Multi-Designated Verifiers Signatures: Anonymity without Encryption

Fabien Laguillaumie, Damien Vergnaud

Laboratoire de Mathématiques Nicolas Oresme
Université de Caen, Campus II, B.P. 5186,
14032 Caen Cedex, France

Abstract

In 1996, Jakobsson, Sako and Impagliazzo and, on the other hand, Chaum proposed the notion of *designated verifier signature* to solve some of the intrinsic problems of undeniable signatures. The generalization of this concept, suggested by Desmedt at Crypto'03's rump session, was formally investigated by Laguillaumie and Vergnaud at ICICS'04 as *multi-designated verifiers signatures.* The protection of the signer's privacy, as defined in that paper, seems difficult to achieve, and the protocols they proposed capture this property with an IND-CCA2 encryption of the signature. In this article, we propose the first multi-designated verifiers signature scheme which protects the anonymity of signers without encryption. This scheme is designed to be the extension of their B2DVS one and relies on Boneh *et al.*'s pairing-based ring signatures. The security of the new protocol relies, in the random oracle model, on the difficulty of solving the Diffie-Hellman problem in a bilinear setting.

Key words: cryptography, designated verifier signature, anonymity

1 Introduction

Designated verifier proofs, proposed in 1996 by Jakobsson, Sako and Impagliazzo [6] and Chaum [4], were introduced to solve some of the problems inherent to undeniable signatures. These proofs can be converted into *designated verifier signatures* via the Fiat-Shamir heuristic. It appears that they have numerous applications in commercial cryptography, and therefore, at Crypto'03's

Email addresses: laguillaumie@math.unicaen.fr (Fabien Laguillaumie), vergnaud@math.unicaen.fr (Damien Vergnaud).

rump session [5], Desmedt raised the problem of generalizing these signatures in a multi-user setting. In the model he proposed, the signature of a message is intended to a specific group of users, the *designated verifiers*, chosen by the signer, who will be the only ones able to check the validity of the signature. No one else than these verifiers can be convinced by this signature because by cooperating, they can also perform the signature by themselves. This new primitive was formally investigated, under the name of *multi-designated verifiers signatures*, by the authors in [9] where a generic multi-designated verifiers signature scheme based on discrete-log ring signatures was proposed (for the definition of ring signatures, see [11]).

As early as 1996, Jakobsson *et al.* suggested that designated verifer signatures should provide an additional notion of privacy: given such a signature and two potential signing public keys, it should be computationally infeasible for an eavesdropper to determine under which of the two corresponding secret keys the signature was performed. This property has been formalized by the authors in [8] and naturally extended to the multi-user setting in [9], where a bi-designated verifiers signatures was also proposed which takes advantage of Joux's non-interactive tripartite key exchange [7] to achieve this property. However, the generic scheme from [9] did not catch the notion of *privacy of signer's identity* without an additional encryption layer.

In this article, we introduce a new efficient multi-designated verifier signature scheme which is based on Boneh *et al.*'s ring signatures [2]. Our scheme captures the privacy of the signer's identity without encrypting the signatures. Although the new protocol requires more computational power for the signer and has a signature length proportional to the number of designated verifiers, it is *spontaneous* and does not require any prior exchange of secret key between the designated verifiers. It is therefore perfectly suited to applications in *ad hoc* groups.

2 Definitions

2.1 Multi-Designated Verifiers Signatures

In this section, we briefly recall the definition of multi-designated verifiers signatures. We refer the reader to [9] for a formal definition. Basically, a multi-designated verifiers signature scheme is composed of the following algorithms: a common parameter generator Setup; two key generation algorithms, $\mathsf{SKeyGen}$ for the signer, and $\mathsf{VKeyGen}$ for the designated verifiers; a signing algorithm Sign and a verification algorithm Verify.

These algorithms must satisfy:

(1) the *correctness* property *i.e.* a properly formed designated verifiers signature must be accepted by the verifying algorithm. Moreover, a putative signature is accepted by the verifying algorithm using one verifying secret key if and only if it is accepted using each verifying secret key;
(2) the *existential unforgeability against a chosen message attack*, whose specificity is that the attacker has access to a verifying oracle since he cannot verify the validity of a given signature by himself;
(3) the *source hiding*, which means that it is unconditionaly infeasible to determine, given a pair message/signature, who, from the signer or the designated verifiers all together, performed this signature;
(4) the *privacy of the signer's identity*, as defined in [8]. In a chosen message attack, an attacker has to determine given a pair message/signature and two potential signers, which one produced the signature.

We recall the definition of privacy of signer's identity from [9]. $\mathcal{H}$ is the random oracle, Σ_0 and Σ_1 are two signing oracles, and Υ is the verifying oracle.

Definition 1 (Privacy of signer's identity) *Let B be a set of n entities, k and t be integers and ε be a real in* $[0,1]$. *Let MDVS be an n-designated verifiers signature scheme with security parameter k, and let $\mathcal{A}$ be a PSI-CMA-adversary against MDVS. We consider the following random experiment, for $r \in \{0,1\}$:*

$$\boxed{\textit{Experiment } \mathbf{Exp}^{\textsf{psi-cma}-r}_{\textsf{MDVS},\mathcal{A}}(k)}$$

$$\text{params} \xleftarrow{R} \textit{MDVS.Setup}(k)$$

$$\textit{For } i=1,\ldots,n \textit{ do } (pk_{B_i}, sk_{B_i}) \xleftarrow{R} \textit{MDVS.VKeyGen}(\text{params}, B_i)$$

$$(pk_{A_0}, sk_{A_0}) \xleftarrow{R} \textit{MDVS.SKeyGen}(\text{params}, A_0)$$

$$(pk_{A_1}, sk_{A_1}) \xleftarrow{R} \textit{MDVS.SKeyGen}(\text{params}, A_1)$$

$$(m^\star, \mathcal{I}^\star) \leftarrow \mathcal{A}^{\mathcal{H},\Sigma_0,\Sigma_1,\Upsilon}(\textit{find}, \text{params}, pk_{B_1}, \ldots, pk_{B_n}, pk_{A_0}, pk_{A_1})$$

$$\sigma^\star \leftarrow \texttt{MDVS.Sign}(\text{params}, m^\star, sk_{A_r}, pk_B)$$

$$d \leftarrow \mathcal{A}^{\mathcal{H},\Sigma_0,\Sigma_1,\Upsilon}(\textit{guess}, \text{params}, m^\star, \mathcal{I}^\star, \sigma^\star, pk_{B_1}, \ldots, pk_{B_n}, pk_{A_0}, pk_{A_1})$$

Return d

where $\mathcal{A}$ has access to the oracles $\mathcal{H}$, Σ_0, Σ_1 and Υ. We define the advantage *of the adversary $\mathcal{A}$,* via

$$\mathbf{Adv}^{\textsf{psi-cma}}_{\textsf{MDVS},\mathcal{A}}(k) = \left| Pr\left[\mathbf{Exp}^{\textsf{psi-cma}-1}_{\textsf{MDVS},\mathcal{A}}(k)=1\right] - Pr\left[\mathbf{Exp}^{\textsf{psi-cma}-0}_{\textsf{MDVS},\mathcal{A}}(k)=1\right]\right|.$$

MDVS is said to be (k,t,ε)-PSI-CMA secure, if no adversary $\mathcal{A}$ running in time t has an advantage $\mathbf{Adv}^{\textsf{psi-cma}}_{\textsf{MDVS},\mathcal{A}}(k) \geq \varepsilon$.

