Paper

ID-Based Digital Signatures with Security Enhanced Approach

Jacek Pomykała

Abstract—In the paper the ID-based digital signatures with signer's protection in case of the private key compromising is investigated. The proposed protocols have two main ingredients. First is the application of the credential system for the suitable verification key approval. Second is the application of the subliminal channel together with the interactive generation of the secret key, to obtain the increased resistance of the system against the powerful adversary. The particular interest was turned towards the significance of the deniable encryption in creation of the corresponding protocols.

Keywords—cryptography, deniable encryption, ID-based schemes.

1. Introduction

The critical point joining the functionality of the digital signatures is the management and authentication of the corresponding public keys. The potential way of cheating of (certificated) public key causes the risk that the identity of the user may be stolen. The concept of the ID-based public key cryptography introduced by Shamir implied the significant simplification of the corresponding management and authentication process.

In this concept the role of public key has been replaced by the user identity on the network (user-ID) like e-mail address, phone number, etc. More precisely there is the secret key of the private key generator (PKG), called the master key that is involved in the creation of the entity's secret key $sk = sk(\mathrm{ID}, s)$, by means of some trapdoor function. In order to keep the consistency, the public key Ω of PKG (known by each entity) should be related to the secret key $sk(\mathrm{ID}, s)$, so that the proof of knowledge of $sk(\mathrm{ID}, s)$ could be checked by any verifier with the aid of Ω.

When comparing with the certificate based cryptosystems the elimination of the public key certificates results here with the evident drawback. It implies that PKG knows the user secret key $sk(\mathrm{ID}, s)$. Moreover, loosing the master key s would compromise the secret keys of all entities. This obstacle make favorable the application of the ID-based schemes only in the systems with intermediate level of security.

In this paper we will encounter this problem suggesting some further ideas towards enhancing the security of some ID-based signature schemes. The first idea is to include in the entity's secret key its private part k generated by the user. The corresponding public part K sent in advance to PKG allows him to approve the corresponding verification key $vk = vk(\mathrm{ID}, K)$ related to Ω. As a result, in the subsequent step the signer is able to compute the new secret key $sk = sk(\mathrm{ID}, s, k)$ related to the suitable verification key. Such general scheme is described in Section 6.

Many ID-based signature schemes use a random parameter r in the signature generation process. This admits to hide the suitable private value k in the corresponding pseudorandom value $r = r(m, k)$. Certainly the corresponding commitment $R = R(r, m, \mathrm{ID}, \Omega, K)$ depends implicitly on k. Recovering this dependence allows the suitable party T (sharing the key k with the signer) to read the hidden (in the subliminal channel) information, with the aid of some trapdoor information t (known only by T). Since R is a one way function of r, its extraction from R is rather unrealistic task. We will show that the one-wayness may be replaced by the suitable trapdoor function depending on the parameters k and t that allows to recover the hidden information by the trustee from the corresponding signature Sig.

The above idea can be further enhanced in order to protect the signer against quite powerful adversary (that forces the signer to unveil his secret key $sk = sk(\mathrm{ID}, s, k)$). We assume that $r = r(m, k, \rho)$ with a random value ρ and the value of k that can be verified only on the basis of some trapdoor information t known only by T. In case of attack the signer show the "fake" values k' and ρ' instead of k and ρ leading to the same value $r = r(m, k', \rho')$.

Therefore the signer, when forced to unveil the signature parameters is able to get persuaded the adversary to believe that all of them have a real-looking data. Concluding, any verifier (except T) is not able to distinguish between the random r and a pseudorandom $r(m, k, \rho)$ even if he knows the signer secret key $sk(\mathrm{ID}) = sk(\mathrm{ID}, s, k)$. The resulting signature is to be called the deniable signature.

To construct the suitable pseudorandom function, the notion of deniable encryption $r = E_{den}(h(m, k))$ is applied similarly as in [1], with h being the secure hash function. The corresponding trapdoor information t applied in the decryption algorithm D, allows the trustee to check if $D_t(r) = h(m, k)$. On the other hand, even the signer (who does not known the trapdoor value t) is not able to find the evidence if a given pseudorandom value has actually the form $r = E_{den}(h(m, k))$. Hence the deniable signature protects the signer against the coercion attack when the adversary demands him to unveil the secret key $sk = sk(\mathrm{ID})$ and the corresponding pseudorandom parameters. The suitable protocol is constructed in Section 8. Summing up we are able not only to extend the basic ID-based signature scheme against the compromising the PKG master key s, but also the signer private key $sk = sk(\mathrm{ID}, s, k)$ when being coerced by the adversary. Throughout the paper we will illustrate the related ideas on digital signatures with security based on hardness either of the factorization problem or the computational Diffie-Hellman problem in the gap Diffie-Hellman groups (GDH groups).

2. Related Work

The ID-based cryptosystems were introduced by Shamir [2]. The idea of gap Diffie-Hellman group based on the Weil pairing has its origin in the paper [3]. Boneh and Franklin [4] have proposed the first provably secure ID-based cryptosystem relating to GDH groups.

