

IEEE P1363a / D3+ (Additions to Draft Version 3)

Standard Specifications for Public Key Cryptography: Additions for “Pintsov-Vanstone” Integrated Signature Scheme with Recovery

Abstract. This addition to the third draft specifies techniques of an integrated (i.e. a hybrid of symmetric key and public key techniques) signature scheme with partial message recovery. Parameters of the scheme may be chosen to permit an arbitrary portion of the message to be recovered from the signature. Parameters of the scheme may also be chosen to securely decrease the signature length through exploitation of an agreed redundancy criterion for message acceptance.

The additional signature scheme here is incorporated in IEEE P1363a / D3 as a new set of primitives and a new signature scheme.

Keywords. Public-key cryptography; key agreement; digital signature; encryption.

Copyright © 2000 by the Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street
New York, NY 10017, USA
All rights reserved.

This is an unapproved draft of a proposed IEEE Standard, subject to change. Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of IEEE standardization activities. If this document is to be submitted to ISO or IEC, notification shall be given to IEEE Copyright Administrator. Permission is also granted for member bodies and technical committees of ISO and IEC to reproduce this document for purposes of developing a national position. Other entities seeking permission to reproduce portions of this document for these or other uses must contact the IEEE Standards Department for the appropriate license. Use of information contained in the unapproved draft is at your own risk.

IEEE Standards Department
Copyright and Permissions
445 Hoes Lane, P. O. Box 1331
Piscataway, NJ 08855-1331, USA

Contents

4 TYPES OF CRYPTOGRAPHIC TECHNIQUES (UPDATED)	3
4.3 SCHEMES (UPDATED).....	3
4.5 TABLE SUMMARY (UPDATED)	3
6. PRIMITIVES BASED ON THE DISCRETE LOGARITHM PROBLEM (UPDATED)	4
6.1.1 Notation (updated).....	4
6.2.12 DLPSP-PV (new).....	4
6.2.13 DLSP-PV (new).....	4
6.2.14 DLVP-PV (new).....	5
7. PRIMITIVES BASED ON THE ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM (UPDATED)	7
7.1.1 Notation (updated).....	7
7.2.12 ECPSP-PV (new).....	7
7.2.13 ECSP-PV (new).....	7
7.2.14 ECVP-PV (new).....	8
10. SIGNATURE SCHEMES (UPDATED)	10
10.6 DL/ECISSR (NEW)	10
10.6.1 Scheme Options	10
10.6.2 Signature Generation Operation	10
10.6.3 Signature Verification Operation	11
12. MESSAGE ENCODING METHODS (UPDATED)	13
12.3 MESSAGE ENCODING METHODS FOR SIGNATURES WITH MESSAGE RECOVERY (NEW).....	13
12.3.1 EMSR3	13
ANNEX D (INFORMATIVE) SECURITY CONSIDERATIONS (UPDATED)	15
D.5.2 SIGNATURE SCHEMES (UPDATED).....	15
D.5.2.1 Primitives (updated)	15
D.5.2.2 Encoding Methods (updated).....	15
ANNEX F (INFORMATIVE) BIBLIOGRAPHY (UPDATED)	16

4 Types of Cryptographic Techniques (updated)

Add the changes indicated below.

4.3 Schemes (updated)

Add the following item(s) to the list:

- Discrete Log or Elliptic Curve Integrated Signature Schemes with (Message) Recovery, in which one party signs a message using its private key, and any other party can verify the signature by examining the message, the signature and the signer's corresponding public key, and where part or all of the message may be recovered from the signature and need not be transmitted separately. The signer and the verifier agree on a set of domain parameters that include criteria for verifying redundancy in the message.

4.5 Table Summary (updated)

Add a DL/ECISSR entry in the table with the following:

Scheme Name	Primitives	Additional Methods
<i>signature schemes with message recovery</i>		
DL/ECISSR	DLPSP-PV, DLSP-PV and DLVP-PV Or ECPSP-PV, ECSP-PV and ECVP-PV	EMSR3

6. Primitives Based on the Discrete Logarithm Problem (updated)

Add new Sections 6.2.12, 6.2.13 and 6.2.14 as indicated below:

EDITOR'S NOTE—The new methods in this section are subject to revision to align with related work within ISO/IEC JTC1 SC27.