As indicated in Jakobsson *et al.* and then proved in [8], an encryption of the signature with an IND-CCA2 encryption scheme ensures the privacy of the signer's identity. This means that the designated verifiers have to share a pair of encryption/decryption keys. This makes the protocol quite inefficient because it is no longer spontaneous. In [9], the authors proposed an efficient bi-designated verifiers signature scheme (B2DVS) and they obtained the anonymity of the signer thanks to Joux's tripartite key agreement [7], without encrypting the signature. The possibility of obtaining the privacy of the signer's identity without this encryption layer was left as an open problem.

2.2 Background

Our scheme is based on bilinear maps which were introduced in the cryptographers's world in 2000 with Joux's paper [7] (whereas they had appeared in the cryptanalists' world earlier with the work of Menezes, Okamoto and Vanstone [10] in 1991). We give here some definitions about these objects.

Definition 2 (Admissible bilinear map [1]) *Let* $(\mathbb{G}_0, +)$, $(\mathbb{G}_1, +)$ *and* $(\mathbb{H}, \times)$ *be three groups of the same prime order* q *and let* P_0 *and* P_1 *be two generators of* $\mathbb{G}_0$ *and* $\mathbb{G}_1$ *respectively. An* admissible bilinear map *is a map* $e : \mathbb{G}_0 \times \mathbb{G}_1 \longrightarrow \mathbb{H}$ *satisfying the following properties:*

(1) bilinear: $e(aQ, bR) = e(Q, R)^{ab}$ *for all* $(Q, R) \in \mathbb{G}_0 \times \mathbb{G}_1$ *and all* $(a, b) \in \mathbb{Z}^2$*;*
(2) non-degenerate: $e(P_0, P_1) \neq 1$*;*
(3) computable: there exists an efficient algorithm to compute e*.*

Definition 3 (prime-order-BDH-parameter-generator [1]) *A* prime-order-BDH-parameter-generator *is a probabilistic algorithm that takes on input a security parameter* k*, and outputs a 5-uple* $(q, P, \mathbb{G}, \mathbb{H}, e)$ *satisfying the following conditions:* q *is a prime with* $2^{k-1} < q < 2^k$, $\mathbb{G}$ *and* $\mathbb{H}$ *are groups of order* q, P *generates* $\mathbb{G}$*, and* $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{H}$ *is an admissible bilinear map.*

Both the unforgeability and the privacy of the signer's identity of our scheme rely on the Computational Diffie-Hellman Assumption, which, roughly speaking, says that given two points aP and bP in a group $\mathbb{G}$ of prime order generated by P, it is computationaly infeasible to compute the point abP. We recall below a formal quantitative definition of CDH:

Definition 4 (CDH) *Let* Gen *be a prime-order-BDH-parameter-generator. Let* D *be an adversary that takes on input a 5-tuple* $(q, P, \mathbb{G}, \mathbb{H}, e)$ *generated by* Gen*, and* $(X, Y) \in \mathbb{G}^2$ *and returns an element of* $Z \in \mathbb{G}$*. We consider the following random experiments, where* k *is a security parameter:*

$$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{cdh}}_{\mathsf{Gen},D}(k)}$$

$(q, P, \mathbb{G}, \mathbb{H}, e) \xleftarrow{R} \mathsf{Gen}(k)$

$\boldsymbol{setup} \leftarrow (q, P, \mathbb{G}, \mathbb{H}, e)$

$(x, y) \xleftarrow{R} [\![1, q-1]\!]^2$

$(X, Y) \leftarrow (xP, yP)$

$Z \leftarrow D(\boldsymbol{setup}, X, Y)$

Return 1 *if* $Z = xyP$, 0 *otherwise*

We define the corresponding success *of D in solving the CDH problem* via $\mathbf{Succ}^{\mathsf{cdh}}_{\mathsf{Gen},D}(k) = Pr\left[\mathbf{Exp}^{\mathsf{cdh}}_{\mathsf{Gen},D}(k) = 1\right]$.

Let $t \in \mathbb{N}$ and $\varepsilon \in [0, 1]$. CDH is said to be (k, t, ε)-secure if no adversary D running in time t is successful with $\mathbf{Succ}^{\mathsf{cdh}}_{\mathsf{Gen},D}(k) \geq \varepsilon$.

3 Description of the SMDVS scheme

Let k be a security parameter. We denote by A the signer, and by B_i a designated verifier, for $i \in [\![1, n]\!]$.

Let SMDVS be the new strong multi-designated verifiers signature scheme, which is an extension of B2DVS from [9]. It is based on Boneh *et al.*'s ring signature [2]. As in the B2DVS scheme, Boneh *et al.*'s ring signature gives the source hiding property which is also a requirement for ring signature, and since it is "discrete log" based it insures the property for the MDVS scheme itself. A Diffie-Hellman key distribution is used to compute the point Y during the signature generation. The important fact is that this key distribution is essentially 2-round, and one of the users has a special role. This user in the MDVS setting is naturally the signer. The point Y can be seen as an "anonymity" key. It makes it possible to achieve the privacy of the signer's identity.

The new scheme SMDVS is described in figure 1.

4 Security results

The correctness of SMDVS is obvious, as well as the source hiding property which naturally comes from the source hiding of the underlying ring signature

Setup: Let Gen be a prime-order-BDH-parameter-generator and $(q, P, \mathbb{G}, \mathbb{H}, e)$ be the output of $\mathsf{Gen}(k)$. Let $[\{0,1\}^* \times \mathbb{G}^{n+2} \longrightarrow \mathbb{G}]$ be a hash function family, and H be a random member of it.

SKeyGen: it randomly picks an integer $a \in [\![1, q-1]\!]$ which is the secret key of the signer A. Her public key is $P_A = aP$.

VKeyGen: it randomly picks an integer $b_i \in [\![1, q-1]\!]$ which is the secret key of the verifier B_i. His public key is $P_{B_i} = b_i P$.

Sign: Given $m \in \{0,1\}^*$, A computes the key $P_B = P_{B_1} + \cdots + P_{B_n}$, she picks at random an integer $r \in [\![1, q-1]\!]$ and sets $Y_{B_i} = rP_{B_i}$ for all $i \in [\![1, n]\!]$. Then, she computes the point $Y = rP$. She hashes $M = H(m, P_A, P_{B_1}, \ldots, P_{B_n}, Y)$, picks at random $r' \in [\![1, q-1]\!]$ and computes $Q_A = a^{-1}(M - r'P_B)$ and $Q_B = r'P$. The $n+2$-tuple $\sigma = (Q_A, Q_B, Y_{B_1}, \ldots, Y_{B_n})$ is the signature of m.

Verify: Given $m \in \{0,1\}^*$ and σ, each B_i retrieves $Y = rP$ as $b_i^{-1} Y_{B_i}$. He verifies, for $j \in [\![1, n]\!] \setminus \{i\}$, that $e(P_{B_j}, rP) = e(Y_{B_j}, P)$. Then he can compute $M = H(m, P_A, P_{B_1}, \ldots, P_{B_n}, Y)$ and checks whether $e(M, P) = e(Q_A, P_A) e(Q_B, P_B)$.

Fig. 1. SMDVS: Description

scheme. In the following, we state the security results concerning the unforgeability and the privacy of signer's identity of SMDVS. The proofs are carried in the random oracle model introduced in [3].

Theorem 1 (Unforgeability of SMDVS) *Let Gen be a prime-order-BDH-parameter-generator and let SMDVS be the associated multi-designated verifiers signature scheme. For any EF-CMA-adversary $\mathcal{A}$, in the random oracle model, against SMDVS, with security parameter k which has success $\varepsilon = \mathbf{Succ}^{\mathsf{ef-cma}}_{\mathit{SMDVS},\mathcal{A}}$, running time τ, and makes $q_{\mathcal{H}}$, q_Σ and q_Υ queries to the random oracle, the signing oracles and the verifying oracle, there exists an adversary $\mathcal{D}$ for CDH which has advantage $\varepsilon' = \mathbf{Succ}^{\mathsf{cdh}}_{\mathit{Gen},\mathcal{D}}(k)$ running in time $\tau' \in \mathbb{N}$ such that*

$$\begin{cases} \varepsilon' \geq \left(\dfrac{1}{n}\varepsilon - \dfrac{q_{\mathcal{H}} q_\Sigma + 1}{2^k}\right)^2 \\ \tau' \leq 2\left(\tau + (q_{\mathcal{H}} + 2q_\Sigma + O(1)) T_{\mathbb{G}} + q_\Sigma T_{\mathbb{H}}\right) \end{cases}$$

where $T_{\mathbb{G}}$ and $T_{\mathbb{H}}$ denote the time complexity to evaluate a discrete exponentiation in $\mathbb{G}$ and $\mathbb{H}$.

PROOF: It is a straightforward adaptation of the proof of unforgeability of B2DVS from [9]. Details are left to the reader.

Theorem 2 (Anonymity of SMDVS) *Let Gen be a prime-order-BDH-parameter-generator and let SMDVS be the associated multi-designated verifiers signature scheme. For any PSI-CMA-adversary $\mathcal{A}$, in the random oracle model, against SMDVS, with security parameter k which has advantage $\varepsilon = \mathbf{Adv}^{\mathsf{psi-cma}}_{\mathit{SMDVS},\mathcal{A}}(k)$, running time τ, and makes $q_{\mathcal{H}}$ queries to the random oracle, q_Σ queries to the signing oracles and q_Υ queries to the verifying oracle, there exists an adversary $\mathcal{D}$ for CDH which has advantage $\varepsilon' = \mathbf{Succ}^{\mathsf{cdh}}_{\mathit{Gen},\mathcal{D}}(k)$ running in time $\tau' \in \mathbb{N}$ such that*

$$\begin{cases} \varepsilon' \geq \left(\dfrac{\varepsilon}{2} - \dfrac{q_\Sigma + q_\Upsilon}{2^k}\right)^2 \\ \tau' \leq 2\tau + 2(q_{\mathcal{H}} + O(1))(q_\Upsilon + O(1))T_P \end{cases}$$

where T_P denotes the time complexity to evaluate a pairing.

PROOF: Let k be a security parameter, and Gen be a prime-order-BDH-parameter-generator. $(q, P, \mathbb{G}, \mathbb{H}, e)$ is an output of $\mathsf{Gen}(k)$. Let $R_x = xP$, $R_{xy} = xyP$ be two elements in $\mathbb{G}$ for (x, y) in $[\![1, q-1]\!]^2$. We construct a machine $\tilde{\mathcal{D}}$ which computes the point yP from these points. The CDH problem can be solved by solving two instances of this previous problem (see [2]).

In the real attack game, n pairs of keys (P_{B_i}, b_i) for $i \in [\![1, n]\!]$ are produced by the key generation algorithm for the verifiers, and two pairs of keys (P_{A_0}, a_0) and (P_{A_1}, a_1) are produced by the key generation algorithm for the signers. The PSI-CMA adversary $\mathcal{A}$ is fed with the n public keys of the verifiers and the public keys of the two potential signers. It outputs a message $m^\star$ at the end of its `find` stage. Then a signature is performed by flipping a coin $b \in \{0, 1\}$ and applying the signing algorithm : $\sigma^\star = \mathsf{SMDVS.Sign}(m^\star, a_b, P_{B_1}, \ldots, P_{B_n})$. This signature is given to $\mathcal{A}$ which outputs a bit $b^\star$ at the end of the `guess` stage. In both stages, the adversary has a permanent access to the random oracle $\mathcal{H}$, the signing oracles Σ_0 and Σ_1, and the verifying oracle Υ. We denote $q_{\mathcal{H}}$, q_{Σ_0}, q_{Σ_1} and q_Υ the number of queries to the corresponding oracles. We set $q_\Sigma = q_{\Sigma_0} + q_{\Sigma_1}$. To simulate the environment of the adversary $\mathcal{A}$, $\tilde{\mathcal{D}}$ proceeds as follows :