In [5]–[7] the ID-based digital signatures from the gap Diffie-Hellman groups were given. The proxy ID-based digital signature with derandomized Weil pairing computation was proposed in [8]. The general concept of transforming the standard signature schemes into the corresponding identity-based signatures (IBS) was the subject of paper [9].

In this paper we investigate the extensions of ID-based signature schemes having in mind the security requirements. The suitable improvements are based on the idea of subliminal channels investigated by Simmons [10] and applied in [11] for IBS scheme from the bilinear pairing. This approach was enhanced in [1] for the standard (certificate-based) signature schemes, referring to the concept of deniable encryption [12], [13].

The approach developed here deals with the ID-based digital signatures of the suitable form. We start from the standard IBS schemes and investigate the subsequent improvements by adding some “space” for the subliminal transfer and applying subsequently the concept of deniable signature.

The development is illustrated on the standard IBS schemes referring to [14] and [15]. At first we propose the extension of the standard IBS by including the signer private part to the secret key $sk(\mathrm{ID})$ and the corresponding approval by PKG (c.f. [16], [17]). Next we investigate the deniable encryption idea in standard signatures [1] to make the relevant transformation to IBS with the suitable security requirements.

3. Gap Diffie-Hellman Groups and Weil Pairing

Let $G = (G, +)$ be a group of prime order q and P, Q be any nontrivial elements of G. The discrete logarithm problem (DLP) in G may be stated as follows:

$$\text{Find } a \in Z_p \text{ such that } aP = Q.$$

Let us formulate the following related problems.

- The computational Diffie-Hellman problem (C-DH) – given the triple (P, aP, bP) find the element abP.
- The decisional Diffie-Hellman problem (D-DH) – given a quadruple (P, aP, bP, cP) decide whether $c = ab (\mathrm{mod}\, q)$ (in which case we shall write that $(P, aP, bP, cP) = \mathrm{DH}$).

We call the group $G = (G, +)$ a gap Diffie-Hellman (GDH) group if (roughly speaking) the D-DH problem is computationally easy, while the C-DH problem is hard.

Let us recall briefly the construction of the gap Diffie-Hellman group based on the elliptic group structure applying the Weil pairing [4]. Let E be an elliptic curve over a finite field K of characteristic p and let n an integer not divisible by p. Denote by $\mathrm{cl}(K)$ the algebraic closure of K. It can be shown that the group $E[n]$ of n-torsion points of $E/\mathrm{cl}(K)$ is isomorphic to $Z_n \times Z_n$. The Weil pairing is a map

$$e : E[n] \times E[n] \to \mathrm{cl}(K)^*,$$

satisfying the following properties:

- alternation: for all $P, Q \in E[n]$, $e(PQ) = e(QP)^{-1}$;
- bilinearity: for any $P, Q, R \in E[n]$ we have $e(P + Q, R) = e(P, R)e(Q, R)$;
- non-degeneracy: if $P \in E[n]$ is such that for all $Q \in E[n]$, $e(PQ) = 1$, then $P = O$;
- computability: there exist an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in E[n]$.

We now turn our attention to a more concrete situation. Let p be prime, $a \in Z_p^*$. Consider the elliptic curve E over F_p and the map $\Phi : E/\mathrm{cl}(F_p) \to E/\mathrm{cl}(Fp)$ defined by

$$E : Y^2 = X^3 + a, \Phi(O) = O;\ \Phi(x, y) = (\zeta x, y),$$

where

$$\zeta \in F_{p^2}^* \backslash \{1\}, \zeta^3 = 1,\ p = 2\ (\mathrm{mod}\, 3)$$

or

$$E : Y^2 = X^3 + aX, \Phi(O) = O;\ \Phi(x, y) = (-x, iy),$$

where

$$i \in F_{p^2}^*,\ i^2 = -1,\ p = 3\ (\mathrm{mod}\, 4).$$

One can easily check that Φ is an automorphism. Pick up a point $P \in E/F_p$ of prime order $q, q|p + 1 = \mathrm{card}\, E/F_p$. Then $E[q] = \langle P, \Phi(P)\rangle$. We define the modified Weil pairing $\hat{e}$ by

$$\hat{e} : G \times G \to G',\ \hat{e}(R,\ S) = e(R, \Phi(S)),$$

where

$$G_1 = \langle P\rangle,\ G' = F_{p^2}^*.$$

It easy to show that for every $R \in \langle P\rangle$ such that $\hat{\mathrm{e}}(R, P) = 1$, we have $R = 0$. It is known that the C-DH problem in G is hard (cf. [4]), but as it is shown in [18] not harder than the DLP in G'. The existence of Weil pairing implies directly that D-DH problem is easy in G. The randomized algorithm computing the Wail pairing was first proposed in [19]. The corresponding derandomized algorithm was shown in [8]. In what follows we will consider the bilinear structure (G, G', e, P) with G, G' and P as defined above and e being the suitable modified Weil pairing.

4. ID-Based Signature Schemes

The standard ID-based signature scheme considered in this paper is the tuple: IBS = (Setup, Extract, Sign, Verify). The corresponding algorithms are described below:

Setup of the system
The algorithm takes as input the security parameter and returns the description of the system, algebraic structure, the suitable hash functions and the pair (s,Ω), where s is the master key and Ω the corresponding PKG-public key.

