Message Recovery for Signature Schemes Based on the Discrete Logarithm Problem

Kaisa Nyberg[1] and Rainer A. Rueppel[2]

[1] Vienna, Austria
[2] $\mathcal{R}^3$ Security Engineering AG, Switzerland

Abstract. The new signature scheme presented by the authors in [9] is the first signature scheme based on the discrete logarithm problem that gives message recovery. The purpose of this paper is to show that the message recovery feature is independent of the choice of the signature equation and that all ElGamal type schemes have variants giving message recovery and achieve five new signature schemes giving message recovery. These schemes have different properties as to implementation and security. It turns out that the scheme proposed in [9] is the only inversionless scheme whereas the message recovery variant of the DSA requires computing of inverses in both generation and verification of signatures. In [9] two applications of message recovery were proposed. In the present paper it is shown how to combine ElGamal encryption and the message recovery scheme of [9] and how to securely integrate the DSA into Diffie-Hellman key exchange.

1 Introduction

Two signature schemes have received widespread attention: the RSA scheme which is based on the difficulty of factoring and NIST's Digital Signature Algorithm (DSA) [6] which is based on the difficulty of taking logarithms *modulo* a prime p. Among the commonly accepted schemes the RSA is unique in the sense that the signature and the encryption transformations are inverses of each other. The RSA signature transformation can be used in two modes: with text hashing or message recovery. On the other hand, NIST's DSA only allows signatures in text hashing mode. We present a general procedure how to modify all previously presented signature schemes based on the discrete logarithm problem to allow message recovery. The advantages are obvious: applications without a hash function are possible, smaller bandwith for signatures of small messages, and direct integration into other schemes such as ElGamal encryption, identity-based public key systems or key agreement protocols. However, the new signature schemes with message recovery cannot be used for encryption as the RSA signature scheme by interchanging the roles of the public and private keys.

2 The Seminal Scheme of ElGamal

Let p be a prime and q equal to $p-1$ or to a large integer factor of $p-1$. Let $g \in \mathbf{Z}_p = GF(p)$ be an element of order q. These are the common parameters

in a network of users where a user or node has a private key $x \in \mathbf{Z}_q$ and a public key $y = g^x \bmod p$. For each message $m \in \mathbf{Z}_q$ to be signed a new and fresh random number $k \in \mathbf{Z}_q$ is privately generated.
In ElGamal's original scheme [5] $q = p-1$ and k is chosen to be relatively prime with $p-1$. The commitment part r of the signature is computed as $r = g^k \bmod p$. The second part s of the signature is then solved from the linear congruence $s = k^{-1}(m - rx) \bmod (p-1)$. Then the triplet $(m;(r,s))$ constitutes the signed message.
For the purpose of compact treatment, we consider in this paper a slight modification of ElGamal's original signature equation

$$s = k^{-1}(m + rx) \bmod q$$

where q is any large divisor of $p-1$ and one - sign is changed to a + sign. Also, throughout the paper we use the notation $r' = r \bmod q$. Correspondingly, the verification equation of ElGamal becomes

$$r^s = g^m y^{r'} \bmod p$$

In what follows this modification will be called ElGamal* scheme. Based on ElGamal's original idea several signature schemes have appeared in the literature [13], [6], [1], [14]. In Section 5 we discuss a general description of Elgamal type schemes which contains all previously proposed schemes as special cases. A similar "meta-scheme" has also been presented in [7].

3 How to Obtain Message Recovery for DLP-Based Signature Schemes

The currently discussed signature schemes based on the discrete logarithm problem (DLP), such as ElGamal and the DSA, have one major shortcoming when compared with the RSA. The RSA can provide

- message recovery: the message can be conveyed within the signature and can be recovered at the verifier's site. That is, the message need not be hashed or sent along with the signature which saves storage space and communication bandwidth.

which the DLP-based signature schemes cannot. On the other hand the RSA has a property, namely

- encryption: through reversal of the private and the public transformation the message can be encrypted.