6.1.1 Notation (updated)

Add the following entries to the list of notation:

d	Signature part, an integer, computed by a signature primitive
i	Pre-signature, an integer, computed by a pre-signature primitive and a verification primitive

6.2.12 DLPSP-PV (new)

DLSP-PV is Discrete Logarithm Pre-Signature Primitive, Pintsov-Vanstone version. It is based on the work of [PV99]. The primitive generates a randomizer and a pre-signature for a signature scheme. It can be invoked in the scheme DLISSR as part of signature generation.

Input: the DL domain parameters q , r and g

Assumptions: DL domain parameters q , r and g are valid

Output:

- the randomizer, which is an integer u where $1 \leq u < r$
- the pre-signature, which is an integer i where $1 \leq i < q$

Operation. The randomizer u and the pre-signature i shall be computed by the following or an equivalent sequence of steps:

1. Generate a key pair (u, v) with the set of domain parameters. (See the note below.)
2. Convert v into an integer i with primitive FE2IP (recall that v is an element of $GF(q)$).
3. Output u as the randomizer and i as the pre-signature.

Conformance region recommendation. A conformance region should include:

- at least one valid set of DL domain parameters q , r and g

NOTE—The key pair in Step 1 should be a one-time key pair which is generated by the signer following the security recommendations of D.3.1, D.4.1.2, D.6 and D.7. A new randomizer / pre-signature pair should be generated for every signature. The randomizer should be stored following the same security recommendations as for a private key, and discarded after it is used in a signature generation operation. Similar to DLSP-NR, the key pair may be precomputed.

6.2.13 DLSP-PV (new)

DLSP-PV is Discrete Logarithm Signature Primitive, Pintsov-Vanstone version. It is based on the work of [PV99]. The primitive generates a signature part on a randomized message digest with the private key of the signer, given a randomizer, in such a way that a pre-signature can be recovered from the signature

using the public key of the signer by the DLVP-PV primitive. It can be invoked in the scheme DLISSR as part of signature generation.

Input:

- the DL domain parameters q , r and g associated with the key s
- the signer's private key s
- the randomizer, which is an integer u where $1 \leq u < r$
- the randomized message digest, which is an integer h such that $0 \leq h$

Assumptions: private key s and DL domain parameters q , r and g are valid and associated with each other; $1 \leq u < r$; $0 \leq h$

Output: a signature part, which is an integer d , where $0 \leq d < r$

Operation. The signature part d shall be computed by the following or an equivalent sequence of steps:

1. Compute an integer $d = u - sh \text{ mod } r$.
2. Output the pair d as the signature part.

Conformance region recommendation. A conformance region should include:

- at least one valid set of DL domain parameters q , r and g
- at least one valid private key s for each set of domain parameters
- all randomizers u in the range $[1, r - 1]$, where r is from the domain parameters of s
- all randomized message digests h in the range $[0, r - 1]$, where r is from the domain parameters of s

NOTE—For security reasons, the randomizer u should be discarded after step 2.

6.2.14 DLVP-PV (new)

DLVP-PV is Discrete Logarithm Verification Primitive, Pintsov-Vanstone version. It is based on the work of [PV99]. This primitive recovers a pre-signature from a signature part generated with DLSP-PV, given only the randomized message digest and public key of the signer. It can be invoked in the scheme DLISSR as part of signature verification and message recovery.

Input:

- the DL domain parameters q , r and g associated with the key w
- the signer's public key w
- the randomized message digest, which is an integer $h \geq 0$
- the signature part from the ephemeral public key is to be recovered, which an integer d

Assumptions: public key w and DL domain parameters q , r and g are valid and associated with each other

Output:

- the pre-signature i , which is an integer i

Operation. The pre-signature i shall be computed by the following or an equivalent sequence of steps:

1. If d is not in $[0, r - 1]$, output "invalid" and stop.
2. Compute a field element $j = \exp(g, d) \times \exp(w, h)$.
3. Convert the field element j to an integer i with primitive FE2IP.
4. Output i as the pre-signature.