Extract
Given the user identity (ID), PKG computes and sends (by a secure channel) the corresponding secret key for the signer $sk(\mathrm{ID}) = sk(\mathrm{ID}, s)$.

Signing
Having as input the message m, the secret key $sk(\mathrm{ID})$ and a random element r, the signer computes the signature of $m:$ $\mathrm{Sig} = [m,R,\sigma]$, where $R = R(m,r,\mathrm{ID})$ is the suitable commitment of r and $\sigma = \sigma(m,r,sk(\mathrm{ID}))$.

Verification
Any entity (verifier) having as input the signer identity ID, the signature Sig and PKG-public key Ω outputs "accept" or "reject" according to the verification process.

As an example we give the ID-based signature scheme from G-DH groups proposed in [7] (see also [15]), which will be the initial bilinear pairing based protocol for the further improvements.

Example 1: ID-based signature schemes from bilinear pairing
The protocol (IBSBP) consists of the following algorithms:

Setup of the system
Having as input the security parameter, it returns:
the bilinear structure (G,G',P,e), the pair $(s,\Omega = sP)$ and the suitable hash functions h and Q

$$h : \{0,1\}^* \to Z_q,$$

$$Q : \{0,1\}^* \to G.$$

Extract
Private key generator takes as input the identity ID of the user and returns the secret key $sk(\mathrm{ID}) = sQ(\mathrm{ID})$.

Signing
Given a message $m \in \{0,1\}^*$ the signer generates a random $r \in Z_q$ and computes the signature of $m : \mathrm{Sig} = [m,R,\sigma]$, where $R = rP$ and $\sigma = r\Omega + h(m,R)sk(\mathrm{ID})$.

Verification
To verify the signature $[m,R,\sigma]$ any user checks if $e(P,\sigma) = e(\Omega, R + h(m,R)Q(\mathrm{ID}))$.

5. Strong ID-Based Signature Scheme

In this section the enhance the IBS scheme to protect the signer against the compromising of the PKG master key s. This is due to the additional interactive protocol between the user U and PKG. The user generates a random (secret) k and the corresponding commitment K sends to PKG. This is the input for PKG to compute the suitable (public) verification key vk and the corresponding secret part for the signer. Consequently, the signer computes the final secret key $sk = sk(k,s,\mathrm{ID})$ not available for PKG. The protocol SIBS consists of the following algorithms: SIBS = (Setup, U-PKG, PKG-U, Keygen, Sign, Verify).

Setup
Having as input the security parameter, the algorithm returns the descryption of the system with the suitable hash functions and the pair (s,Ω).

***U*-PKG**
The signer U having as input the identity ID and Ω, generates the random secret key $k = k_{\mathrm{ID}}$ and publish the corresponding commitment $K = K(k,\mathrm{ID},\Omega)$.

PKG-*U*
PKG having as input the master key s and K computes and publish the verification key $vk = vk(s,\mathrm{ID},K)$ and the corresponding secret part $s_{\mathrm{PKG}-U}$ sends to the signer by the secure channel.

Keygen
The signer U having as input the private value $k = k(\mathrm{ID})$ and the secret part $s_{\mathrm{PKG}-U}$ computes the secret key $sk_U = sk_U(k, s_{\mathrm{PKG}-U})$ relating to the verification key $vk(s,\mathrm{ID},K)$.

Sign
The signer having as input the message m and the secret key sk_U computes the suitable signature $\mathrm{Sig} = \mathrm{Sig}(m, sk_U)$.

Verify
Any entity having as input the signature and the verification key $vk = vk(\mathrm{ID},K,\Omega)$ returns as output: accept or reject according to the verification process.

Example 2: Strong bilinear pairing based signature with credential delegation
Referring to Example 1 the above protocol is specified as follows:

Setup
Having as input the security parameter the algorithm returns the bilinear structure (G,G',e,P), the master key $s \in Z_q$ together with PKG-public key $\Omega = sP$ and suitable hash functions h,H and Q as below:

$$h : \{0,1\}^* \to Z_q,$$

$$H : G \to G,$$

$$Q : \{0,1\}^* \to G.$$

U-PKG

The algorithm is performed by the signer. Having as input P and random element $k \in Z_q$ the value $K = kP$ is computed and sent to PKG.

PKG-*U*

Having as input the master key s and the pair: (K,ID), PKG computes the credential approval $sH(vk) = sH(k\Omega) = sH(ksP) = sH(sK)$, the corresponding secret part sQ(ID) and send them to the signer.

Keygen

The algorithm is performed by the signer. Having as input the tuple $(k, sQ(\text{ID}))$ the signer computes the secret key $sk = sk(\text{ID}, k, s) = ksQ(\text{ID})$

Sign

The algorithm is performed by the signer. It has as input the message and signer's secret key $sk = ksQ(\text{ID})$ and returnes the signature: $[\text{Sig}, vk(\text{ID}),\ sH(vk(\text{ID}))]$, where $vk(\text{ID}) = k\Omega$, $\text{Sig} = [m, R, \sigma]$, with $R = rP$ and $\sigma = r[vk(\text{ID})] + h(m,R)sk(\text{ID})$.

