ElGamal type signature schemes for n-dimensional vector spaces

Iwan M. Duursma[†] and Seung Kook Park[†]

Abstract

We generalize the ElGamal signature scheme for cyclic groups to a signature scheme for n-dimensional vector spaces. The higher dimensional version is based on the untractability of the vector decomposition problem (VDP). Yoshida has shown that under certain conditions, the VDP on a two-dimensional vector space is at least as hard as the computational Diffie-Hellman problem (CDHP) on a one-dimensional subspace.

1 Introduction

Intractable mathematical problems such as the integer factorization problem, the discrete logarithm problem (DLP), and the computational Diffie-Hellman problem (CDHP) are being used to provide secure protocols for cryptosystems. A new hard problem which is called the vector decomposition problem (VDP) was proposed in [8]. Yoshida [7] states the conditions that are required for the VDP on a two-dimensional vector space to be at least as hard as the CDHP on a one-dimensional subspace. The VDP on a two-dimensional vector space can serve as the underlying intractable problem for cryptographic protocols but the only protocol presented so far that uses the VDP is watermarking [7]. In this paper we present a signature scheme based on VDP and we generalize the ElGamal signature scheme for cyclic groups to a signature scheme for n-dimensional vector spaces.

Algorithms of the generalized ElGamal signature scheme, DSA and ECDSA are given in Section 2. In Section 3 we state the definitions of CDHP and VDP. Yoshida's conditions for the VDP on a two-dimensional vector space to be at least as hard as the CDHP on a one-dimensional subspace are stated without proof. Examples of the VDP are given. In Section 4 we present a signature scheme based on the VDP using hyperelliptic curves. In Section 5 we present an ElGamal type signature scheme for n-dimensional vector spaces and compare it with the signature schemes of Section 2 and Section 4.

[†]Department of Mathematics, University of Illinois at Urbana-Champaign (duursma@math.uiuc.edu, skpark@uiuc.edu)

2 The ElGamal signature scheme

In this paper, we construct a signature scheme which generalizes the ElGamal signature scheme to higher dimensional vector spaces. To do this, we need to characterize the ElGamal signature scheme and the two variations DSA and ECDSA. In this section, we state the generalized ElGamal signature scheme [5], [6] and describe how it generalizes the classical ElGamal signature scheme [4], [6]. We also state the two variations of the classical ElGamal signature scheme DSA [1], [6] and ECDSA [2]. Then, we compare and analyze the signature schemes in detail.

The ElGamal signature scheme is based on the difficulty of the discrete logarithm problem (DLP). That is, given group elements g and $h = g^a$ it is hard to compute a. Although there are various protocols of the ElGamal signature scheme, they all share a common idea:

Let m be a message. Then the signature of m is a pair (r, s) such that $ks + dr = m$ or $s = k^{-1}(m - dr)$, where $r = \alpha^k$, $y = \alpha^d$ for a random k and d that are chosen and kept secret by the signer. To verify the signature we compute $r^s y^r$ and α^m then check if $r^s y^r = \alpha^m$. This works since for a valid signature $r^s y^r = (\alpha^k)^s(\alpha^d)^r = \alpha^{ks+dr} = \alpha^m$. Now, we give the algorithm of the generalized ElGamal signature scheme [5], [6]:

Algorithm of the generalized ElGamal signature scheme

(a) Each entity A does the following:

1. Select an appropriate cyclic group G of order n, with generator α. (Assume that G is written multiplicatively.)

2. Select a random secret integer d, $1 \leq d \leq n - 1$. Compute the group element $y = \alpha^d$.

3. A's public key is (α, y), together with a description of how to multiply elements in G; A's private key is d.

Let $h : \{0, 1\}^* \longrightarrow \mathbb{Z}_n$ be a hash function where n is the number of elements in G. In general $h : \{0, 1\}^* \longrightarrow \mathbb{Z}_n$ will be the composition of an iterated hash function from $\{0, 1\}^*$ to $\{0, 1\}^N$ followed by an encoding from $\{0, 1\}^N$ to $\mathbb{Z}_n$.

(b) To sign a message m, A does the following;

1. Select a random secret integer k, $1 \leq k \leq n - 1$ with $\gcd(k, n) = 1$.

2. Compute the group element $r = \alpha^k$.

3. Compute $k^{-1} \bmod n$.

4. Define a function $\phi : G \longrightarrow \{0, 1\}^*$. For ease of notation write $\tilde{h}(r)$ instead of $h(\phi(r))$ for $r \in G$.