Conformance region recommendation. A conformance region should include:

- at least one valid set of DL domain parameters q , r and g
- at least one valid public key w for each set of domain parameters
- all signature parts d that can be input to the implementation; this should at least include all d such that d is in the range $[0, r - 1]$, where r is from the domain parameters of W
- all randomized message digest $h \geq 0$

7. Primitives Based on the Elliptic Curve Discrete Logarithm Problem (updated)

Add new Sections 7.2.12, 7.2.13 and 7.2.14 as indicated below:

EDITOR'S NOTE—The new methods in this section are subject to revision to align with related work within ISO/IEC JTC1 SC27.

7.1.1 Notation (updated)

Add the following entries to the list of notation:

d	Signature part, an integer, computed by a signature primitive
i	Pre-signature, an integer, computed by a pre-signature primitive and a verification primitive

7.2.12 ECSP-PV (new)

ECSP-PV is Elliptic Curve Pre-Signature Primitive, Pintsov-Vanstone version. It is based on the work of [PV99]. The primitive generates a randomizer and a pre-signature for a signature scheme. It can be invoked in the scheme ECISSR as part of signature generation. .

Input: the EC domain parameters q, a, b, r and G

Assumptions: EC domain parameters q, a, b, r and G are valid

Output:

- the randomizer, which is an integer u where $1 \leq u < r$
- the pre-signature, which is an integer i where $1 \leq i < q$

Operation. The randomizer u and the pre-signature i shall be computed by the following or an equivalent sequence of steps:

1. Generate a key pair (u, V) with the set of domain parameters. (See the note below.) Let $V = (x_v, y_v)$ ($V \neq \mathbf{0}$ because V is a public key).
2. Convert x_v into an integer i with primitive FE2IP (recall that x_v is an element of $GF(q)$).
3. Output u as the randomizer and i as the pre-signature.

Conformance region recommendation. A conformance region should include:

- at least one valid set of EC domain parameters q, a, b, r and G

NOTE—The key pair in Step 1 should be a one-time key pair which is generated by the signer following the security recommendations of D.3.1, D.4.1.2, D.6 and D.7. A new randomizer / pre-signature pair should be generated for every signature. The randomizer should be stored following the same security recommendations as for a private key, and discarded after it is used in a signature generation operation. Similar to ECSP-NR, the key pair may be precomputed.

7.2.13 ECSP-PV (new)

ECSP-PV is Elliptic Curve Signature Primitive, Pintsov-Vanstone version. It is based on the work of [PV99]. The primitive generates a signature part on a randomized message digest with the private key of

the signer, given a randomizer, in such a way that a pre-signature can be recovered from the signature using the public key of the signer by the ECVP-PV primitive. It can be invoked in the scheme ECISSR as part of signature generation.

Except for having EC domain parameters as input rather than DL domain parameters, the primitive is identical to DLSP-PV.

Input:

- the EC domain parameters q, a, b, r and G associated with the key s
- the signer's private key s
- the randomizer, which is an integer u where $1 \leq u < r$
- the randomized message digest, which is an integer h such that $0 \leq h$

Assumptions: private key s and EC domain parameters q, a, b, r and G are valid and associated with each other; $1 \leq u < r$; $0 \leq h$

Output: a signature part, which is an integer d , where $0 \leq d < r$

Operation. The signature part d shall be computed by the following or an equivalent sequence of steps:

1. Compute an integer $d = u - sh \text{ mod } r$.
2. Output the pair d as the signature part.

Conformance region recommendation. A conformance region should include:

- at least one valid set of EC domain parameters q, a, b, r and G
- at least one valid private key s for each set of domain parameters
- all randomizers u in the range $[1, r - 1]$, where r is from the domain parameters of s
- all randomized message digests h in the range $[0, r - 1]$, where r is from the domain parameters of s

NOTE—For security reasons, the randomizer u should be discarded after step 2.

7.2.14 ECVP-PV (new)

ECVP-PV is Elliptic Curve Verification Primitive, Pintsov-Vanstone version. It is based on the work of [PV99]. This primitive recovers a pre-signature from a signature part generated with ECSP-PV, given only the randomized message digest and public key of the signer. It can be invoked in the scheme ECISSR as part of signature verification and message recovery.

Input:

- the EC domain parameters q, a, b, r and G associated with the key W
- the signer's public key W
- the randomized message digest, which is an integer $h \geq 0$
- the signature part from the ephemeral public key is to be recovered, which an integer d

Assumptions: public key W and EC domain parameters q, a, b, r and G are valid and associated with each other

Output:

- the pre-signature i , which is an integer i

Operation. The pre-signature i shall be computed by the following or an equivalent sequence of steps:

1. If d is not in $[0, r - 1]$, output “invalid” and stop.
2. Compute an elliptic curve point $P = dG + hW$. If $P = \mathcal{O}$, output “invalid” and stop. Otherwise, $P = (x_p, y_p)$.
3. Convert the field element x_p to an integer i with primitive FE2IP.
4. Output i as the pre-signature.

Conformance region recommendation. A conformance region should include:

- at least one valid set of EC domain parameters q, a, b, r and G
- at least one valid public key W for each set of domain parameters
- all signature parts d that can be input to the implementation; this should at least include all d such that d is in the range $[0, r - 1]$, where r is from the domain parameters of W
- all randomized message digest $h \geq 0$

10. Signature Schemes (updated)

Add new Section 10.6 as indicated below:

10.6 DL/ECISSR (new)

DL/ECISSR is Discrete Logarithm and Elliptic Curve Integrated Signature Scheme with Recovery.

SUBMITTER'S NOTE—This method is subject to revision to align with related work within ISO/IEC JTC1 SC27.

10.6.1 Scheme Options

The following options shall be established or otherwise agreed upon between the parties to the scheme (the signer and the verifier):

- the pre-signature, signature and verification primitives, which shall be one of the following triple of primitives:
 - the triple DLPSP-PV, DLSP-PV and DLVP-PV, or the triple ECPSP-PV, ECSP-PV and ECVP-PV
- the message encoding method for signatures with recovery, which should be EMSR3 (Section 12.3.3) including the encoding parameter *padLen*, which is an integer between 1 and 255 giving the amount of added redundancy in octets, or a technique designated for use with DL/ECISSR in an addendum to this standard
- the length of the part of the message that is to be recovered, given by an integer *mLen* (see Note 4)
- the redundancy criteria necessary for acceptance of the message after it has been recovered and successfully decoded (see Note 3)
- the message enciphering mechanism, which shall be one of:
 - a default stream cipher as indicated below based on an agreed mask generation function, which should be MGF1 (Section 14.2.1) or a mask generation function designated for use with DL/ECISSR in an addendum to this standard, or
 - another enciphering mechanism designated for use with DL/ECISSR in an addendum to this standard, with an agreed key derivation function, which should be KDF1 (Section 13.1) or a key derivation function designated for use with DL/ECISSR and the agreed enciphering mechanism in an addendum to this standard
- the hash function *Hash*, which shall be SHA-1 or RIPEMD-160

The above information may remain the same for any number of executions of the signature scheme, or it may be changed at some frequency. The information need not be kept secret.

10.6.2 Signature Generation Operation

A signature (C , d) and a (possibly empty) message part M_2 shall be generated by a signer from a message M (where the signer has ensured that the message M meets the agreed redundancy criteria) by the following or an equivalent sequence of steps:

1. Select a valid private key s and its associated set of domain parameters for the operation.

2. Apply the selected pre-signature primitive (see Section 10.6.1) to generate a randomizer u and a pre-signature i . Convert the pre-signature i to an octet string I of length $\lceil \log_{256} q \rceil$ octets using I2OSP.
3. Use the encoding operation of the selected message encoding method (see Section 10.6.1) to produce a message representative f (f will be a non-negative integer) and a (possibly empty) message part M_2 from the message M . (The encoding operation is given inputs M , $padLen$, and $mLen$.)
4. Convert f to an octet string T of length $\lceil \log_{256} f \rceil$ using I2OSP.
5. Generate a octet string C by enciphering T with a key derived from I , as follows:
 - if the agreed enciphering mechanism is the default stream cipher, let K be the key stream with the same length in octets as T derived from I with the agreed mask generation function and let $C = T \oplus K$
 - if another enciphering mechanism is the agreed one, let C be the encryption of T under a key K derived from I with the agreed key derivation function
6. Generate an octet string $H = Hash(C || M_2)$.
7. Convert H to an integer h with the primitive OS2IP.
8. Apply the selected signature primitive (see Section 10.6.1) to the randomized message digest h , the private key s , and the randomizer u to generate a signature part d .
9. Output the signature (C, d) and the message part M_2 .

Conformance region recommendation. A conformance region should include:

- at least one valid set of domain parameters
- at least one valid private key s for each set of domain parameters
- a range of messages M

10.6.3 Signature Verification Operation

A signature (C, d) shall be verified and the message M recovered by a verifier, given a (possibly empty) message part M_2 , by the following or an equivalent sequence of steps:

1. Obtain the signer's purported public key w' and its associated set of domain parameters for the operation.
2. (*Optional.*) Validate the public key w' and its associated set of domain parameters. Output "invalid" and stop if validation fails.
3. Generate an octet string $H = Hash(C || M_2)$.
4. Convert H to an integer h with the primitive OS2IP.
5. Apply the selected verification primitive (see Section 10.6.1) to the signature part d , the randomized message digest h and the signer's public key to recover a pre-signature i .
6. Generate an octet string T by deciphering C with a key derived from I , as follows:
 - if the agreed enciphering mechanism is the default stream cipher, let K be the key stream with the same length as C in octets derived from I with the agreed mask generation function and let $T = C \oplus K$
 - if another enciphering mechanism is the agreed one, let T be the decryption of C under a key K derived from I with the agreed key derivation function
7. Convert the octet string T to an integer f using the primitive OS2IP.
8. Use the decoding operation of the selected message encoding method (see Section 10.6.1) to verify that the integer f is a correctly encoded representative according to the encoding method and parameter $padLen$, and to recover the message M given the (possibly empty) message part M_2 . If the output of the decoding operation is "invalid," output "invalid." Otherwise, output the message M .

The verifier then ensures the message M meets the agreed redundancy criteria; if it does not, then reject the signature as invalid; else accept the signature as valid.

Conformance region recommendation. A conformance region should include:

- at least one valid set of domain parameters
- at least one valid public key w' for each set of domain parameters; if key validation is performed, invalid public keys w' that are appropriately rejected by the implementation may also be included in the conformance region
- all message parts M_2 that can be input to the implementation
- all purported signatures (C, d) that can be input to the implementation; this should at least include all (C, d) such that d is in the range $[0, r - 1]$, where r is from the domain parameters of w'

NOTES

1—The length of a message that can be signed by this scheme is unrestricted. The length of the message part that is not recovered is unrestricted.

2—These schemes, with appropriate restrictions on the scheme options and inputs, may be compatible with techniques in ISO/IEC 9796-4 [ISO99] (soon to be renamed ISO/IEC 9796-3).

3—The security of DL/ECISSR depends (in part) on the total of the amount of redundancy, which is $redundancy = (8 \times padLen) + agreed$ bits, where $agreed$ is the amount of redundancy in bits within the recoverable message part ensured by the agreed redundancy criteria for the acceptance of a message by both parties. (Redundancy in the visible message part M_2 does not affect the value of $agreed$.) Implementers should ensure that the total $redundancy$ is at least at an acceptable level, which is generally half the key size or half the hash length. Implementers may decide to reduce the amount of padding $padLen$ in order to shorten the signature, provided the $agreed$ value is high enough to make the total $redundancy$ acceptable.

4—The parameter $mLen$ does not have to be known to the verifier.

12. Message Encoding Methods (updated)

Add a new Subsection 12.3.3 as indicated below:

12.3 Message Encoding Methods for Signatures with Message Recovery (new)

Add a new Subsection 12.3.3 as indicated below:

12.3.1 EMSR3

EMSR3 is an encoding method for signatures with message recovery based on simple message padding. It is recommended for use with DL/ECISSR (Section 10.6).

The method is parameterized by the following choices:

- an integer *padLen* between 1 and 255 inclusive
- an integer *mLen* which is non-negative

12.3.3.1 Encoding Operation

Input:

- a message, which is an octet string M of length $mLen$ octets

Output:

- a message representative, which is an integer $f \geq 0$ or “error”
- a message part, which is an octet string M_2 of length $mLen - mLen$ octets if $mLen > mLen$, or an empty string otherwise

The message representative f shall be computed by the following or an equivalent sequence of steps:

1. Let $m2Len = mLen - mLen$. Let $max1Len = mLen$.
2. If $m2Len < 0$, let $m2Len = 0$ and $max1Len = mLen$.
3. Let M_1 be the leftmost $max1Len$ octets of M . Let M_2 be the rightmost $m2Len$ octets of M . (M_2 will be empty if $m2Len = 0$).
4. Convert $padLen$ to an octet string P_1 of length one, with primitive I2OSP.
5. If $padLen = 1$, let P_2 be the empty octet string. If $padLen = 2$, let P_2 be the octet string 01. If $padLen > 2$, let P_2 be the octet string of length $padLen - 1$ octets consisting of $padLen - 2$ octets with value 00 on the left and a single octet with value 01 on the right.
6. Let $T = P_1 || P_2 || M_1$.
7. Convert T to an integer f with primitive OS2IP.
8. Output f as the message representative and M_2 as the message part.

12.3.3.2 Decoding Operation

Input:

- the message representative, which is an integer $f \geq 0$
- the message part, which is an octet string M_2 of length $m2Len$ octets

Output: the message M or “invalid”

The message shall be recovered by the following or an equivalent sequence of steps:

1. Let $fLen = \lceil \log_{256} f \rceil$.
2. Convert f to an octet string T of length $fLen$ with I2OSP.
3. Let P_1 be the first octet of T .
4. Convert P_1 to an integer $padLen$ with OS2IP.
5. If $padLen \neq padLen$, stop and output "invalid".
6. If $padLen > 2$ and any of the $padLen - 2$ octets to the right of the first octet of T does not have value 00, then stop and output "invalid".
7. If the octet of T in position $padLen$ in the string does not have the value 01, then stop and output "invalid".
8. Let M_1 be the rightmost $fLen - padLen$ octets of T .
9. Output the octet string $M = M_1 || M_2$.

NOTES

1—The padding octet string $P_1 || P_2$ is: 01 if $padLen = 1$, 02 01 if $padLen = 2$, 03 00 01 if $padLen = 3$, 04 00 00 01 if $padLen = 4$, 05 00 00 00 01 if $padLen = 5$, and so on.

2—The octet string T is the shortest octet string encoding of the integer f . Since f is converted back to an octet string in DL/ECISSR, it is equivalent to use T directly in DL/ECISSR rather than performing unnecessary conversions back and forth between octet strings and integers.

Annex D (Informative) Security Considerations (updated)

Update Section D.5.2 of IEEE 1363-2000 and IEEE P1363a Draft 3 as indicated below:

D.5.2 Signature Schemes (updated)

Update Sections D.5.2.1 and D.5.2.2 as indicated below:

D.5.2.1 Primitives (updated)

Replace the first paragraph and the list of choices with the following:

Signature and verification primitive choices include the following pairs or triples (the particular choices vary among the signature schemes):

- DLSP-NR and DLVP-NR, DLSP-DSA and DLVP-DSA, DLPSP-NR and DLSP-NR2 and DLVP-NR2, DLPSP-PV and DLSP-PV and DLVP-PV, ECSP-NR and ECVP-NR, ECSP-DSA and ECVP-DSA, ECSP-NR and ECSP-NR2 and ECVP-NR2, ECPSP-PV and ECSP-PV and ECVP-PV, IFSP-RSA1 and IFVP-RSA1, IFSP-RSA2 and IFVP-RSA2, IFSP-RW and IFVP-RW

D.5.2.2 Encoding Methods (updated)

Update Section D.5.2.2 of IEEE P1363a Draft 3 as indicated below:

Replace the second sentence of the first paragraph with the following:

The recommended encoding methods for signatures with message recovery are EMSR1 (for DL/ECSSR), EMSR3 (for DL/ECISSR) and EMSR2 (for IFFSR).

Replace the fifth paragraph “An encoding method for a signature scheme with message recovery...” with the following:

An encoding method for a signature with message recovery in this method (i.e. DL/ECSSR, DL/ECISSR and IFFSR) should have the following properties, stated informally:

Add to the seventh paragraph “EMSR1 is considered to have...” the following sentence:

EMSR3 is considered to have these properties for DL/ECISSR provided the agreed redundancy criteria and the parameter *padLen* give a sufficient total amount of redundancy.

Annex F (Informative) Bibliography (updated)

Add the following new reference:

[PV99] L. Pintsov and S. Vanstone, "Postal Revenue Collection in the Digital Age", *Proceedings of the Fourth International Financial Cryptography Conference*, 2000.