In this paper we show that message recovery can be built as a general feature also into DLP-based schemes. The new schemes giving message recovery cannot be used as encryption algorithms. However, some of them can be naturally combined with ElGamal encryption [5] as we show in Section 8.1. This feature allows separation of confidentiality and authenticity functions.
Let us outline the **message recovery** approach:

1. Multiply the exponential (or its inverse) in the commitment r with the message m (or m^{-1}).
2. Replace the message m by 1 within the signature equation.
3. Rebuild the verification equation such that the exponential $g^k \bmod p$ is computed and the message can be recovered from the commitment part r of the signature (r, s).

4 Variants of the ElGamal* Scheme

4.1. The ElGamal scheme with Message Recovery

We show how to extend ElGamal* to provide message recovery. To sign a message $m \in \mathbf{Z}_p$ a random number $k \in \mathbf{Z}_q^*$ is generated and the signature (r, s) is computed as follows:

$$r = mg^{-k} \bmod p$$
$$s = k^{-1}(1 + r'x) \bmod q$$

The message m can be recovered by computing

$$m = g^{s^{-1}} y^{r's^{-1}} r \bmod p$$

We call this scheme MR(p)-ElGamal* scheme.

4.2. The DSA with Message Recovery

With the conventions of this paper the difference between the DSA of NIST and ElGamal is basically that ElGamal's signature (r, s) is replaced by the shorter signature $(r \bmod q, s)$. This reduces storage space and transmission bandwidth. To make the verification step work it is assumed that q is prime.
To sign a message $m \in \mathbf{Z}_p$ with the DSA, a random number $k \in \mathbf{Z}_q$ is generated and the signature (r, s) is computed as follows:

$$r = (g^k \bmod p) \bmod q$$
$$s = k^{-1}(m + rx) \bmod q$$

The corresponding verification equation for the DSA is

$$r = (g^{ms^{-1}} y^{rs^{-1}} \bmod p) \bmod q$$

We show now how to extend the DSA to provide message recovery. We will call the resulting scheme MR(q)-DSA to indicate that message recovery is provided within $\mathbf{Z}_q$. To sign a message $m \in \mathbf{Z}_q$ a random number $k \in \mathbf{Z}_q$ is generated and the signature (r, s) is computed as follows:

$$r = m^{-1}(g^k \bmod p) \bmod q$$
$$s = k^{-1}(1 + rx) \bmod q$$

The signature is as long and computation intensive as the DSA signature, but it is not necessary to send or store the message m along with the signature since it can be recovered from (r, s) as follows:

$$m = r^{-1}(g^{s^{-1}} y^{rs^{-1}} \bmod p) \bmod q$$

¿From the point of view of applications the MR(q)-DSA scheme has little or no advantage over the original DSA. The MR(p)-ElGamal* has potentially a much broader application range.

4.3. Equivalences

In this section we consider the relationship of ElGamal and DSA and their message recovery variants.

Definition. *Two signature schemes are called strongly equivalent if the signatures of the first scheme can be transformed efficiently into signatures of the second scheme and vice versa, without knowledge of the private key.*

Theorem 1. *Let q be a prime and $m \in Z_q$. Then the following signature schemes are strongly equivalent:* (i) ElGamal*, (ii) DSA *and* (iii) DSA *with message recovery in* $\mathbf{Z}_q$ (MR(q)-DSA).

Proof: We first prove equivalence between (i) and (ii). Let (r, s) be an ElGamal* signature to be appended to the message m. Then $(r \bmod q, s)$ is a DSA signature. Conversely, assume that (r, s) is a DSA signature, then $(y^{rs^{-1}} g^{ms^{-1}} \bmod p, s)$ is an ElGamal* signature.
Secondly, we prove equivalence between (ii) and (iii). Assume that (r, s) is a DSA signature of m. Multiplying the commitment and the signature equation by m^{-1} yields

$$m^{-1}r = m^{-1}(g^k \bmod p) \bmod q$$
$$m^{-1}s = k^{-1}(1 + xrm^{-1}) \bmod q$$

from which we see that $(m^{-1}r \bmod q, m^{-1}s \bmod q)$ is a MR(q)-DSA signature for m. Conversely, given a MR(q)-DSA signature (r, s) of m, we first recover m and obtain its DSA-signature as $(mr \bmod q, ms \bmod q)$.

4.4. How the DSA Should Have Been Defined

The reader may ask why the equivalence does not cover MR(p)- ElGamal*. The reason is the incompatibility of the modulo reductions in $\mathbf{Z}_p$ and $\mathbf{Z}_q$. For any integers a and b we must note that

$$(ab) \bmod q \neq (ab \bmod p) \bmod q.$$

Therefore the DSA does not have a strongly equivalent version giving message recovery in $\mathbf{Z}_p$ To avoid this shortcoming the DSA should have been defined as follows:

$$r = (mg^{-k} \bmod p) \bmod q$$
$$s = k^{-1}(1 + rx) \bmod q$$

This variant, which we call the reduced MR(p)-ElGamal* does not give message recovery and is best suited for use in text hashing mode by taking $m = H(M)$. Note that the scheme allows the size of the hash value to be any positive number up to $p - 1$. The only essential difference between the reduced MR(p)-ElGamal*

and the MR(q)-DSA (which is equivalent to the DSA) is that in the reduced MR(p)-ElGamal* the multiplication is done before reducing modulo q.
In this scheme the signed message contains the information $(m,(r,s))$ and verification is positive if the equality

$$r = (mg^{-s^{-1}} y^{-s^{-1}r} \bmod p) \bmod q$$

holds.

Theorem 2. *Let $m \in \mathbf{Z}_p$. Then the* MR(p)-ElGamal* *and its reduced version are strongly equivalent.*

Proof: Clearly if (r,s) is a MR(p)-ElGamal* signature for m, then m can be recovered and the reduced signature is $(r \bmod q, s)$. Conversely if m and its reduced signature (r,s) is given, then its MR(p)-ElGamal* signature can be efficiently computed as $(mg^{-s^{-1}} y^{-s^{-1}r} \bmod p, s)$.

5 The NEW Schemes

5.1. Search for an Inversionless Scheme

For the schemes presented above either the computation of signatures or the procedures for verification or message recovery involve inversion of elements in $\mathbf{Z}_q$ which requires q to be a prime if one wants to avoid repeated trials in the random parameter k. Specifically, we wish to find a scheme where

1. signatures can be computed without inverses;
2. the verification equation can be computed without inverses;
3. the verifier is able to recover $g^k \bmod p$, thereby allowing us to apply the message recovery technique.

Let us consider the following general description of ElGamal type DLP-based signature schemes. For all schemes the commitment is fixed as

$$r = g^k \bmod p$$

The generalized signature equation for ElGamal type schemes can be written as

$$ak + bx + c = 0 \bmod q \tag{1}$$

where the coefficients (a,b,c) involve the values of (r',s,m).
All previously proposed ElGamal type schemes are included in the cases where (a,b,c) is a permutation of $(\pm r', \pm s, \pm m)$, $(\pm r'm, \pm s, \pm 1)$ or $(\pm r', \pm sm, \pm 1)$. Let us now apply the message recovery approach presented in Section 3 to all these schemes. For all schemes the commitment r is computed as follows

$$r = mg^{-k} \bmod p$$

and the signature part s is solved from the equation

$$ak + bx + c = 0 \bmod q$$

where (a, b, c) is a permutation of $(\pm r', \pm s, \pm 1)$. For each of the six permutations we fix one combination of $\pm$ signs. The different signature equations for computing the second part s of the signature and the corresponding message recovery equations are the following.

	Signature Equation	Message Recovery Equation
(S1)	$sk - r'x - 1 = 0 \bmod q$	$m = g^{s^{-1}} y^{s^{-1}r'} r \bmod p$
(S2)	$r'k + sx - 1 = 0 \bmod q$	$m = g^{(r')^{-1}} y^{-s(r')^{-1}} r \bmod p$
(S3)	$k - r'x - s = 0 \bmod q$	$m = g^{s} y^{r'} r \bmod p$
(S4)	$sk - x - r' = 0 \bmod q$	$m = y^{s^{-1}} g^{s^{-1}r'} r \bmod p$
(S5)	$r'k + x - s = 0 \bmod q$	$m = y^{(r')^{-1}} g^{-s(r')^{-1}} r \bmod p$
(S6)	$k - sx - r' = 0 \bmod q$	$m = y^{s} g^{r'} r \bmod p$

We have chosen the $\pm$ signs in such a way that there is a direct correspondence to the existing ElGamal type schemes. Scheme (S1) is the MR(p) variant of ElGamal* scheme discussed in Section 4. The scheme proposed by Agnew, Mullin and Vanstone in [1] originally for use in $GF(2^n)$ leads to (S2). Scheme (S5) is the MR(p) variant of the scheme of Yen and Laih [14].

A message recovery signature scheme satisfies requirements 1-3 if and only if s and k can be solved from the signature equation without computation of inverses. As we immediately see (S3) is the only scheme to satisfy this requirement. On the other hand scheme (S1) derived from the DSA involves computation of inverses of different elements every time a signature is generated and verified. Let us notice that the Agnew-Mullin-Vanstone scheme was motivated by the fact that in signature generation it suffices to compute only one inverse x^{-1}. But the MR(p) variant (S2) of this scheme requires also inversion of r'. How to handle this problem if q is not prime was discussed by Piveteau in [12].

5.2. The NEW Signature Scheme and its Variants

There is no reason to presume that the number of inverses that have to be computed would be related to the security of the scheme. Therefore we choose the inversionless scheme (S3) to present a set of five signature schemes corresponding to the five variants of ElGamal* discussed in Section 4. Requirements 1 and 2 also apply also to the q-versions of the scheme. The three signature equations of the form (1) leading to (S3) are $km - r'x - s = 0$, $k - r'mx - s = 0$ and $k - r'x - sm = 0$, from which only the second one allows the computation of s and k without inverses.

1. p-NEW scheme (corresponding to ElGamal*)
2. MR(p)-NEW scheme with message recovery (corresponding to MR(p)-ElGamal*)
3. q-NEW scheme (corresponding to the DSA)
4. MR(q)-NEW scheme with message recovery (corresponding to MR(q)-DSA)
5. reduced MR(p)-NEW scheme (corresponding to the DSA variant presented in Section 4.4.)

The first two have a long commitment part, the other three are short.

Scheme	Signature	Recovery / Verification
p-NEW	$r = g^k \bmod p$ $s = k - r'mx \bmod q$	$r = g^s y^{r'm} \bmod p$
MR(p)-NEW	$r = mg^{-k} \bmod p$ $s = k - r'x \bmod q$	$m = g^s y^{r'} r \bmod p$
q-NEW	$r = (g^k \bmod p) \bmod q$ $s = k - rmx \bmod q$	$r = (g^s y^{rm} \bmod p) \bmod q$
MR(q)-NEW	$r = m(g^k \bmod p) \bmod q$ $s = k - rx \bmod q$	$m = (g^s y^r \bmod p)^{-1} r \bmod q$
Reduced MR(p)-NEW	$r = (mg^{-k} \bmod p) \bmod q$ $s = k - rx \bmod q$	$r = (mg^{-s} y^{-r} \bmod p) \bmod q$

We have the following strong equivalences.

Theorem 3. *Let q be prime and $m \in \mathbf{Z}_q$. Then the following signature schemes are strongly equivalent:* (i) p-NEW, (ii) q-NEW *and* (iii) q-NEW *with message recovery in* $\mathbf{Z}_q$ (MR(q)-NEW).

Proof: We first prove equivalence between (i) and (ii). If (r, s) is a p-NEW signature to be appended to the message m then $(r \bmod q, s)$ is a q-NEW signature. Conversely, if (r, s) is a q-NEW signature, then $(g^s y^{rm} \bmod p, s)$ is p-NEW signature.
Secondly, we prove equivalence between (ii) and (iii). If (r, s) is a q-NEW signature of m then (mr, s) is a MR(q)- NEW signature. Conversely, if (r, s) is MR(q)-NEW signature of m, then $(m^{-1}r, s)$ is a q-NEW signature provided that the inverse of m exists.

Note that the definitions and equivalence of p-NEW and q-NEW schemes do not impose any requirements on q, but MR(q)-NEW signatures can be properly defined and proved to be equivalent with the other two schemes only if q is a prime divisor of $p-1$. This will be the case for all MR(q) schemes. Note also that, for the same reason as MR(p)-ElGamal* is not covered by the equivalences in Theorem 1, we cannot include MR(p)-NEW in Theorem 3. However, analogously to Theorem 2 we have the following equivalence.

Theorem 4. *For messages in $\mathbf{Z}_p$ the* MR(p)-NEW *scheme and its reduced version are strongly equivalent.*

The reduced MR(p)-NEW does not provide message recovery and we propose its use for signatures with text hashing in an environment where authentication is based on the MR(p)-NEW scheme.
To conclude this section, let us notice that a similar set of five schemes with similar strong equivalences can be derived starting from any ElGamal type scheme.

6 Text Recovery and Text Hashing

The previous DLP-based schemes were not able to provide message recovery. For such schemes the signature is appended to the message and the verification is only possible if the message is known. All discussed ElGamal type schemes are vulnerable to substitution attack: given a valid signature for a message it is easy to modify the given signature in such a way that it is a valid signature for some other known message [5]. This attack is typically prevented by the use of a cryptographic hash function. This is inevitable also for the scheme in [14] too optimistically claimed to be secure without use of a hash function (see [3], [11]). In the verification procedure the hash value of the message is computed first and then the hash value is entered into the verification equation. The validity of the signature is established through checking the verification equation.
For schemes with message recovery the process runs differently. The verification equation recovers the message itself, but we need an additional step which tells us that the recovered message is the correct one. This is typically achieved through adding redundancy to the message before it is signed and through checking the redundancy after recovery. A good example for a redundancy generating function can be found in [8].
Of course, if a signature scheme provides message recovery, it can always be used in text hashing mode. Then the message is hashed and the hash value is signed. At the verifier the hash value is recovered (using the message recovery feature of the signature scheme) and the authenticity of the message is verified through comparison of the such recovered hash value with the locally computed hash value of the message. This is the process that most of us are accustomed to with applications of RSA.
To conclude this section let us mention the relationship between two variants of the NEW scheme and Schnorr's scheme [13]. Let H be a cryptographic hash function which maps messages M of arbitrary length to $\mathbf{Z}_q$ and set $h(g^k, M) = H(M)(g^k \bmod p) \bmod q$. If the MR($q$)-NEW scheme is used in text hashing mode it coincides with Schnorr's scheme [13] with the hash function h. This particular example of a hash function shows that to prevent non-repudiation of Schnorr's signatures it is essential that h is collision-resistant with respect to M. The reduced MR(p)-scheme gives a second example of Schnorr's schemes if we choose the hash function h to be $h(M, g^k) = (H(M)g^{-k} \bmod p) \bmod q$, where H is any collision-resistant hash function with values in $\mathbf{Z}_p$.

7 Security Considerations

7.1. Security Classes

To forge a signature for a given message without the knowledge of the private key one has to solve the signature (r, s) from the verification equation. Hence the security depends on the difficulty of the following problem:

> Given $g \in \mathbf{Z}_p$, $y \in \mathbf{Z}_p$ and $m \in \mathbf{Z}_p$ find $r \in \mathbf{Z}_p$ and $s \in \mathbf{Z}_q$
> such that the message recovery equation is satisfied.

In this sense, some of the MR(p)-schemes offer equivalent security. Indeed, (S1) and (S4), (S2) and (S5), (S3) and (S6) are pairs of schemes providing equivalent security since they are obtained from each other by interchanging the roles of the given quantities y and g. Note that the corresponding security equivalences hold for the p-variants of the schemes. For example, the Agnew-Mullin-Vanstone scheme and the Yen-Laih scheme are of equivalent security.
It is an open problem whether there exist other security equivalences. Specifically, it seems hard to say in what degree a p-scheme and the corresponding MR(p)-scheme are related. For example, no relevant definition of equivalence is known to justify the claimed equivalence in [12]. Of particular importance is the question whether the message recovery equations are as hard as the discrete logarithm problem. In the next subsection we discuss one aspect in which the proposed six MR(p)-schemes offer different security and which is of particular importance when message recovery signatures are combined with other DLP-based cryptosystems.

7.2. Forgery of Signatures for a Known Message with Known Log

As a consequence of the message recovery property it is possible to forge signatures of any given user with a known message. As discussed above this forgery is typically prevented by redundancy in the message. For some of the DLP-based message recovery signature schemes a stronger forgery is possible, which has to be taken into account in applications like authenticated key exchange or distribution of public keys described in [9].

Theorem 5. *Let q be a prime divisor of $p-1$. Then given a user U and a message $M \in \mathbf{Z}_p$ it is possible in schemes (S1), (S2), (S4) and (S6) to find $e \in \mathbf{Z}_q$, $r \in \mathbf{Z}_p$ and $s \in \mathbf{Z}_q$ such that (r,s) is a signature of U giving message recovery of the message $m = Mg^e \bmod p$.*

Proof: Without loss of generality we can consider the signature equation $ak + bx + c = 0 \bmod q$ where (a,b,c) is a permutation of $(r',s,1)$. Then the message recovery equation is

$$m = g^{-k}r = y^{a^{-1}b}g^{a^{-1}c}r \bmod p$$

where y is the public key of U. By substituting $m = Mg^e \bmod p$ we get the equation

$$Mr^{-1} = y^{a^{-1}b}g^{a^{-1}c-e} \bmod p \tag{2}$$

and we look for its solution r, s and e. We start by choosing any A and B in $\mathbf{Z}_q$ and computing

$$r = y^{-A}g^{-B}M \bmod p.$$

The schemes can be devided in three cases.
1. $r' = a$. Then $A = a^{-1}b \bmod q$ can be solved for b if and only if $b = s$ and then $s = Ar' \bmod q$. Hence $c = 1$ and we get $e = r'^{-1} - B \bmod q$. Consequently, this attack works for scheme (S2) but not for (S5).

2. $r' = b$. In this case $A = a^{-1}b \bmod q$ has always a solution $b = Aa$ if and only if $a = s$. Then $c = 1$ and with $e = s^{-1} - B \bmod q$ we have a solution of (2). From this we see that the forgery succeeds for scheme (S1) but not for (S3).
3. $r' = c$. Similarily, as in previous cases we can see that the forgery is possible for both schemes (S4) and (S6).

8 Applications

8.1. Signing and Encrypting

One of the main advantages of the new MR(p) signature schemes over the traditional ElGamal-type schemes is that they may be combined with ElGamal encryption in a natural manner. This is due to the mathematical fact that the messages are now elements of $GF(p)$. We show in detail how this combination works for (S3). It works equally well for (S5). But due to Theorem 5 this method is unsecure for the other MR(p) schemes, since the receiver can forge senders signatures for any given messages.
Let M be a message that A wants to send to B encrypted and signed. First A generates its MR(p) signature of M

$$r = f(M)g^{-k} \bmod p$$
$$s = k - r'x \bmod q$$

where f is a redundancy generating function. After that A encrypts r using ElGamal encryption [5] with the public key y_B of B and a privately generated random K. Then the signed and encrypted message consists of three parts

$$c_1 = g^K \bmod p$$
$$c_2 = f(M)g^{-k}y_B^K \bmod p$$
$$c_3 = s$$

When receiving (c_1, c_2, c_3) B decrypts r from c_2 using c_1 and its private key and then recovers $f(M)$ from r using c_3 and A's public key.
A second application of the MR(p) schemes and ElGamal encryption is the secret key establishment procedure described in [9] for the MR(p)-NEW scheme (S3). Using this procedure two parties can securely establish a shared secret session key by transfering only one message from one party to another. After a certain change this key establishment procedure is secure also if (S3) is replaced by any other MR(p) scheme. An implementation of this method is given in the next section.

8.2. How to Securely Integrate the DSA to Key Distribution

The parties who want to establish a shared secret and authenticated key can naturally authenticate their Diffie-Hellman key exchange messages ([4]) by signing them using the DSA of NIST. A more straightforward procedure proposed by Arazi [2] fails to give sufficient protection for the secrecy of the keys as shown in [10].

The purpose of this section is to show that Diffie-Hellman key establishment can be securely integrated to the DSA. Our starting point is to use the key establishment procedure of [9] with the MR(p)-ElGamal* signature scheme. Then we show that the most computation intensive parts of the procedure can be implemented using the DSA of NIST and essentially only some interface values need to be changed.
To generate the key exchange message the sender A

1. generates two random numbers K and k;
2. computes $r = y_B^K g^{-k} \bmod p$;
3. reduces $r' = r \bmod q$ as in the DSA;
4. computes $s = k^{-1}(H(M) + r' x_A) \bmod q$ as in the DSA but with $H(M) = 1$;
5. sends r and s to the other party B.

Then A computes the key as $g^K \bmod p$. Let us point out that in 2. we raise y_B, instead of g as in [9], to the exponent K to prevent the forgery described in Theorem 5. The receiving party B

1. recovers the value $g^k \bmod p$ by computing

$$g^{s^{-1}H(M)} y_A^{s^{-1}r'} \bmod p$$

   as in the DSA with $H(M) = 1$ but without reducing it modulo q;
2. computes $y_B^K = r g^k \bmod p$ and the session key as $(y_B^K)^{x_B^{-1}} \bmod p$.

9 Summary

We have presented a general idea how to derive new digital signature schemes giving message recovery from the previous schemes based on the difficulty of the discrete logarithm problem. Since the message recovery *modulo p* and *modulo q* can be combined with every signature equation we obtain many new schemes which we relate to each other in a systematic way within a general framework. Specifically, the framework allows us to compare the properties and functionality of the schemes. For example, we have seen that the NEW-schemes can be implemented without inverses *modulo q* and hence it is not necessary to choose q prime.
Although our new schemes give message recovery in the same way as the RSA, they cannot be used as encryption algorithms, since the signature and recovery transformations do not commute.
The benefits of the message recovery are: applications without a hash function, smaller bandwidth for signatures of short messages, direct use in other schemes such as identity-based public key systems (see [9]) or key agreement protocols and natural combination with ElGamal encryption. We also show how to securely integrate the DSA to secret key establishment.
We have seen that message recovery variants exist for all signature schemes based on the discrete logarithm problem in $GF(p)$. Further, it is obvious that

message recovery schemes can be considered over any group with a large cyclic subgroup, for example over $GF(2^n)$ or over an elliptic curve, that is, where ever ElGamal-type signature schemes exist. The main difference between the old ElGamal-type schemes and the new schemes giving message recovery lies in the fact that in the new schemes the messages to be signed are down in the group itself and not in the exponent set of integers as in the old schemes. As we have seen this mathematical fact is not only of theoretical significance but offers wider functionality and integrability to other cryptographic systems based on the discrete logarithm problem.

Acknowledgement
We wish to thank Jim Massey for his invaluable comments which greatly improved the presentation.

References

1. G. B. Agnew, B. C. Mullin and S. A. Vanstone, *Improved Digital Signature Scheme Based on Discrete Exponentiation*, Electronics Letters **26** (14), 1990, pp. 1024-1025
2. B. Arazi, *Integrating a Key Distribution Procedure into the Digital Signature Standard*, Electronics Letters **29** (11), 1993, pp. 966-967.
3. C. Boyd, *Comment: New Digital Signature Scheme Based on Discrete Logarithm*, Electronics Letters, **30** (6), March 1994, p. 480.
4. W. Diffie and M. Hellman, *New Directions in Cryptography*, IEEE Trans. Inform. Theory, IT-22(6), November 1976, pp. 644-654.
5. T. ElGamal, *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Trans. Inform. Theory, IT-31(4), July 1985, pp. 469-472.
6. FIPS PUB XX, 1993 February 1, *Digital Signature Standard.*
434,
7. P. Horster and H. Petersen, *Verallgemeinerte ElGamal-Signaturen*, Proceedings der Fachtagung SIS'94, Verlag der Fachvereine, Zürich 1994.
8. ISO/IEC 9796, Information technology–Security techniques– *Digital signature scheme giving message recovery.*
9. K. Nyberg and R. A. Rueppel, *A New Signature Scheme Based on the DSA Giving Message Recovery*, 1st ACM Conference on Computer and Communications Security, Nov 3-5, 1993, Fairfax, Virginia
10. K. Nyberg and R. A. Rueppel, *Weaknesses in Some Recent Key Agreement Protocols*, Electronics Letters, **30** (1), January 1994, pp. 26-27.
11. K. Nyberg, *Comment: New Digital Signature Scheme Based on Discrete Logarithm*, Electronics Letters, **30** (6), March 1994, p. 481.
12. J.-M. Piveteau, *New signature scheme with message recovery*, Electronics Letters, **29** (25), December 1993, p. 2185.
13. C. P. Schnorr, *Efficient Signature Generation by Smart Cards*, J. Cryptology, **4**, 1991, pp. 161-174.
14. S.-M. Yen and C.-S. Laih, *New Digital Signature Scheme Based on Discrete Logarithm*, Electronics Letters, **29** (12), 1993, pp. 1120-1121