Verify

The algorithm is performed by any verifier. First he checks the credential: $(vk(\text{ID}), sH(vk(\text{ID}))$ using the PKG-public key Ω. The credential is accepted provided $e(H(vk(\text{ID})), \Omega) = e(sH(vk(\text{ID})), P)$. If so he verifies the signature returning "accept" as output provided $e(P, \sigma) = e(vk(\text{ID}),\ R + h(m,R)Q(\text{ID}))$.

Example 3: Strong Fiat-Feige-Shamir ID-based signature scheme

The above protocol is the tuple: SFFSIBS = (Setup, PKGKeygen, U-PKG, PKG-U, Keygen, Sign, Verify) and is specified as follows:

Setup

Having as input the security parameter the algorithm returns the ring Z_n, the secure hash functions: $g, H : \{0,1\}^* \to Z_n$ with n to be specified latter on and the pair $(s, \Omega) = ((p,q), pq)$, where p, q are random prime numbers of suitable size.

U-PKG

Having as input the identity ID of the signer, the algorithm returns the private key $k = (p', q')$ and the corresponding commitment $K = p'q'$.

PKG-*U*

The algorithm is performed by PKG. Having as input the triple (ID,Ω, K) and the master key $s = (p,q)$, it returns the verification key $vk = vk(\text{ID}, K, \Omega) = (v_1, v_2, \ldots, v_l)$ mod $K\Omega$, with $v_j = H(\text{ID}\|_j)$ and the secret value $s_{\text{PKG}-U} = (s'_1, \ldots, s'_l)$ satisfying the equalities $(s'_j)^2 = v_j \bmod \Omega$, $j = 1, 2, \ldots, l$. The secret value $s_{\text{PKG}-U}$ is sent to the signer by the secure channel. Here H is the given hash function with the corresponding value of n being equal to $K\Omega$.

Keygen

Having as input the value of $s_{\text{PKG}-U}$ and $k = (p', q')$ the signer computes the secret key $sk = sk(\text{ID}, k, s) = (s_1, \ldots, s_l)$, satisfying the inequalities $(s_j)^2 = v_j \bmod K\Omega$, $j = 1, 2, \ldots, l$.

Sign

The algorithm has as input the message m, secret key $sk = (s_1, \ldots, s_l)$ and a random element $r \in Z_{K\Omega}$. As an output it returns $\text{Sig} = [m, \boldsymbol{R}, \sigma]$. Here $\sigma = r(s_1^{b_1}, \ldots, s_j^{b_l})$ with $\boldsymbol{R} = (b_1, \ldots, b_l)$ and $b_j (j = 1, 2, \ldots, l)$ are the subsequent bits of $g(m, U)$, with $U = r^2 \bmod K\Omega$, where g is the given hash function with the corresponding value $n = K\Omega$.

Verify

Having as input the message the signature $\text{Sig} = [m, \boldsymbol{R}, \sigma]$ and the verification key $vk = vk(\text{ID}, K, \Omega)$ the algorithm outputs "accept" provided $\sigma^2(v_1^{b_1}, v_2^{b_2}, \ldots, v_l^{b_l})^{-1} \bmod K\Omega$ is equal to U', such that $g(m, U')$ has the subsequent bits equal to $b_j, j = 1, 2, \ldots, l$.

6. T-Shared Key ID-Based Signature Scheme (*T*-SKIBS)

Some types of digital signatures require the presence of the selected third party in the verification process (see, eg., [20] and [21] – relating to the IBS schemes from bilinear pairing). We recall that we consider the signatures of the form $\text{Sig} = [m, R, \sigma]$, where R denotes the suitable commitment of the pseudorandom element $r = r(m, k, \text{ID})$ involved in the generation of σ. Now we will create the relevant subliminal channel to hide in R the information readable only for some third party T that shares the secret key k with the signer U. We assume that R is the commitment of r derived by the application of one-way (trapdoor) homomorphism Φ and H is a suitable hash function with the image included in the domain of Φ. The general scheme is the following: T-SKIBS = (Setup, Share$U$-T, Extract, Sign, Verify, Verify*), where the algorithms are described as follows:

Setup

Having as input the security parameter, the corresponding algebraic structure, the suitable hash functions and the pair (s, Ω) are given as output.

Share*U*-*T*

This is an interactive protocol between the signer U and the trustee T, at the end of which the secret shared key k is computed.

Extract

The algorithm is performed by PKG. It takes as input the pair (ID,s) and outputs the secret key of the signer $sk = sk(\text{ID}, s)$.

Sign

The input is the tuple $(m,k,sk(\mathrm{ID}),r)$, where $r = H(m,k,\Omega)$. The output is the signature that has the form $\mathrm{Sig} = [m,R,\sigma]$, where $\sigma = \sigma(m,k,sk(\mathrm{ID}),r)$.

Verify

The algorithm is performed by any entity. Having as input the signature Sig and the PKG-public key Ω. The algorithm outputs "accept" if σ is consistent with the pseudorandom parameter R and the message m.

Verify*

The algorithm is performed by the trustee T. Having as input the tuple (σ,k,Ω), the output is "accept" provided R is consistent with the value of $\Phi(H(m,k))$.

The algorithm Verify* provides us with the additional protection of the signer against the compromising of the secret key $sk = sk(\mathrm{ID})$, since it is no more equivalent to loosing the identity of the signer.

Example 4: *T*-shared key ID-based signature with bilinear pairing

Let (G,G',e,P) be a bilinear structure and $\Phi : Z_q \to G$ be the corresponding additive one-way group homomorphism. We let $R = \Phi(r)$ and $H : \{0,1\}^* \to Z_q$ be the suitable hash function. Using this specification for the protocol from Section 4, we see that the suitable changes will concern only the algorithms: Sign, Verify and Verify*. Namely the signer selects a random $r' \in Z_q$ and computes the commitment $R' = r'P$. Next he computes $r'' = H(m,k,R')$ and the generated signature has the form: $\mathrm{Sig} = [m,R',r'',\sigma^*]$, where $\sigma^* = \sigma(m,sk(\mathrm{ID}),r'+r'')$. The verification algorithm (Verify) is actually the same as in the basic scheme with the commitment R replaced by $R'+R''$. The additional (strong) verification (algorithm Verify*) checks if actually $\Phi(H(m,k,R')) = R''$.

The additional information r'' (superfluous for any verifier except T) is just the "hint" for the trustee to decide if the signature was generated by the authorized signer or not (algorithm Verify*).

7. Strong *T*-Shared ID-Based Signature Scheme (ST-SIBS)

Joining the concept of T-shared key (Section 6) with the extended communication (U-PKG, PKG-U) between the signer and PKG (Section 5) we arrive at the following protocol: ST-SIBS = (Setup, Share*U*-*T*, *U*-PKG, PKG-*U*, Keygen, Sign, Verify, Verify*).

Setup

Having as input the public data, the algorithm returns the corresponding algebraic structure suitable hash functions and the pair (s,Ω).

Share*U*-*T*

This is an interactive protocol between the signer U and the trustee T, at the end of which the secret shared key κ is computed.

U-PKG

The signer U having as input the identity ID and Ω, generates the random secret key $k = k_{\mathrm{ID}}$ and publish the corresponding commitment $K = K(k,\mathrm{ID},\Omega)$.

PKG-*U*

PKG having as input the master key s and K computes and publish the verification key $vk = vk(s,\mathrm{ID},K)$ and the corresponding secret part $s_{\mathrm{PKG}-U}$ sends to the signer by the secure channel.

Keygen

Having as input the private value $k = k(\mathrm{ID})$ and the secret part $s_{\mathrm{PKG}-U}$ the signer U computes the secret key $sk_U = sk_U(\mathrm{ID},k,s_{\mathrm{PKG}-U})$ relating to the verification key $vk(s,\mathrm{ID},K)$.

Sign

Having as input the message m, the secret key sk_U and the corresponding pseudorandom value $r = r(m,k)$ the signer computes the suitable signature $\mathrm{Sig} = [m,R,\sigma]$, where $\sigma = \sigma(m,k,r,sk_U)$ and $R = R(m,k,r,\Omega)$ is the corresponding commitment of r.

Verify

Any entity having as input the verification key $vk = vk(\mathrm{ID},K,\Omega)$ and signature Sig returns as output accept or reject according to the verification process.

Verify*

Having as input the tuple (σ,κ,Ω), the trustee (T) outputs "accept" provided R is consistent with the value of $\Phi_\kappa(m)$ and "reject" otherwise.

The validity of the signature is checked by two kind of verification-weak verification which can be made by any entity and strong verification performed only by the trustee. The secret key of the signer $sk(\mathrm{ID}) = sk(\mathrm{ID},s_{\mathrm{PKG}-U},k)$ is computed with the aid of PKG master key s and the secret key κ shared with T. Therefore neither PKG nor T is able to forge the signature. Moreover, even in the case when the secret key $sk(\mathrm{ID})$ is compromising (but not the shared key κ), trustee can still distinguish between the signature generated by the real signer and the forger. This is due to the algorithm Verify* in the above protocol. Below we show the suitable example based on the Fiat-Feige-Shamir signature.

Example 5: Strong FFS *T*-shared key ID-based signature scheme

The protocol follows the above steps with the suitable specifications (see Example 3), i.e., SFFST-SKIBS = (Setup, Share*U-T, U*-PKG, PKG-*U*, Keygen, Sign, Verify, Verify*).

The only changes when compared with Example 3 concern the additional algorithms Share$U$-T, Verify* and the suitable modification in algorithm Sign (related to the dependence of the pseudorandom value r on the shared key κ). Namely:

Share*U-T*
The signer (U) and trustee (T) proceed the interactive protocol which outputs the shared secret key κ.

Sign
The algorithm takes as input the message m, secret key $sk = (s_1, \dots, s_l)$ and a random element $r' \in Z_\Omega$. Applying the chineese remainder theorem we compute $r \bmod K\Omega : r = h(m,\kappa) \bmod K$ and $r = r' \bmod \Omega$. As an output it returnes $\mathrm{Sig} = [m, \boldsymbol{R}, \sigma]$. Here $\sigma = r(s_1^{b_1}, \dots, s_l^{b_l}) \bmod K\Omega$, where $\boldsymbol{R} = (b_1, \dots, b_l)$ and $b_j (j = 1, 2, \dots, l)$ are the subsequent bits of $g(m,U)$, with $U = r^2 \bmod K\Omega$ and g being the suitable hash function $g : \{0,1\}^* \to Z_n (n = K\Omega)$.

Verify*
The trustee applying the verification key $vk = vk(\mathrm{ID}, K, \Omega) = (v_1, v_2, \dots, v_l) \bmod K\Omega$ computes first the value $r^2 \bmod K\Omega$ then using the secret value $k = (p', q')$ corresponding to the commitment K computes the square root $r \bmod K$ and check if $r = h(m,\kappa)$, where κ is the secret key shared between the signer and trustee.

8. Deniable T-Shared Key ID-Based Signature (DT-SKIBS)

The signature scheme considered in Section 7 has the following (sometimes undesired) properties:

- the same message signed twice has the same signature;
- given the valid signature (satisfying both algorithms Verify and Verify*) the signer can check (prove another party) that the signature satisfies indeed the strong verification algorithm.

The second property might be used by the adversary in the so called coercion attack considered in the context of an encryption process in [12]. Suppose that the sender encrypts the message and sends it to the receiver. After some time the sender can be coerced by an adversary to give up the plaintext message together with the random choices involved in the encryption process. We can pose the question: can the sender protect himself against such an attack? The original idea of translucent sets applied in [12] was then exploited in [13] to give the more practical solution. Here we adopt the idea of deniable encryption in the context of the signature schemes that allows to avoid the above weakness. The idea is as follows. We let the pseudorandom element r be depending on some "random" factor ρ, so that for any fixed m, κ, κ' and ρ, there exists a corresponding ρ' satisfying $r = r(m,\kappa,\rho) = r(m,\kappa',\rho')$. In the case of the coercion attack the signer could recover the "fake" value κ' still keeping the real value of κ secret. This idea was investigated in [1] in the case of the certificate based signature schemes. Below we adopt it for the ID-based signature schemes. In the following protocol we incorporate the deniable encryption affecting merely the algorithms Share$U$-T, Sign and Verify* in the above ST-SIBS scheme. The corresponding protocol runs as follows:

Setup
Having as input the security parameter the algorithm returns the corresponding algebraic structure of the system together with the suitable hash function $h : \{0,1\}^* \to \{0,1\}^*$, and the pair (s, Ω).

Share*U-T*
In this part the suitable deniable encryption function $E_{\mathrm{den}} : Z_n \to Z_n$ and the corresponding decryption function $D_t : \mathrm{Im}(E_{\mathrm{den}}) \to Z_n$ are defined. Moreover, the signer and trustee (T) compute the shared secret key κ, while the corresponding trapdoor information t is known only for T.

***U*-PKG**
The signer U having as input the identity ID and Ω, generates the random secret key $k = k_{\mathrm{ID}}$ and publish the corresponding commitment $K = K(k, \mathrm{ID}, \Omega)$.

PKG-*U*
PKG having as input the master key s and the commitment K computes and publish the verification key $vk = vk(s, \mathrm{ID}, K)$ and the corresponding secret part $s_{\mathrm{PKG}-U}$ sends to the signer by the secure channel.

Keygen
Having as input the private value $k = k_{\mathrm{ID}}$ and the secret part $s_{\mathrm{PKG}-U}$ the signer U computes the secret key $sk_U = sk_U(\mathrm{ID}, k, s_{\mathrm{PKG}-U})$ relating to the verification key $vk(s, \mathrm{ID}, K)$.

Sign
Having as input the message m the secret key sk_U and the pseudorandom value $r = E_{\mathrm{den}}[h(m,\kappa,\rho)]$ the suitable signature $\mathrm{Sig} = [m, R, \sigma]$, with $\sigma = \sigma(m, k, r, sk_U)$ and the commitment $R = R(m, k, r, \Omega)$ of r is computed as output.

Verify
Any entity having as input the verification key $vk = vk(\mathrm{ID}, K, \Omega)$ and signature Sig returns as output "accept" if σ is consistent with (m, R) or "reject" otherwise.

Verify*
Having as input the tuple $(t, \sigma, \kappa, \Omega)$ the trustee (T) outputs "accept" provided the (t,k)-"trapdoor" inverse of R agrees with $h[(m,\kappa)]$ and "reject" otherwise.
Below we illustrate the above protocol using the standard ID-based signatures given in [7] and [14].

Example 6: Deniable *T*-shared ID-based signature from bilinear pairing

The protocol consists of the following algorithms: Setup, Share$U$-T, Extract, Sign, Verify, Verify*.

Setup

Given the security parameter the bilinear structure (G, G', e, P), the suitable hash functions $h : \{0,1\}^* \rightarrow Z_n$, $H : Z_n \rightarrow Z_q$, $Q : \{0,1\}^* \rightarrow G$ and the pair (s, Ω) are returned.

Share*U*-*T*

Here the suitable deniable encryption function $E_{\text{den}} : Z_n \rightarrow Z_q$ and the corresponding decryption function $D_t : \text{Im}(E_{\text{den}}) \rightarrow Z_n$ are defined. The signer (U) and trustee (T) compute the shared secret key κ, while the corresponding trapdoor information t (allowing to decrypt the deniably encrypted message) is known only for T.

Extract

Having as input the signer identity ID the algorithm returns the secret key for the signer $sk(\text{ID}) = sQ(\text{ID})$.

Sign

The signer selects a random $r' \in Z_q$ and computes the commitment $R' = r'P$. Next he applies the deniable encryption algorithm to compute $r'' = E_{\text{den}}[h(m, \kappa, R')]$. The signature is: $\text{Sig} = [m, R', r'', \sigma]$, where $\sigma = (r' + r'')\Omega + H(m, R' + r''P)sQ(\text{ID})$.

Verify

Any user checks if $e(P, \sigma) = e(\Omega, R' + r''P + H(m, R' + r''P)Q(\text{ID}))$.

Verify*

Using the trapdoor information t the trustee decrypts the value r'' and checks if it is equal to $h(m, \kappa, R')$.

Example 7: Strong FFS deniable *T*-shared key ID-based signature

Specifying the above general DT-SKIBS scheme with the SFFST-SKIBS protocol (see Example 5) we obtain the following protocol SFFSDT-SKIBS = (Setup, Share*U*-*T*, PKGKeygen, PKG-*U*, Keygen, Sign, Verify, Verify*). The corresponding algorithms are as follows:

Setup

Having as input the security parameter the ring Z_n together with the suitable hash functions: $H, g, h : \{0,1\}^* \rightarrow Z_n$ (with the values $n = n(H)$, $n = n(g)$, $n = n(h)$ to be specified, respectively) and the pair $(s, \Omega) = ((p, q), pq)$ with the suitable prime numbers p, q are given as output with the image of h contained in the interval $[M, M + K]$ for the suitable values of M and K.

Share*U*-*T*

Here the suitable deniable encryption function $E_{\text{den}} : Z_{n(h)} \rightarrow Z_N$ and the corresponding decryption function $D_t : \text{Im}(E_{\text{den}}) \rightarrow Z_n$ are defined. The signer (U) and trustee (T) compute the shared secret key κ, while the corresponding trapdoor information t (allowing to decrypt the deniably encrypted message) is known only for T.

U-PKG

Having as input the identity ID of the signer, the algorithm returns the private key $k = (p', q')$ and the corresponding commitment $K = p'q'$ is sent to PKG.

PKG-*U*

The algorithm is performed by PKG. Having as input the triple (ID, Ω, K) and the master key $s = (p, q)$, it returns the verification key $vk = vk(\text{ID}, K, \Omega) = (v_1, v_2, \ldots, v_l)$ mod $K\Omega$, with $v_j = H(\text{ID}\|_j)$ and the secret value $s_{\text{PKG}-U} = (s'_1, \ldots, s'_j)$ satisfying the equalities $(s'_j)^2 = v_j$ mod Ω, $j = 1, 2, \ldots, l$. The secret value is sent to the signer. Here H is the given hash function with the corresponding value of $n(H)$ being equal to $K\Omega$.

Keygen

Having as input the value of $s_{\text{PKG}-U}$ and $k = (p', q')$ the signer computes the secret key $sk = sk(\text{ID}, k, s) = (s_1, \ldots, s_l)$, satisfying the inequalities $(s_j)^2 = v_j$ mod $K\Omega$, $j = 1, 2, \ldots, l$.

Sign

The algorithm has as input the message m, secret key $sk = (s_1, \ldots, s_l)$ and a random element $r' \in Z_\Omega$. Applying the chineese remainder theorem we compute r mod $K\Omega : r = E_{\text{den}}(h(m, \kappa))$ mod K and $r = r'$ mod Ω (with r belonging to the interval $[M, M + K]$). As an output it returns $\text{Sig} = [m, \boldsymbol{R}, \sigma]$. Here $\sigma = r(s_1^{b_1}, \ldots s_l^{b_l})$, where $\boldsymbol{R} = (b_1, \ldots, b_l)$ and $b_l (j = 1, 2, \ldots, l)$ are the subsequent bits of $g(m, U)$, with $U = r^2$ mod $K\Omega$ and $n(g) = 2^{1+1}$.

Verify

Having as input the signature $\text{Sig} = [m, R, \sigma]$ and the verification key $vk = vk(\text{ID}, K, \Omega) = (v_1, v_2, \ldots, v_l)$ mod $K\Omega$, the algorithm outputs "accept" provided $\sigma^2 (v_1^{b_1}, v_2^{b_2}, \ldots, v_l^{b_l})^{-1}$ mod $K\Omega$ is equal to U', such that $g(m, U')$ has the subsequent bits equal to $b_j, j = 1, 2, \ldots, l$.

Verify*

Applying the verification key $vk = vk(\text{ID}, K, \Omega) = (v_1, v_2, \ldots, v_l)$ mod $K\Omega$ the trustee computes first the value r^2 mod $K\Omega$. Then using the secret value $k = (p', q')$ corresponding to the commitment K computes the square root r mod K and check if $D_t(a)$ is equal to $h(m, \kappa)$, where κ is the secret key shared between the signer and trustee, while a is the unique number congruent to r mod K belonging to the interval $[M, M + K]$.

9. Concluding Remarks

In the paper we have investigated the possible improvements of the ID-based signature schemes in a successive way, from the simpler protocols to the more advanced ones. To increase the clarity of presentation we have illustrated the ideas by the examples of two basic schemes due to Fiat-Feige-Shamir (see [14]) and bilinear pairing based protocol due to X.Yi [7]. The security of the underlying schemes relies on different computational problems namely the integer factorization problem and C-DH problem in the group of n-torsion points of elliptic curve over the finite field, respectively. Both the complexity and security of the basic schemes were studied in details in the literature. Here we have focused our attention towards the protection of the investigated schemes against the risk of compromising the private key of the signer and the so called coercion attack (see, e.g., [1]). The main ingredient applied in this approach was the construction of the suitable subliminal channel in the underlying digital signatures.

This channel can be used to protect the signer in case of force, blackmail, etc. The critical information leaked subliminally can be read only by the pointed third party that knows some trapdoor information. The given evidence prove that some weakness of the ID-based digital signatures could be overcome by application of the interactive secret key generation and the idea of the deniable encryption.

Acknowledgement

The author is grateful to dr Jerzy Pejaś for reading of the first version of the article and his valuable remarks concerning the paper.

References

[1] K. Durnoga, J. Pomykała, and T. Trabszys, "Signature scheme with blackmail warning" (preprint).

[2] A. Shamir, "Identity-based cryptosystems and digital signatures", in *Proc. Crypto'87*, Santa Barbara, USA, 1987, pp. 47–53.

[3] A. Joux, "A one-round protocol for tripartite Diffie-Hellman", *J. Cryptol.*, vol. 17, no. 4, pp. 263–276, 2004.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing", *SIAM J. Comp.*, vol. 32, no. 3, pp. 586–615, 2003.

[5] J. C. Cha and J. H. Cheon, *An Identity-Based Signature from Gap Diffie-Hellman Groups*, LNCS, vol. 2567. Berlin: Springer, 2003, pp.18–30.

[6] R. Sakai and M. Kasahara, "ID based cryptosystems with pairing on elliptic curve", in *Symp. Cryptogr. Inform. Secur. SCIS'2003*, Hamamatsu, Japan, 2003.

[7] X. Yi, "An identity based signature scheme from the Weil pairing", *IEEE Commun. Lett.*, vol. 7, no. 2, pp. 76–78, 2003.

[8] J. Pomykała and B. Źrałek, "A model of Id-based proxy signature scheme", in *Proc. 6th Coll. Iberoam. Collab. Electron. Commun. eCommerce Tech. Res. Conf.*, Madrid, Spain, 2008.

[9] M. Bellare, C. Namprempre, and G. Neven, *Security Proofs for Identity-Based Identification and Signature Schemes*, LNCS, vol. 3027. Berlin: Springer, 2004, pp. 268–286.

[10] G. J. Simmons, "The subliminal channel and digital signatures", in *Proc. EUROCRYPT'84 Worksh. Adv. Cryptol. Theory Appl.*, Paris, France, 1985, pp. 364–378.

[11] J. Pomykała and T. Trabszys, "Blackmail warning verifiably encrypted signatures from bilinear pairing", *Bull. WAT*, vol. LVII, no. 4, pp. 167–182, 2008.

[12] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, *Deniable Encryption*, LNCS, vol. 1294. Berlin: Springer, 1997, pp. 90–104.

[13] M. Klonowski, P. Kubiak, and M. Kutyłowski, *Practical Deniable Encryption*, LNCS, vol. 4910. Berlin: Springer, 2008, pp. 599–609.

[14] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity", *J. Cryptol.*, vol. 1, iss. 2, pp. 77–94, 1988.

[15] X. Cheng, J. Liu, and X. Wang, *Identity-Based Aggregate and Verifiably Encrypted Signatures from Bilinear Pairing*, LNCS, vol. 3483. Berlin: Springer, 2005, pp. 1046–1054.

[16] R. Tamassia and D. Yao, "Cascaded authorization with anonymous – signer aggregated signatures", in *Proc. IEEE Inform. Assur. Worksh.*, Royal Holloway, UK, 2006, pp. 84–91.

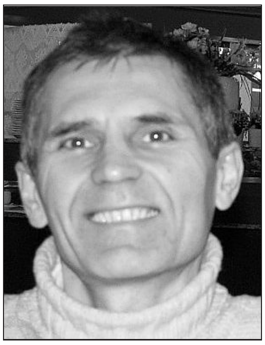
[17] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights", *Cryptol. ePrint Arch.*, 2003, Rep. 2003/096.

[18] A. Menezes, T. Okamoto, and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Trans. Inform. Theory*, vol. 39, pp. 1639–1646, 1993.

[19] V. S. Miller, "The Weil pairing, and its efficient calculation", *J. Cryptol.*, vol. 17, no. 4, pp. 233–334, 2004.

[20] J. