an acceptable security level [30].

In our research work, we show how DSA and ECDSA standards can be modified to derive related multi-signature algorithms, while keeping the same level of security. The proposed two schemes belong to the class of Multi-signature schemes with undistinguished signing authorities. They will be referred as Digital Multi-signature Algorithm (DMSA) and Elliptic Curve Digital Multi-signature Algorithm (ECDMSA). Results show the superiority of the proposed schemes over conventional repeated DSA (RDSA) and repeated ECDSA (RECDSA). The rest of the paper is divided as follows. Section 2 and section 3 describe the mathematical background of the proposed DMSA and ECDMSA respectively. Section 4 discusses the security aspects of the proposed schemes. Section 5 presents the results and achieved performance over conventional DSA and ECDSA schemes respectively. Section 6 concludes the paper.

2. DSA-based Multi-Signature Scheme

2.1 Common Parameters

The common parameters are similar to those defined in [22, 24] for DSA standard to which we have added the group dimension. Assuming a group of *n* signers, where $signer_1$ is the group manager ***GM***, the following parameters are defined:

Manuscript received July 5, 2007
Manuscript revised July 25, 2007

- p, q: Two large prime numbers such that $q|(p-1)$ as defined in Digital Signature Algorithm [22].
- g: Generator of the cyclic group of order q in Z_p^* (selects an element $h \in Z_p^*$ and computes $g = h^{(p-1)/q} \bmod p$ such that $g \neq 1$).
- $x_1, x_2, \ldots, x_n$: Group members' secret keys such that $1 < x_i < q$, x_i is selected randomly and known only by the member P_i. Adding/deleting a member j requires adding/deleting the corresponding x_j by the **GM**.
- $y_1, y_2, \ldots, y_n$: Group members' public keys such that $y_i = g^{x_i} \bmod p$ is computed and published by the group members P_i. Adding/deleting a member j requires adding/deleting the corresponding y_j by the **GM**.
- $H(.)$: a cryptographic strong hash function (one-way function) such as SHA-1 or SHA-2 [6-7].

2.2 Signature Generation

The scheme requires the group manager ***GM*** and other signing group members to carry out an exchange of data during the signature generation process. The signature process for the manager is similar to the standard process described in [22] and [24] for one signer only.

- ***GM*** computes $H(M) = m$ (M is the message to be signed) and make the bit string m an integer.
- ***GM*** chooses a random integer k_1, $(1 < k_1 < q)$ and computes:

$a_1 = (g^{k_1} \bmod p) \bmod q$ (repeat until $a_1 \neq 0$) and $k_1^{-1} \bmod q$, where k_1^{-1} is the multiplicative inverse of k_1; i.e., $k_1 \cdot k_1^{-1} \bmod q = 1, \quad 0 < k_1^{-1} < q$.

- ***GM*** computes $b = k_1^{-1}(m + a_1 x_1) \bmod q$ (repeat until $b \neq 0$) and $s = b^{-1} \bmod q$.
- ***GM*** sends M and $Sign_1(M) = \{a_1, s\}$ to other signers (and keep k_1 secret).

Each other $Signer_i, i \neq 1$, checks the signature of the manager as follows:

- Verify that (a_1, s) are integers less than q and not equal to zero,
- Compute $H(M) = m$ and make m an integer less than q,
- Compute $u = m \cdot s \bmod q$ and $v = a_1 \cdot s \bmod q$,
- Use public key of ***GM*** to check if the following equation is true:

$$(g^u \cdot y_1^v \bmod p) \bmod q = a_1 \qquad (1)$$

Theorem 1 If equation (1) is true, then signature $Sign_1(M) = \{a_1, s\}$ of message M is valid.

Proof of Theorem 1

We have

- $b = k_1^{-1}(m + a_1 x_1) \bmod q, \quad s = b^{-1} \bmod q$ and $k_1 = s(m + a_1 x_1) \bmod q = k_1 \bmod q$. Using modular arithmetic properties, we obtain:

$k_1 = (sm) \bmod q + (sa_1 x_1) \bmod q + nq = w + nq$,

Where $w = (sm) \bmod q + (sa_1 x_1) \bmod q$. Therefore:

$g^{k_1} \bmod p = g^{w+nq} \bmod p = g^w g^{nq} \bmod p$.