5. Compute $h(m)$ and $\tilde{h}(r)$.

6. Compute $s = k^{-1}\{h(m) - d\tilde{h}(r)\} \bmod n$.

7. A's signature for m is the pair (r, s).

(c) To verify A's signature (r, s) on m, B should do the following:

1. Obtain A's authentic public key (α, y).

2. Compute $h(m)$ and $\tilde{h}(r)$.

3. Compute $v_1 = y^{\tilde{h}(r)} r^s$.

4. Compute $v_2 = \alpha^{h(m)}$.

5. Accept the signature if and only if $v_1 = v_2$.

The classical ElGamal signature scheme is a special case of the generalized ElGamal signature scheme with G being the multiplicative group $\mathbb{Z}_p^*$ and $\tilde{h} : \mathbb{Z}_p^* \longrightarrow \mathbb{Z}/(p-1)\mathbb{Z}$ being defined as $r \ (\bmod p) \longmapsto r \ (\bmod p-1)$. One point to mention is that in the classical ElGamal signature scheme there is an extra process of checking that $1 \leq r \leq p-1$ in the verification step. Note that, in both of the algorithms, the verification is done in the group. That is, we verify if $v_1 = v_2$ as group elements. Algorithms of the two variations of the classical ElGamal signature scheme DSA [1], [6] and ECDSA [2]are given below:

Algorithm of the DSA

(a) Each entity A does the following :

1. Select a prime number q such that $2^{159} < q < 2^{160}$.

2. Choose t so that $0 \leq t \leq 8$, and select a prime number p where $2^{511+64t} < p < 2^{512+64t}$, with the property that q divides $(p-1)$.

3. Select a generator α of the unique cyclic group of order q in $\mathbb{Z}_p^*$ by choosing an element $g \in \mathbb{Z}_p^*$ and then computing $\alpha = g^{(p-1)/q} \bmod p$ until $\alpha \neq 1$.

4. Select a random integer d such that $1 \leq d \leq q-1$.

5. Compute $y = \alpha^d \bmod p$.

6. A's public key is (p, q, α, y); A's private key is d.

(b) To sign a message m, A does the following :

1. Select a random secret integer k, $0 < k < q$.

2. Compute $r = (\alpha^k \bmod p) \bmod q$.

3. Compute $k^{-1} \bmod q$.

4. Compute $s = k^{-1}\{h(m) + dr\} \bmod q$, where h is the Secure Hash Algorithm.

5. A's signature for m is the pair (r, s).

(c) To verify A's signature (r, s) on m, B should do the following :

1. Obtain A's authentic public key (p, q, α, y).
2. Verify that $0 < r < q$ and $0 < s < q$; if not, then reject the signature.
3. Compute $w = s^{-1} \bmod q$ and $h(m)$.
4. Compute $u_1 = w \cdot h(m) \bmod q$ and $u_2 = rw \bmod q$.
5. Compute $v = (\alpha^{u_1} y^{u_2} \bmod p) \bmod q$.
6. Accept the signature if and only if $v = r$.

The significant difference between the classical ElGamal signature scheme and the DSA is in the verification process. In the generalized or classical ElGamal scheme, we compute $r \in G$ as a group element and send r as a part of the signature. Thus we reveal the group element r. To verify the signature, we compute $y^{\tilde{h}(r)} r^s$ and $\alpha^{h(m)}$ in the group G and check if they are same as group elements. But in the DSA, we take the group element $\alpha^k \bmod p \in \mathbb{Z}_p^*$ and take the remainder modulo q. That is, $r = (\alpha^k \bmod p) \bmod q$. Therefore, r is not an element of the group $\mathbb{Z}_p^*$. In the verification process, we compute $\alpha^{u_1} y^{u_2} \pmod{p}$ $(= \alpha^{s^{-1}h(m) + s^{-1}dr} \pmod{p} = \alpha^k \pmod{p})$ and take the remainder modulo q. Then we check if the outcome equals r. We compare the two elements $(\alpha^{u_1} y^{u_2} \bmod p) \bmod q$ and r in $\mathbb{Z}/q\mathbb{Z}$ not in the group $\mathbb{Z}_p^*$. To summarize, in the classical case, we need to compute $v_1 = y^r r^s \pmod{p}$ and $v_2 = \alpha^{h(m)} \pmod{p}$ to verify the signature, which requires three modular exponentiations. But in the DSA we just need to compute $v = (\alpha^{u_1} y^{u_2} \bmod p) \bmod q$ for verification, using only two modular exponentiations. Since DSA has a computational advantage in the verification process, we will use this feature with a modification in our construction for higher dimensional ElGamal signature schemes. One other difference is the signs in the signature s. In the classical case $s = k^{-1}\{h(m) - d\tilde{h}(r)\} \pmod{p-1}$ and in the DSA $s = k^{-1}\{h(m) + dr\} \pmod{q}$. The sign appears as "+" in the DSA because of the modification in the verification process, but it is not essential.

The ECDSA is the elliptic curve version of DSA. That is, instead of working in a group of order q in $\mathbb{Z}_p^*$, we work in a group of order n in $E(\mathbb{Z}_p)$.

Algorithm of the ECDSA

(a) Each entity A does the following:

1. Select an elliptic curve E defined over $\mathbb{Z}_p$. The number of points in $E(\mathbb{Z}_p)$ should be divisible by a large prime n.
2. Select a point $P \in E(\mathbb{Z}_p)$ of order n.
3. Select a random integer d, $2 \le d \le n - 2$.

4. Compute $Q = dP$.
5. A's public key is (E, P, n, Q); A's private key is d.

(b) To sign a message m, A does the following :

1. Select a random integer k, $2 \leq k \leq n-2$.
2. Compute $kP = (x_1, y_1)$ and $r = x_1 \bmod n$. If $r = 0$, then go to step 1.
3. Compute $k^{-1} \bmod n$.
4. Compute $s = k^{-1}\{h(m) + dr\} \bmod n$, where h is the Secure Hash Algorithm. If $s = 0$, then go to step 1.
5. The signature for the message m is the pair of integers (r, s).

(c) To verify A's signature (r, s) on m, B should do the following :

1. Obtain an authentic copy of A's public key (E, P, n, Q). Verify that r and s are integers in the interval $[1, n-1]$.
2. Compute $w = s^{-1} \bmod n$ and $h(m)$.
3. Compute $u_1 = h(m)w \bmod n$ and $u_2 = rw \bmod n$.
4. Compute $u_1P + u_2Q = (x_0, y_0)$ and $v = x_0 \bmod n$.
5. Accept the signature if and only if $v = r$.

The main difference between DSA and ECDSA is in the computation of r. In DSA, r is computed by selecting a random k and computing $\alpha^k \pmod p$ and then reducing it modulo q. In ECDSA, we select a random k and compute the point kP. Then take the x-coordinate of the point kP and reduce it modulo n. As it was with DSA, the element (or point) $kP = (x_1, y_1)$ of the group $E(\mathbb{Z}_p)$ is not revealed. Here, $r = x_1 \bmod n$ is given as a part of the signature. Since we do not know kP, the final verification is not done in $E(\mathbb{Z}_p)$ but in $\mathbb{Z}/n\mathbb{Z}$.

3 The vector decomposition problem

The ElGamal signature scheme is based on the difficulty of the discrete logarithm problem (DLP). We will discuss a different problem called the vector decomposition problem (VDP). We state the definition of the VDP and the conditions for the VDP to be a hard problem. Examples of the VDP using elliptic curves by Yoshida [7] and hyperelliptic curves by Duursma and Kiyavash [3] are presented.

Definition 3.1. *The Vector Decomposition Problem on $\mathcal{V}$ (a two-dimensional vector space over $\mathbb{F}$) is " Given e_1, e_2, $v \in \mathcal{V}$ such that e_1, e_2 is an $\mathbb{F}$-basis for $\mathcal{V}$, find the vector $u \in \mathcal{V}$ such that $u \in \langle e_1 \rangle$ and $v - u \in \langle e_2 \rangle$".*

Definition 3.2. *The computational Diffie-Hellman problem on $\mathcal{V}'$ (a one-dimensional vector space over $\mathbb{F}$) is " Given $e \in \mathcal{V}' \setminus \{0\}$ and $ae,\ be \in \langle e \rangle$, find $abe \in \langle e \rangle$".*

Theorem 3.3. (Yoshida [7]) *The Vector Decomposition Problem on $\mathcal{V}$ is at least as hard as the computational Diffie-Hellman problem on $\mathcal{V}' \subset \mathcal{V}$ if for any $e \in \mathcal{V}'$ there are linear isomorphisms $\phi_e,\ F_e\ :\ \mathcal{V} \to \mathcal{V}$ which satisfy the following three conditions:*

(1) For any $v \in \mathcal{V}$, $\phi_e(v)$ and $F_e(v)$ are effectively defined and can be computed in polynomial time.

(2) $e,\ \phi_e(e)$ is an $\mathbb{F}$-basis for $\mathcal{V}$.

(3) There are $\alpha_1,\ \alpha_2,\ \alpha_3 \in \mathbb{F}$ with

$$\begin{aligned} F_e(e) &= \alpha_1 e, \\ F_e(\phi_e(e)) &= \alpha_2 e + \alpha_3 \phi_e(e), \end{aligned}$$

and $\alpha_1,\ \alpha_2,\ \alpha_3 \neq 0$. The elements $\alpha_1,\ \alpha_2,\ \alpha_3$ and their inverses can be computed in polynomial time.

Proof. The proof is in [7] and is also included in [3] □

Example 3.4. *(Example of Yoshida [7]) We choose $\mathcal{V} = E[n]$, the full group of n-torsion points on an elliptic curve, and $\mathcal{V}' = E(\mathbb{F}_p) \cap E[n]$, the subgroup of $\mathbb{F}_p$-rational n-torsion points, where*

$$\begin{aligned} &p\ :\ \textit{a prime with } p \equiv 2\ (mod\ 3), \\ &E\ :\ y^2 = x^3 + 1,\ \textit{an elliptic curve over } \mathbb{F}_p, \\ &n\ :\ \textit{a prime such that } 6n = p + 1, \\ &E[n] = \{P \in E \mid nP = 0\} \subset E(\mathbb{F}_{p^2}). \end{aligned}$$

Let $F : (x, y) \longmapsto (x^p, y^p)$ be the Frobenius map and let $\phi : (x, y) \longmapsto (\omega x, y)$, where $\omega^2 + \omega + 1 = 0$. Then Theorem 3.3 applies with $\alpha_1 = 1,\ \alpha_2 = -1,\ \alpha_3 = -1$.

By Theorem 3.3, the VDP is hard if the CDHP on a one-dimensional subspace is hard. The curve $E\ :\ y^2 = x^3 + 1$ in Example 3.4 is supersingular. Thus the ECDLP and hence the CDHP on the one-dimensional subspace is vulnerable to the MOV attack. Duursma and Kiyavash [3] showed that any elliptic curve that satisfies the conditions of Theorem 3.3 is supersingular. Thus, using the VDP with the full n-torsion points on an elliptic curve introduces a vulnerability that needs to be compensated by choosing larger parameters. To avoid this, the VDP may be used with higher genus curves.

Example 3.5. *(Example of Duursma and Kiyavash [3]) The Jacobian of the hyperelliptic curve*

$$C\ :\ y^2 = x^6 - ax^3 + 1,\ \textit{where } a \in \mathbb{F}_p \textit{ for an odd prime } p$$

is isogenous to a product of elliptic curves $E_1 \times E_2$, where

$$\begin{aligned} E_1 &:\ y^2 = x^3 + (3x + 2 + a)^2, \\ E_2 &:\ y^2 = x^3 + (3x + 2 - a)^2. \end{aligned}$$

The curves E_1 and E_2 are 3-isogenous over $\mathbb{F}_{p^2}$ with j-invariants

$$\begin{aligned} j_1 &= 4 \cdot 1728 \frac{(5+2a)^3}{(2+a)(2-a)^3}, \\ j_2 &= 4 \cdot 1728 \frac{(5-2a)^3}{(2-a)(2+a)^3}. \end{aligned}$$

We choose $C : y^2 = x^6 - ax^3 + 1$ such that E_1 has a large cyclic subgroup $\mathbb{Z}/n\mathbb{Z}$ of rational points over $\mathbb{F}_p$, for $p \equiv 2 (mod\ 3)$. Then we choose as two-dimensional vector space $\mathcal{V}$ the n-torsion $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ in the Jacobian of the hyperelliptic curve C over the extension field $\mathbb{F}_{p^2}$ and choose as one-dimensional subspace $\mathcal{V}'$ the subspace $\mathbb{Z}/n\mathbb{Z}$ of $\mathcal{V}$ that is rational over $\mathbb{F}_p$.

$$\begin{aligned} &p \ :\ \text{an odd prime with } p \equiv 2 \ (mod\ 3), \\ &C \ :\ y^2 = x^6 - ax^3 + 1, \ \text{a curve with } a \in \mathbb{F}_p, \\ &Jac(C) \ :\ \text{Jacobian of the curve } C, \\ &n \ :\ \text{a prime greater than 3}, \\ &\mathcal{V} = \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} \subset Jac(C)(\mathbb{F}_{p^2}), \\ &\mathcal{V}' = \mathbb{Z}/n\mathbb{Z} \subset Jac(C)(\mathbb{F}_p). \end{aligned}$$

Let $F : (x, y) \longmapsto (x^p, y^p)$ be the Frobenius map and let $\phi : (x, y) \longmapsto (\omega x, y)$, where $\omega^2 + \omega + 1 = 0$. Then Theorem 3.3 applies with $\alpha_1 = 1,\ \alpha_2 = -1,\ \alpha_3 = -1$.

4 An ElGamal type signature scheme for two- dimensional vector spaces

The VDP on a hyperelliptic curve can serve as the underlying intractable problem for cryptographic protocols. Other problems such as integer factorization and the discrete logarithm problem have been studied for cryptography for many years. Many protocols have been formulated for each of those problems. The only protocol presented so far that uses the VDP is for watermarking [7]. In this section we introduce a signature scheme based on VDP. The infinite family of genus 2 hyperelliptic curves of Example 3.5 is used in the signature scheme. The signature scheme consists of three parts: key generation, signature generation, and verification. The ingredients of the algorithm are the following: Let V be a vector space with basis $\{e_1, e_2\}$. The signer chooses randomly a new basis $(Q_1,\ Q_2)^T \ =\ \mathcal{D}(e_1,\ e_2)^T$, where

$$\mathcal{D} = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix} : \ \text{private key matrix}$$

.

For the signature scheme to be secure it is necessary that the transformation matrix $\mathcal{D}$ is not easily obtained from the given basis $\{e_1, e_2\}$ and $\{Q_1, Q_2\}$. Let $M = (m_1, m_2)$ be a message divided into two parts m_1 and m_2. A signature for M is a pair $(\mathcal{R}, \mathcal{S})$ such that

$$\mathcal{S}(\mathcal{H}(M) + \mathcal{R}\mathcal{D}) = \mathcal{K},$$

where

$$\mathcal{H}(M) = \begin{pmatrix} h(m_1) & h(m_2) \\ h(m_1) & h(m_2) \end{pmatrix} \ : \text{ hashed message matrix}$$
$$\mathcal{K} = (k_1 \ k_2) \ : \text{ random matrix}$$
$$\mathcal{R} = \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix} \ : \text{ signature matrix such that } (r_1, r_2) = \Psi(k_1e_1 + k_2e_2)$$
$$\mathcal{S} = (s_1 \ s_2) \ : \text{ signature matrix.}$$

Clearly, provided that $\mathcal{H}(M)+\mathcal{RD}$ is nonsingular, the signature $(\mathcal{R}, \mathcal{S})$ can be generated efficiently by the signer. In order to verify the signature, we compute

$$\Psi \ (\ (s_1 + s_2)(h(m_1)e_1 + h(m_2)e_2) + s_1r_1Q_1 + s_2r_2Q_2 \ )$$
$$(\quad = \Psi \ (\ \mathcal{S}(\mathcal{H}(M) + \mathcal{RD})(e_1, e_2)^T \ ) \ \ )$$

and check that it is equal to (r_1, r_2) ($=$ Ψ ($\mathcal{K}(e_1, e_2)^T$)). Now, we introduce the algorithm of the VDP signature scheme.

Algorithm of the VDP signature scheme (Algorithm A)

(a) Each entity A does the following :

1. Select a hyperelliptic curve C from Example 3.5 and choose as two-dimensional vector space $\mathcal{V}$ the n-torsion $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ in the Jacobian of the hyperelliptic curve C over the extension field $\mathbb{F}_{p^2}$ of $\mathbb{F}_p$ for a large enough prime n.

2. Select a basis $\{e_1, e_2\}$ for $\mathcal{V}$, where $e_1 \in \mathbb{F}_p$.

3. Select $d_{11}, d_{12}, d_{21}, d_{22} \in \mathbb{Z}/n\mathbb{Z}$ such that

$$det \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix} \neq 0 \pmod{n}.$$

4. Compute $Q_1 = d_{11}e_1 + d_{12}e_2, \ Q_2 = d_{21}e_1 + d_{22}e_2$.

5. A's public key is $\ C, \ n, \ (e_1, e_2), \ (Q_1, Q_2)$.
   A's private key is $\ (d_{11}, d_{12}), \ (d_{21}, d_{22})$.

(b) To sign a message $M = (\ m_1, m_2 \ )$, A does the following :
Let $h : \{0, 1\}^* \longrightarrow \mathbb{Z}/n\mathbb{Z}$ be a hash function and for notational convenience let $h(M) = h(m_1)e_1 + h(m_2)e_2$.

1. Select $k_1, k_2 \in \mathbb{Z}/n\mathbb{Z}$.

2. Compute $K = k_1e_1 + k_2e_2$.

3. Express K in Mumford representation, $K = (x^2 + u_1 x + u_2, \cdots)$, $u_1, u_2 \in \mathbb{F}_p^2$. Since $p \equiv 2(\mod 3)$, we have $\mathbb{F}_{p^2} \cong \mathbb{F}_p[x]/\langle x^2 + x + 1\rangle$. Thus, for $u_i \in \mathbb{F}_{p^2}$, there exist $u_{i1}, u_{i2} \in \mathbb{F}_p$ such that $u_i \longmapsto u_{i1} + u_{i2}x$, $i = 1, 2$. Let $r_i = u_{i1} + u_{i2}p \pmod n$. Hence, for each $K \in \mathcal{V}$, we can assign an ordered pair (r_1, r_2). We will call this function Ψ. If $K = (x + u_1, \cdots)$ or $K = (1, 0)$ then return to step 1.

4. Compute $s_1, s_2 \in \mathbb{Z}/n\mathbb{Z}$ that satisfy the following :

$$(s_1 + s_2)h(M) + s_1 r_1 Q_1 + s_2 r_2 Q_2 = K, \quad \text{where } \Psi(K) = (r_1, r_2),$$

$$\text{that is} \begin{cases} s_1 h(m_1) + s_2 h(m_1) + s_1 r_1 d_{11} + s_2 r_2 d_{21} = k_1 \pmod n, \\ s_1 h(m_2) + s_2 h(m_2) + s_1 r_1 d_{12} + s_2 r_2 d_{22} = k_2 \pmod n, \end{cases}$$

$$\text{or} \quad (s_1\ s_2) \left[\begin{pmatrix} h(m_1) & h(m_2) \\ h(m_1) & h(m_2) \end{pmatrix} + \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix} \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix} \right] = (k_1\ k_2) \pmod n.$$

If $\begin{pmatrix} h(m_1) & h(m_2) \\ h(m_1) & h(m_2) \end{pmatrix} + \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix} \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}$ is singular, then go to step 1.

5. The signature for the message M is (s_1, s_2), (r_1, r_2).

(c) To verify A's signature, B should do the following .

1. Obtain A's public key C, n, (e_1, e_2), (Q_1, Q_2).

2. Compute $(s_1 + s_2)h(M) + s_1 r_1 Q_1 + s_2 r_2 Q_2$.

3. Accept the signature if and only if

$$\Psi((s_1 + s_2)h(M) + s_1 r_1 Q_1 + s_2 r_2 Q_2) = (r_1, r_2).$$

(Security aspects of Algorithm A)

1. A necessary condition for the security of the signature scheme is that given $Q_1 = d_{11}e_1 + d_{12}e_2$ and $Q_2 = d_{21}e_1 + d_{22}e_2$ it is hard to compute the coefficientsd_{11}, d_{12}, d_{21}, and d_{22}. This is certainly the case when the VDP, which asks to compute ae_1 and be_2 for a given $ae_1 + be_2$, is hard. Even if the VDP is solved, it remains to solve four instances of the DLP: given $d_{11}e_1$, find d_{11}, etc. Thus, a direct attack on the private key is at least as hard as solving both the VDP and the DLP. It is not clear how solving one of the two problems could be used to solve the other one.

2. It is also necessary that given $K = k_1 e_1 + k_2 e_2$ it is hard to compute the coefficients k_1 and k_2. An attacker with knowledge of k_1 or k_2 can use the equations under (b, item 4) to reduce the key space for the private key. In fact, an attacker that intercepts two messages and knows k_1 and k_2 for each of the two messages will be able to recover the private key completely.

3. An attacker trying to sign a message may start with choosing t, t_1, t_2 and computing $K = th(M) + t_1 Q_1 + t_2 Q_2$. However, after computing r_1, r_2 from K, the attacker then faces three equations $t = s_1 + s_2$, $t_1 = s_1 r_1$, $t_2 = s_2 r_2$ for the two unknowns s_1, s_2.

5 An ElGamal type signature scheme for n-dimensional vector spaces

We present an ElGamal type signature scheme for n-dimensional vector spaces and explain that it contains the signature schemes of Section 2 and Section 4 as special cases. The main idea is the following:
Let V be an n-dimensional vector space with basis $\{e_1, \cdots, e_n\}$. The signer chooses randomly a new basis $(Q_1, \cdots, Q_2)^T = \mathcal{D}(e_1, \cdots, e_2)^T$, where

$$\mathcal{D} = \begin{pmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & \ddots & \vdots \\ d_{n1} & \cdots & d_{nn} \end{pmatrix} : \ n \times n \text{ private key matrix.}$$

Let $M = (m_1, \cdots, m_n)$ be a message divided into n parts. A signature for M is a pair $(\mathcal{R}, \mathcal{S})$ such that

$$\mathcal{S}(\mathcal{H}(M) + \mathcal{R}\mathcal{D}) = \mathcal{K},$$

where

$$\mathcal{H}(M) = \begin{pmatrix} h(m_1) & \cdots & h(m_n) \\ \vdots & \ddots & \vdots \\ h(m_1) & \cdots & h(m_n) \end{pmatrix} \ : \ n \times n \ \text{ hashed message matrix}$$

$$\mathcal{K} = (k_1 \ \cdots \ k_n) \ : \ 1 \times n \ \text{ random matrix}$$

$$\mathcal{R} = \begin{pmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_n \end{pmatrix} \ : \ n \times n \ \text{ diagonal signature matrix}$$

$$\mathcal{S} = (s_1 \ \cdots \ s_n) \ : \ 1 \times n \ \text{ signature matrix.}$$

In order to verify the signature, we compute

$$f \ (\ (s_1 + \cdots + s_n)(h(m_1)e_1 + \cdots + h(m_2)e_2) + r_1 s_1 Q_1 + \cdots + r_n s_n Q_n \ )$$
$$(\ \ = f \ (\ \mathcal{S}(\mathcal{H}(M) + \mathcal{R}\mathcal{D})(e_1, \cdots, e_2)^T \ ) \ \ )$$

and check that it is equal to $(r_1, \cdots, r_n)$ ($=$ f ($\mathcal{K}(e_1, \cdots, e_2)^T$)), for a specified f.

Algorithm of the n-dimensional ElGamal signature scheme (Algorithm B)

Let V be a vector space over $\mathbb{F}$, where $\mathbb{F}$ is a field.
Let f be a function

$$\begin{aligned} f : V &\longrightarrow \mathbb{F}^n \\ K &\longmapsto (r_1, \cdots, r_n) \end{aligned}$$

which is easy to compute and, for each $(r_1,\cdots,r_n)$, $f^{-1}((r_1,\cdots,r_n))$ is small.

(a) Each entity A does the following :

1. Select a basis $\{e_1,\cdots,e_n\}$ for V.
2. Select $(d_{11},\cdots,d_{1n}),\cdots,(d_{n1},\cdots,d_{nn})\in\mathbb{F}^n$ such that

$$det\begin{pmatrix} d_{11} & \cdots & d_{1n} \\ \vdots & & \vdots \\ d_{n1} & \cdots & d_{nn}\end{pmatrix}\neq 0.$$

3. Compute

$$\begin{aligned}&Q_1=d_{11}e_1+\cdots+d_{1n}e_n\\ &\vdots\\ &Q_n=d_{n1}e_1+\cdots+d_{nn}e_n\end{aligned}$$

4. A's public key is $V,\ \mathbb{F},\ (e_1,\cdots,e_n),\ (Q_1,\cdots,Q_n)$.
   A's private key is $(d_{11},\cdots,d_{1n}),\cdots,(d_{n1},\cdots,d_{nn})$.

(b) To sign a message $M=(\ m_1,\ \cdots,m_n\ )$, A does the following :

1. Select $(k_1,\cdots,k_n)\in\mathbb{F}^n$.
2. Compute $K=k_1e_1+\cdots+k_ne_n\in V$.
3. Compute $f(K)=(r_1,\cdots,r_n)$
4. Compute the matrix equation for $(s_1,\cdots,s_n)\in\mathbb{F}^n$

$$(s_1,\cdots,s_n)\left[\begin{pmatrix} h(m_1) & h(m_2) & \cdots & h(m_n)\\ h(m_1) & h(m_2) & \cdots & h(m_n)\\ \vdots & \vdots & & \vdots\\ h(m_1) & h(m_2) & \cdots & h(m_n)\end{pmatrix}+\begin{pmatrix} r_1 & 0 & \cdots & 0\\ 0 & r_2 & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & r_n\end{pmatrix}\begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n}\\ d_{21} & d_{22} & \cdots & d_{2n}\\ \vdots & \vdots & & \vdots\\ d_{n1} & d_{n2} & \cdots & d_{nn}\end{pmatrix}\right]$$
$$=(k_1,\cdots,k_n),\quad \text{where } h \text{ is a hash function.}$$

5. The signature for the message M is $(s_1,\cdots,s_n),\ (r_1,\cdots,r_n)$.

(c) To verify A's signature, B does the following :

1. Obtain A's public key.
2. Compute $v=(s_1+\cdots+s_n)h(M)+r_1s_1Q_1+\cdots+r_ns_nQ_n\in V$.
3. Compute $f(v)=(t_1,\cdots,t_n)\in\mathbb{F}^n$.

4. Accept the signature if and only if $(t_1, \cdots, t_n) = (r_1, \cdots, r_n)$.

We compare Algorithm B with the three versions of the ElGamal signature scheme in Section 2, and with the VDP signature scheme in Section 4.

(ECDSA)

We consider the n-dimensional ElGamal signature scheme with $n = 1$ and apply it to the signature schemes of Section 2. Let E be an elliptic curve defined over $\mathbb{Z}_p$ such that the number of points in $E(\mathbb{Z}_p)$ is divisible by a large prime n. We take V to be the cyclic group of order n generated by $P \in E(\mathbb{Z}_p)$ and define

$$\begin{array}{rcll} f : V & \longrightarrow & \mathbb{Z}_n & \text{by} \\ K & \longmapsto & r \equiv x_1 \pmod{n} & \text{for} \;\; K = kP = (x_1, y_1). \end{array}$$

Then V is a one-dimensional vector space over the field $\mathbb{Z}_n$ with basis $\{P\}$. We apply the above V and f to ECDSA. In the ECDSA, the signature s is $s = k^{-1}\{h(m) + dr\} \bmod n$. In Algorithm B, the signature s is such that $sh(M) + srQ = K$ that is $s = k\{h(m) + dr\}^{-1} \bmod n$ which is the inverse of the signature in ECDSA. They both require one modular inverse, two modular multiplications, and one addition. In the ECDSA, we compute $s^{-1}h(m)P + s^{-1}rQ$ to verify the signature and in Algorithm B, we compute $sh(M) + srQ = sh(m)P + srQ$ for verification. Thus in ECDSA, we need to compute s^{-1} but in Algorithm B, we do not require the computation of the inverse. Therefore, Algorithm B is more efficient than ECDSA.

(DSA)

We let V be the cyclic group of order q generated by α in $\mathbb{Z}_p^*$ and define

$$\begin{array}{rcl} f \; : \; V & \longrightarrow & \mathbb{Z}_q \quad \text{by} \\ \alpha^k & \longmapsto & (\alpha^k \bmod p) \bmod q \; . \\ (0 < k < q) & & \end{array}$$

Then V is a one-dimensional vector space over the field $\mathbb{Z}_q$ with basis $\{\alpha\}$. The rest of the argument is similar to the ECDSA.

(Generalized ElGamal signature scheme)

Let V be the cyclic group G of order n, with generator α. To apply Algorithm B we restrict n to be a prime number. Then V is a one-dimensional vector space over $\mathbb{Z}_n$ with basis $\{\alpha\}$. Define $f : V \longrightarrow \mathbb{Z}_n$ by $f = h \circ \phi$. The generalized ElGamal signature scheme uses $s = k^{-1}\{h(m) - dh(\phi(r))\} \bmod n$ as the signature and Algorithm B uses $s = k\{h(m) + dh(\phi(r))\}^{-1} \bmod n$. Thus in the signature generation process, the amount of computation needed are the same for both signature schemes. The main difference is in the verification process. In the former case we need to compute $y^{h(\phi(r))} \cdot r^s$ and $\alpha^{h(m)}$ but in the latter case we only need to compute $\alpha^{sh(m)} y^{sh(\phi(r))}$ for verification. Thus the former case requires three exponentiations whereas the latter case requires only two. Hence Algorithm B is more efficient.

(VDP signature scheme)

We consider Algorithm B with $n = 2$. If we let V be the n-torsion $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ of Example 3.5 and define $f : V \longrightarrow \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ by $f = \Psi$, then we have Algorithm A of Section 4.

References

[1] ANSI X9.30. Public Key Cryptography for the Financial Services Industry: Part I: The Digital Signature Algorithm (DSA). 1997.

[2] ANSI X9.62. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). 1999.

[3] Iwan Duursma and Negar Kiyavash. The vector decomposition problem for elliptic and hyperelliptic curves. *J. Ramanujan Math. Soc.*, 20(1):59–76, 2005.

[4] Taher Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, 31(4):469–472, 1985.

[5] P. Horster and H. Petersen. Verallgemeinerte elgamal signaturen. *Sicherheit in Informationssystemen, Proceedings der Fachtagung SIS'94*, pages 89–106, 1994. Verlag der Fachvereine Zürich.

[6] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1997. With a foreword by Ronald L. Rivest.

[7] M. Yoshida. Inseparable multiplex transmission using the pairing on elliptic curves and its application to watermarking. *In Proceedings of Fifth Conference on Algebraic Geometry, Number Theory, Coding Theory and Cryptography, Graduate School of Mathematical Sciences, University of Tokyo*, 2003.

[8] M. Yoshida, S. Mitsunari, and T. Fujiwara. Vector decomposition problem and the trapdoor inseparable multiplex transmission scheme based problem. *In Proceedings of Symposium on Cryptography and Information Security, SCIS'03*, 2003.