- it picks a_0 and a_1 in $[\![1, q-1]\!]$ at random and the signer's key is defined as $P_{A_0} = a_0 P$ and $P_{A_1} = a_1 P$,
- it chooses $s_i \in [\![1, q-1]\!]$ for $i \in [\![1, n]\!]$ and replaces each P_{B_i} by $s_i R_x$,
- the random oracle is simulated by maintaining an H-List in a classical way;
- the signing oracles Σ_0 and Σ_1 can be perfectly simulated thanks to the knowledge of a_0 and a_1;
- to simulate the verifying oracle, once a signature $\sigma = (Q_A, Q_B, Y_{B_1}, \ldots, Y_{B_n})$ on m is queried to Υ, $\tilde{\mathcal{D}}$ browses the H-List looking for all $n+2$-tuples of the form $(m, P_{B_1}, \ldots, P_{B_n}, R)$ and tests whether $e(R, P_{B_1}) = e(Y_{B_1}, P)$. If

such is the case, it verifies, for $j \in [\![2, n]\!]$, that $e(R, P_{B_j}) = e(Y_{B_j}, P)$. Then he can compute $M = H(m, P_A, P_{B_1}, \ldots, P_{B_n}, R)$ and outputs `Valid` if and only if $e(M, P) = e(Q_A, P_A)e(Q_B, P_B)$.

For the challenge simulation, $\tilde{\mathcal{D}}$ picks $b \in \{0, 1\}$ at random,

- it computes $Y^\star_{B_i} = s_i R_{xy}$ for $i \in [\![1, n]\!]$
- it picks at random $M^\star \in \mathbb{G}$
- it picks at random $r^\star \in [\![1, q-1]\!]$, computes $Q^\star_B = r^\star P$ and $Q^\star_A = a_b^{-1}(M^\star - r^\star P_B)$.

The simulated challenger outputs $\sigma^\star = (Q^\star_A, Q^\star_B, Y^\star_{B_1}, \ldots, Y^\star_{B_n})$ without updating the $\mathsf{H\text{-}List}$ with $M^\star$.

Eventually, when the adversary $\mathcal{A}$ stops outputing a bit $b^\star$, the algorithm $\tilde{\mathcal{D}}$ browses the $\mathsf{H\text{-}List}$ looking for all $n+2$-tuples of the form $(m^\star, P_{B_1}, \ldots, P_{B_n}, R)$ and tests whether $e(R, R_x) = e(R_{xy}, P)$. If such is the case, it outputs this point R, else it outputs a random element from $\mathbb{G}$. Let $\tilde{\tau}$ and $\tilde{\varepsilon}$ denote the running time of $\tilde{\mathcal{D}}$ and the success for $\tilde{\mathcal{D}}$ to solve the intermediate problem. Clearly $\tilde{\tau}$ is upper-bounded by $\tau + (q_{\mathcal{H}} + O(1))(q_\Upsilon + O(1))T_P$ where T_P denotes the time complexity to evaluate a pairing.

The previous simulation is perfectly indistinguishable from the real game unless

(1) a valid signature $\sigma = (Q_A, Q_B, Y_{B_1}, \ldots, Y_{B_n})$ on m is queried to Υ during the simulation whereas $(m, P_A, P_{B_1}, \ldots, P_{B_n}, R)$ was not queried to $\mathcal{H}$ before. This happens with probability at most $q_\Upsilon 2^{-k}$.

(2) $(m^\star, P_{B_1}, \ldots, P_{B_n}, Y^\star)$ where $Y^\star = yP$ is queried from $\mathcal{H}$ by the signing oracle, the verifying oracle or the adversary. The first case happens with probability at most $q_\Sigma 2^{-k}$, and by definition of $\mathsf{PSI\text{-}CMA}$ security, the second case cannot occur, otherwise, the verifying query would be the challenge signature. The probability that $(m^\star, P_{B_1}, \ldots, P_{B_n}, Y^\star)$ is queried from $\mathcal{H}$ by the adversary, is upper-bounded by $\tilde{\varepsilon}$. The challenge signature gives $\mathcal{A}$ no information about b if $(m^\star, P_{B_1}, \ldots, P_{B_n}, Y^\star)$ is not queried from $\mathcal{H}$, therefore in this case it succeeds with probability $1/2$.

Thus we get

$$\begin{aligned}
\frac{\varepsilon}{2} &= \left| \frac{\mathbf{Adv}^{\mathsf{psi-cma}}_{\mathsf{SMDVS},\mathcal{A}}(k) + 1}{2} - \frac{1}{2} \right| \\
&= \Big| \Pr[\mathcal{A} \text{ succeeds in the real game}] - \Pr[\mathcal{A} \text{ succeeds in } \tilde{\mathcal{D}}\text{'s simulation}] \Big| \\
&\leq \tilde{\varepsilon} + \frac{q_\Sigma + q_\Upsilon}{2^k}, \text{ and the theorem follows.}
\end{aligned}$$

$\square$

5 Conclusion

We designed the first n-designated verifiers signature scheme for an arbitrary $n \in \mathbb{N}$ protecting the signer's anonymity without encrypting the signature. Its security relies on a very classical assumption (CDH) in the random oracle model. The signature length is $n + 2$ points on an elliptic curve, and the verification is quite time consuming. However, the signature generation is efficient. Contructing an MDVS protocol protecting the signer's anonymity with constant signature length remains an open problem .

Ackowledgements We thankfully acknowledge Raghav Bhaskar for his talk at Caen's cryptography seminar which has motivated this research.

References

[1] D. Boneh, M. Franklin: Identity-based Encryption from the Weil Pairing. SIAM J. Computing, 32(3), 586–615 (2003).

[2] D. Boneh, C. Gentry, B. Lynn, H. Shacham: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. Proc of Eurocrypt'03, Springer LNCS Vol. 2656, 416–432 (2003)

[3] M. Bellare, P. Rogaway: Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. Proc. of 1st ACM Conference on Computer and Communications Security, 62–73 (1993)

[4] D. Chaum: Private Signature and Proof Systems. US Patent 5,493,614 (1996)

[5] Y. Desmedt: Verifier-Designated Signatures, Rump Session, Crypto'03 (2003)

[6] M. Jakobsson, K. Sako, R. Impagliazzo: Designated Verifier Proofs and their Applications. Proc. of Eurocrypt'96, Springer LNCS Vol. 1070, 142–154 (1996)

[7] A. Joux: A One Round Protocol for Tripartite Diffie–Hellman. Proc. of ANTS IV, Springer LNCS Vol. 1838, 385–394 (2000)

[8] F. Laguillaumie, D. Vergnaud: Designated Verifier Signature: Anonymity and Efficient Construction from *any* Bilinear Map. Proc. of SCN 2004, Springer LNCS, to appear.

[9] F. Laguillaumie, D. Vergnaud: Multi-designated Verifiers Signatures. Proc. of ICICS 2004, Springer LNCS Vol. 3269, 495–507 (2004)

[10] A. Menezes, T. Okamoto, S. Vanstone: Reducing elliptic curve logarithms to logarithms in finite fields. Proc. of STOC 1991, 80–89 (1991)

[11] R. L. Rivest, A. Shamir, Y. Tauman: How to Leak a Secret. Proc. of Asiacrypt'01, Springer LNCS Vol. 2248, 552–565 (2001)