Based on Fermat's little theorem, which implies that $g^q \bmod p = h^{(p-1)} \bmod p = 1$, the previous equation can be rewritten as $g^{k_1} \bmod p = g^w \bmod p$. Let us rewrite w as $w = u + vx_1$. We have:

$g^{k_1} \bmod p = g^w \bmod p = g^u g^{vx_1} \bmod p = g^u y_1^v \bmod p$,

which leads to:

$(g_1^{k_1} \bmod p) \bmod q = (g_1^u y_1^v \bmod p) \bmod q = a_1$.

End of proof.

Then each $Signer_i, i \neq 1$ computes:

- $k_i = s(m + a_1 x_i) \bmod q$ and $a_i = (g^{k_i} \bmod p) \bmod q$ (Change x_i or repeat steps 1 to 5 until $a_i \neq 0$).
- Each $Signer_i, i \neq 1$ sends $sign_i(M) = \{a_i\}$ to the manager (and keeps k_i secret).
- ***GM*** checks collectively or individually the group's members signature using the verification procedure as described in sub-section C for group signature authentication or membership authentication and sends the message M and its complete group signature $gsign(M) = \{a_1, a_2, \ldots, a_n, s\}$ to the final destination. Table 1 shows a comparison between RDSA and DMSA for signature generation in case of group authentication, assuming *N* signers. It shows advantages of JGSS in term of signature size and the number of random numbers generated by both algorithms. However, (DMSA & RDSA) schemes are almost equivalent and major differences will appear only for large group sizes.

Table 1: DMSA Scheme vs. RDSA scheme for group authentication (Signing process)

Parameter	Number of Operations		
	DMSA	RDSA	RDSA - DMSA
Number of components in signature	*N+1*	*2N*	*(N-1)*
Exponentiations	*N*	*N*	*0*
Products of 2 elements.	*2N*	*2N*	*0*
Inversion mod q	*2*	*1*	*-1*
Random numbers k	*1*	*N*	*N-1*
Modulus	*2(N+1)*	*2N*	*-2*

2.3 Signature Verification

Prior to verifying the signature of a signed message, the parameters (p,q,g,y_i) are made available to the verifier in an authenticated manner. This assertion is also true for the manager. The scheme provides four ways to verify the signature:

1- Verification that all members of the group have signed the message (or group signature authentication and non-repudiation).

2- Verification that a sub-group of group members (including the ***GM***) has signed the message (or subgroup signature authentication and non-repudiation).

3- Verification that a member of group has signed the message (or group membership authentication).

4- Verification that the manager has signed the message.

In all cases, the verifier performs the following:

- Verifies that (a_i, s) are integers less than q and not equal to zero.
- Computes $H(M) = m$ and make m an integer less than q,
- Computes $u = m \times s \bmod q$, and $v = a_1 \times s \bmod q$

a- Group signature authentication

The verifier computes: $a = (\prod_{i=1}^{n} a_i) \bmod q$ and checks if the following equation holds:

$$(\prod_{i=1}^{n} g^u y_i^v \bmod p) \bmod q = a \qquad (2)$$

Theorem 2 If equation (2) is true, then the group signature $gsign(M) = \{a_1, a_2, \ldots, a_n, s\}$ is valid.

Proof of Theorem 2

The proof of theorem 1 can be reused by just replacing (k_1, x_1) by (k_i, x_i), and a_1 by a_i respectively to prove that: $(g^u \cdot y_i^v \bmod p) \bmod q = a_i$. Since $a = (\prod_{i=1}^{n} a_i) \bmod q$ then $a = (\prod_{i=1}^{n} (g^u \cdot y_i^v \bmod p) \bmod q) \bmod q$ and finally $a = (\prod_{i=1}^{n} g^u \cdot y_i^v \bmod p) \bmod q$.

End of proof.

Remarque1: to implement this case only, the value a can be calculated during signature generation, and the group signature is reduced to $gsign(M) = \{a_1, a, s\}$. Therefore, the group signature size is reduced to three elements and the verification process is accelerated. A comparison between RDSA and DMSA for signature verification in case of group authentication, assuming N signers, is presented in table 2.

Table 2: DMSA scheme vs. RDSA for group authentication (Verification process)

Parameter	Number of operations		
	DMSA	RDSA	RDSA - DMSA
Exponentiations	*N+1*	*2N*	*N-1*
Products of 2 elements.	*2N*	*3N*	*N*
Inversion mod q	*0*	*1*	*1*
Modulus	*2N+1*	*3N*	*N-1*

Table 2 shows major advantages of DMSA over RDSA for message verification in term of exponentiations, products and modulus operations. The performance gain in case of message verification is therefore higher compared to the gain that can be achieved in case of message signing. This advantage is essential since in most applications, message verification occurs much more often than message signing; a message is usually signed once and verified several times.

b- Sub-group signature authentication (Including the manager)

The verification here works like for group authentication. We replace $\prod_{i=1}^{n}$ by $\prod_{i=r}^{t}$ and follow the same procedure.

c- Group membership authentication.

The verifier checks only that the following equation holds:

$$(g^u y_i^v \bmod p) \bmod q = a_i \qquad (3)$$

Theorem 3 If equation (3) is true, then signature $sign_i(M) = \{a_1, a_i, s\}$ of the sent message M is valid.

Proof of Theorem 3

The verification works like for theorem 1, where $y_1 = y_i$. All equations hold.
End of proof.
.

3. ECDSA-based Multi-Signature Scheme

3.1 Common Parameters

The common parameters are similar to those defined in [22, 24] for ECDSA standard to which we have added the group dimension. Assuming a group of n signers, the following parameters are defined:
- A field size q, where $q=p$ a large prime number with $p \geq 163$, or $q = 2^m$ with $m \geq 155$;
- An indication FR (field representation) of the representation used for the elements of F_q.
- Two field elements a and b in F_q which define the equation of the elliptic curve E over F_q (i.e., $y^2 = x^3 + ax + b$ in the case of $p = q$, and $y^2 + xy = x^3 + ax^2 + b$ in the case $q = 2^m$);
- Two field elements x_G and y_G which define a finite point $G = (x_G, y_G)$ of prime order in (F_q).
- The order n with $n > 2^{160}$ and $n > 4\sqrt{q}$;
- The co-factor $h = \#E(F_q)/n$.
- $H(.)$: a cryptographic strong hash function (one-way function) such as SHA-1 or SHA-2.
- $d_1, d_2, \ldots, d_n$: Group members' secret keys such that $1 < d_i < n$, d_i is selected randomly and known only by the group member P_i. Adding/deleting a member j requires adding/deleting the corresponding d_j by the GM.
- $Q_1, Q_2, \ldots, Q_n$: group members public keys such that $Q_i = d_i G$ is computed, validated as it is described in [24] and published by the group members P_i. A signer key pair (d_i, Q_i) should be associated with $D = (q, FR, a, b, G, n, h)$, a valid elliptic curve domain parameters and the corresponding signer *i*. This association can be assured with certificates. Adding/deleting a member j requires adding/deleting the corresponding Q_j by the GM.

3.2 Signature Generation

As for the DSA-Based approach, this scheme requires *signer*$_1$ (the group manager ***GM***) and other signing group members to carry out an exchange of data during the signature generation process. The manager signature process is similar to the standard process described in [22] and [24] for one signer only.

GM computes $H(M) = m$ (M is the message to be signed), and make this bit string to an integer.
GM chooses a random integer k, $(1 < k < n)$ and computes:
- $kG = (x_1, y_1)$ and convert x_1 to an integer $\bar{x}_1$;
- $a_1 = \bar{x}_1 \bmod n$. If $a_1 = 0$ then go to step 2.
- $k^{-1} \bmod n$

GM computes:
- $b = k^{-1}(m + d_1 a_1) \bmod n$. If $b = 0$ then go to step 2.
- $s = b^{-1} \bmod n$

GM sends M and $Sign_1(M) = \{a_1, s\}$ to other signers (and keep k secret). Each other $Signer_i$, $i \neq 1$, checks the signature of the manager as follows:
- Verify that (a_1, s) are integers less than n and not equal to zero,
- Compute $H(M) = m$, and make m an integer less than n,
- Compute $u = m \cdot s \bmod n$ and $v = a_1 \cdot s \bmod n$,
- Compute $P = uG + vQ_1$. If $P = O$, reject the signature.
- Convert the x-coordinate x_P of P to an integer $\bar{x}_P$ and compute $w = \bar{x}_P \bmod n$.
- Check if the following equation is true:

$$w = a_1 \qquad (4)$$

Theorem 4 If equation (4) is true, then signature $Sign_1(M) = \{a_1, s\}$ of message M is valid.

Proof of Theorem 4
We have
- $b = k^{-1}(m + a_1 d_1) \bmod n$, $s = b^{-1} \bmod n$. Then $k = s(m + a_1 d_1) \bmod n$
- $k = (sm \bmod n + sa_1 d_1 \bmod n) \bmod n$, which leads to $k = (u + vd_1) \bmod n$.

Therefore, $P = uG + vQ_1 = uG + vd_1 G$. By factoring G, we obtain $P = (u + vd_1)G = k_1 G \Rightarrow \bar{x}_p = \bar{x}$. We get $w = a_1$.
End of proof.

6. Then each $Signer_i$, $i \neq 1$ computes $a_i = s(m + a_1 d_i) \bmod n$
7. Each $Signer_i$, $i \neq 1$ sends $sign_i(M) = \{a_i\}$ to the

manager (and keeps d_i secret).

8. The group manager ***GM*** checks collectively or individually the group members' signature using the verification procedure for group signature authentication or membership authentication and sends to the final destination: M and $gsign(M) = \{a_1, a_2, \ldots, a_n, s\}$.

A comparison between repeated ECDSA and ECDMSA for signature generation in case of group authentication assuming *N* signers is presented in table 3. It shows great advantages of ECDMSA in term of signature size, point additions, modulus calculation and random number generation. These factors will lead to faster implementations.

Table 3: ECDMSA Scheme vs. RECDSA for group authentication (Signing process)

Parameter	Number of operations		
	ECDMSA	RECDSA	RECDSA - ECDMSA
Number of components in signature	*N+1*	*2N*	*(N-1)*
Point Addition	*k*	*$(K_1+k_2+\ldots)$~kN*	*k(N-1)*
Products of 2 el.	*2N*	*2N*	*0*
Inversion *mod q*	*2*	*1*	*-1*
Random *k* generation.	*1*	*N*	*N-1*
Modulus	*N+2*	*2N*	*N-2*

3.3 Signature Verification

The ECDSA-based scheme provides three ways to verify the signature: group signature authentication, subgroup signature authentication and group membership authentication. In all cases, the verifier performs the following:

- Verify that (a_1, s) are integers less than n and not equal to zero,
- Compute: $H(M) = m$, and make m an integer less than n,
- Compute: $u = m \cdot s \bmod n$, and $v = a_1 \cdot s \bmod n$,

a- Group signature authentication

The verifier computes $P = \sum_{i=1}^{N}(uG + vQ_i)$, and $R = \sum_{i=1}^{N} a_i G$, *N* is the number of signers. If $P = O$ or $R = O$, reject the signature.

- Convert the x-coordinate x_P of P to an integer $\bar{x}_P$ and computes $w_1 = \bar{x}_P \bmod n$.
- Convert the x-coordinate x_R of R to an integer $\bar{x}_R$ and computes $w_2 = \bar{x}_R \bmod n$.
- Check if the following equation is true:

$$w_1 = w_2 \quad (5)$$

Theorem 5 If equation (5) is true, then signature $gsign(M) = \{a_1, a_2, \ldots, a_n, s\}$ is valid.

Proof of theorem 5

We have $P = \sum_{i=1}^{N}(uG + vQ_i) = \sum(uG + vd_iG)$. By replacing (u, v), we get $P = \sum_{i}^{N}(sm + sa_1d_i)G = \sum_{i}^{N} a_iG = R$. This leads to $w_1 = w_2$.

End of proof.

Remarque2: to implement this case only, $a = w_1$ can be calculated during signature generation, and the group signature is reduced to $gsign(M) = \{a_1, a, s\}$. Therefore, the size of group signature is reduced to three elements and the verification process is accelerated. A comparison between RECDSA and ECDMSA for signature verification in case of group authentication assuming *N* signers is presented in table 4. It shows advantages of ECDMSA in term of point additions, products and modulus calculations.

Table 4: ECDMSA Scheme vs. RECDSA for group Authentication (Verification process)

Parameter	Number of operations		
	ECDMSA	RECDSA	RECDSA - ECDMSA
Point Addition	*N(v+1)+u*	*N(u+v)*	*N(u-1)-u*
Products of 2 elements	*2*	*2N*	*2N-2*
Inversion *mod q*	*0*	*1*	*1*
Modulus	*4*	*3N*	*3N-4*

b- Sub-group signature authentication (Including the group manager)

Using the group signature authentication described in a), sub-group signature authentication is achieved by replacing $\sum_{i=1}^{N}$ with $\sum_{i=r}^{t}$.

c- Group membership authentication.

- The verifier computes $P = uG + vQ_i$, and $R = a_iG$, if $P = O$ or $R = O$, reject the signature.
- Convert the x-coordinate x_P of P to an integer $\bar{x}_P$ and compute $w_1 = \bar{x}_P \bmod n$.
- Convert the x-coordinate x_R of R to an integer $\bar{x}_R$ and

compute $w_2 = \bar{x}_R \ mod \ n$.

- Check if the following equation is true:

$$w_1 = w_2 \quad (6)$$

Theorem 6 If equation (6) is true, then signature $gsign(M) = \{a_1, a_i, s\}$ is valid.

Proof of theorem 6

We have $P = uG + vQ_i = uG + vd_iG$. By replacing (u, v), we get $P = (sm + sa_1d_i)G = a_iG = R$. This leads to $w_1 = w_2$.

End of proof.

4. Security Considerations

Unforgeability is a required property for any basic signature scheme. In other words, it must be infeasible to compute the signature of a message with respect to a public key without knowing the corresponding secret key. The security of the public-key schemes is based on the assumption that there exists a hard problem that is difficult to be solved by the cryptanalyst, such as Factorization Problem, Discrete Logarithm Problem (DLP) [23] in finite group or Elliptic Curve DLP (ECDLP)[27]. Based on this problem and a trap door function, a secure public-key system can be constructed. By DLP we mean the problem of determining the least positive integer x, if it exists, which satisfies the equation:

$\beta \equiv \alpha^x \pmod p$, where p is a large prime number, and α, β are nonzero integers mod p. By ECDLP, we mean the problem of determining the least positive integer x, if it exists, which satisfies the equation:

$Q = xP$, where P and Q are two points on a prime elliptic curve E_p with p a large prime number or binary elliptic curve E_{2^m}

4.1 DSA-based Scheme

In the proposed scheme, the system parameters are similar to those used in DSA signature, which provides the same security level as DSA algorithm if the signature verification during the exchange process between the *manager* and other group members is done as it is indicated in the scheme. Secret keys x_i and numbers k_i should be kept secret. Security issues of DSA are explained in [24] and can be extended to our scheme. The security of the DSA-based multi-signature scheme relies on two distinct but related Discrete Logarithm (DL) Problems, $y_i = g_i^{x_i} \bmod p$ and $a_i = (g_i^{k_i} \bmod p) \bmod q$. The first one is the Discrete Logarithm Problem in Z_P^* where the number field sieve algorithm applies [25]. This algorithm has a sub-exponential running time:

$$O(exp((c + O(1))(Ln\,p)^{1/3}(LnLn\,p)^{2/3})) \quad (7)$$

Where $c \cong 1.923$ and $Ln\,p$ denotes the natural logarithm function. If $p = 1024$-bit prime, then equation (7) represents an infeasible amount of computation [24]. The second Discrete Logarithm Problem works to the base g in the subgroup of order q in Z_P^*. For a large p ($p \geq 1024$ bits), the best known algorithm for this problem is the Pollard's rho method [26], which requires a number of steps S given by:

$$S = \sqrt{(q/2)\pi} \quad (8)$$

If $q \cong 2^{160}$ ($q = 160$-bit prime), equation (8) represents an infeasible amount of computation [24].

4.2 ECDSA-based Scheme

In the proposed scheme, the system parameters are similar to those used in ECDSA signature, which provides the same security level as ECDSA algorithm if the signature verification during the exchange process between the ***GM*** and other group members is done as it is indicated in the scheme. Secret keys d_i and number k should be kept secret. If the elliptic curve is chosen carefully, or as it is recommended by NIST [22], and under the assumption that the hash function employed is collision resistant, ECDSA has been proven secure [27,28]. The best attacks known use the Pollard's rho method [26], which requires a number of steps S given by:

$$S = \sqrt{(n/2)\pi} \quad (9)$$

With $n > 2^{160}$, the order of the elliptic curve, equation (9) yields an infeasible amount of computation [24].

5. Simulation Results

In this section we will present results showing the performance of DMSA approach compared to RDSA. Both algorithms have been coded in C++ and run on a Pentium 4 processor clocked at 1 GHz. The worst case has been simulated, where all members of the group have to sign/verify the message. The simulations have taken into account two parameters: the size of the group, and the number of bits used to generate the common parameters (160 and 256 bits respectively).

Figures 1 and 2 show the execution times corresponding to

the signing process, while figures 3 and 4 show the execution times corresponding to the verifying process.

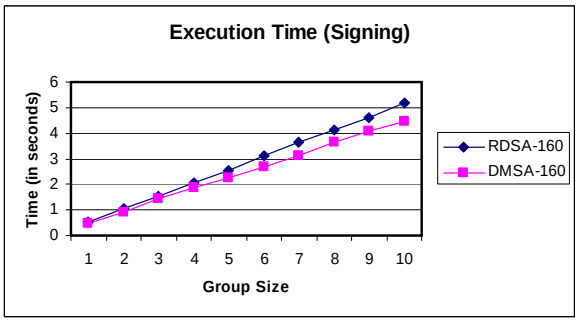

Fig. 1. DMSA vs. RDSA (160 bits Signing)

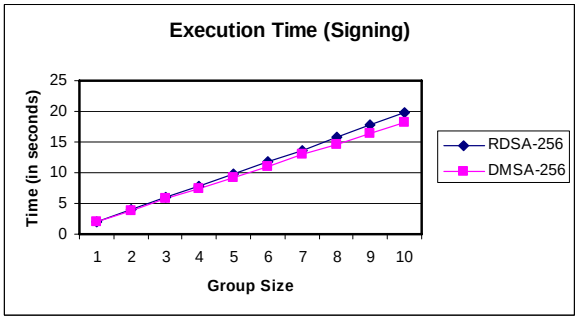

Fig. 2. DMSA vs. RDSA (256 bits Signing).

Figures 1 and 2 show that the superiority of DMSA over RDSA increases with the group size. However, both approaches are almost equivalent for small group sizes. This confirms the estimates presented in table 1.

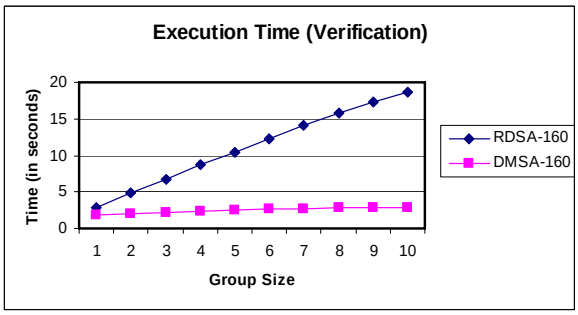

Fig. 3. DMSA vs. RDSA (160 bits Verification)

Figures 3 & 4 show the superiority of DMSA over RDSA in terms of execution times. They also show that increasing the size group has less impact on DMSA than RDSA, which is more sensitive as a result of the additional complexity. Figures 5 and 6 show the speed-up of DMSA over RDSA.

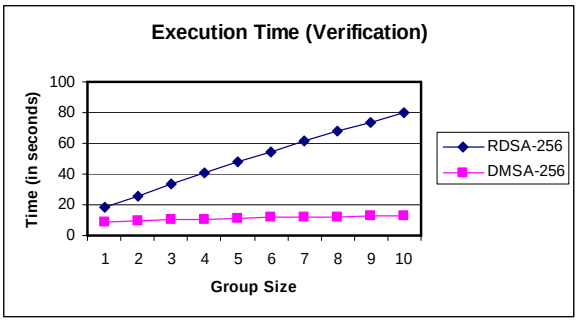

Fig. 4. DMSA vs. RDSA (256 bits Verification).

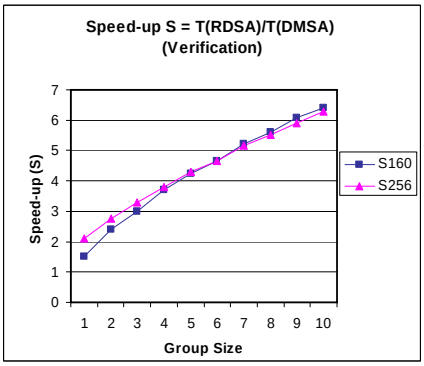

Fig. 5. Speed-up of DMSA over RDSA (Verification

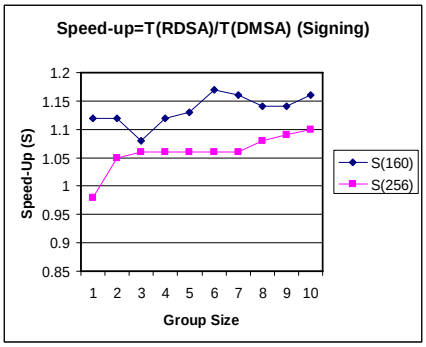

Fig. 6. Speed-up of DMSA over RDSA (Signing)

6. Conclusion

In this paper, two new multi-signature schemes are presented. These schemes provide data authenticity, integrity, and non-repudiation. The signature security is based on the security of DSA and ECDSA. The signature can be verified individually for one signer (or group membership authentication), for a subgroup of signers (or

subgroup authentication), or for all signers of the group (group authentication). These schemes are faster than repeated DSA or ECDSA. The performance gain is more significant in the case of signature verification, since a message is usually signed once and verified several times. Furthermore the group signature elements are connected together, which enhances the robustness of the proposed schemes against forgery. The new schemes can be directly used in many applications, such as E-Business for a joint signature of a contract between two or more organizations, or E-Government to sign an electronic legal document by many higher authorities, or in membership for access right authentication.

Acknowledgment

The authors would like to thank the Master student in Computer Engineering, Mr. Sohail Banihashimi, who helped in implementing and testing the new proposed schemes.

References

[1] W. Diffie and M.E. Hellman, New direction in cryptography, *IEEE Transactions On Information Theory*, vol. IT-22, (1976), 644-654.

[2] L.R. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signature and public-key cryptosystem, *Comm. ACM*, 21(2), (1978), 120-126.

[3] T. ElGamel, A public-key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions On Information Theory*, (31), (1985), 469-472.

[4] N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, (48), (1987), 203-209.

[5] C.k. Wong and S.S. Lam, Digital signature for flows and multicasts, *IEEE/ACM Transaction on networking*, 7(4), (1999), 502-513.

[6] National Institute of Standard and Technology, Secure Hash (SMS), FIPS Publication, 180-1, 1995.

[7] R.L. Rivest, The MD5 Message Digest Algorithm, *RFC 1321*, April (1992), 502-513.

[8] L. Harn and T. Kiesler, New scheme for digital multisignature, *Electronic Letters*, 25(15), (1989), 1002–1003.

[9] K. Itakura and K. Nakamura, A public-key cryptosystem suitable for digital multisignature, *NEC Research and Development*, (1971), 1-8.

[10] K. Ohta and T. Okamoto, Multisignature schemes secure against active insider attacks. *IEICE Trans. Fundamentals*, E82-A(1), (1999), 21–31.

[11] M. Burmester, Y. Desmedt, H. Doi, M. Mambo, E. Okamoto, M. Tada and Y. Yoshifuji, A Structured ElGamal-Type Multisignature Scheme, in: *Advances in Cryptology-Proceedings of PKC, Lecture notes in computer science*, Springer-Verlag (2000), 466-482.

[12] Z. Zhang and G. Xiao, New Multisignature Scheme for Specified Group of Verifiers, *Journal of Applied Mathematics and Computation*, 157(2), (2004), 425-431.

[13] P. Kotzanikolaou, M. Burmester and V. Chrissikopoulos, Dynamic multi-signatures for secure autonomous agents, in: *Proceedings 12th International Workshop on Database and Expert Systems Applications (DEXA 2001)*, IEEE Computer Society, 2001, pp. 587–591.

[14] C.-Y. Lin, T.-C. Wu and J.-J. Hwang, ID-based Structured Multisignature Schemes, *Advances in Network and Distributed Systems Security*, Kluwer Academic Publishers (IFIP Conference Proceedings 206), Boston, 2001, pp. 45-59.

[15] S. P. Shieh, C. T. Lin, W. B. yang and H. M. Sun, Digital Multisignature Schemes for Authenticating Delegates in Mobile Code Systems, *IEEE Transactions on Vehicular Technology*, 49(4), (2000), 1462-1473.

[16] C.J. Mitchell and N. Hur, On the security of a structural proven signer ordering multisignature scheme, in: B. Jerman-Blazic and T. Klobucar (eds.), in: *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security* (*CMS 2002*), Kluwer Academic Publishers (IFIP Conference Proceedings 228), Boston, 2002, pp.1-8.

[17] S. Mitomi and A. Miyaji, A Multisignature Scheme with Message Flexibility, Order Flexibility and Order Verifiability, in: *Proceedings of the 5th Australasian Conference on Information Security and Privacy*, Spring-Verlag, 2000, pp. 298-312.

[18] S. Micali, K. Ohta and L. Reyzin, Accountable-subgroup multisignatures: extended abstract, in: *Proceedings of the ACM Conference on Computer and Communications Security*, ACM press, 2001, pp. 245-254.

[19] Z.C. Li., L.C.K. Hui., K.P. Chow., C.F. Chong., W.W. Tsang and H.W. Chan, Cryptanalysis of Harn Digital Multisignature with Distinguished Signing Authorities, *IEE Electronics Letters*, 36 (4), (2000), 314-315.

[20] C. C. Chang and I. C. Lin, An improvement of delegated multisignature scheme with document decomposition, *ACM SIGOPS Operating Systems Review*, 38(4), (2004), 52-57.

[21] Y. L. Wang and L. H. Wang, A New Type of Digital Multisignature, in: *Proceedings of 9th Conference on Computer Supported Cooperative Work in Design*, 2(5), 2005, pp. 24-25.

[22] National Institute of Standard and Technology, Digital Signature Standard, FIPS Publication, 186-2, 2000.

[23] W. Trappe and L.C. Washington, *Introduction to cryptography with coding theory*, Prentice-Hall edition, 2002.

[24] D. Johnson, A. Menezes and S.Vanstone, The elliptic Curve Digital Signature Algorithm (ECDSA), White paper, w.w.w.certicom.com.

[25] D. Gordon, Discrete logarithms in GF(p) using the number field sieve, *SIAM Journal on Discrete Mathematics*, (6), (1993), pp. 124-138.

[26] J. Pollard, Monte Carlo methods for index computation mod p, *Mathematics of Computation*, (32), (1978), pp. 918-924.

[27] I. Blake, G. Seroussi and N. Smart, *Elliptic Curve in Cryptography*, London Mathematical Society Lecture Note Series, 265, Cambridge University Press 2004.

[28] D. Brown, The exact security of ECDSA, Technical report CORR 2000-54, Dept. of C&O, University of Waterloo, 2000, http://www.cacr.math.uwaterloo.ca.

[29] S. Zhou and D. Lin, Group signatures with reduced bandwidth, in: *Proceedings of IEE Information Security*, 153(4), 2006, pp. 146-152.

[30] S. Wang, G. Wang, F. Bao and J. Wang, Cryptanalysis of A Proxy-Protected Proxy Signature Scheme Based on Elliptic Curve Cryptosystem, in: *Proceedings of 60th IEEE Vehicular Technology Conference*, September 2004, pp. 3240-3243.

Dr. Hakim khali is an Assistant Professor of the Faculty of Computer Science & Computer Engineering at Ajman University of Science & Technology. He got his B.Sc in Computer Engineering from I.N.I (Algeria) in1989 and his M.Sc.A and PhD in 1993 and 2000 respectively from Ecole Poytechnique of Montreal (Canada). His research interests are Hardware-Software Codesign, VLSI architectures, and FPGA-based designs for Neural Networks and Cryptography. Before Joining Ajman University, he worked as a System Designer for Mirotech Microsystems on reconfigurable computing systems. Dr. Khali is an IEEE Member.

Prof.Ahcene Farah received the Electronics Engineer degree (1977) and Information Processing Master (1983) degree from the "Ecole Nationale Polytechnique" (ENP), Algiers, and Es-Science French State Doctorate degree (equivalent to PhD in computer engineering, 1989) from the National Polytechnic Institute of Lorraine, France (Institut National Polytechnique de Lorraine, INPL). He worked as teacher assistant from 1979 to 1984 at ENP. Between 1984 and 1989, he was Es-science Doctorate candidate at the INPL, and researcher at the Research Center in Automatics of Nancy. He was Associate-Professor (1990 –1998), then Professor (1998-1999) at the ENP, Algiers. September 1999– present, he is Professor at the faculty of Computer science and Computer Engineering, Ajman University, UAE. His research interests include**:** Computer and Network Information Security, Soft Computing (Neural Networks & Fuzzy Logic), Network Reliability, and Computer Forensic. He published more than 36 papers in Scientific Journals and Proceedings of International Conferences. He supervised 4 PhDs and 4 Masters of Science that have been completed and defended by the candidates.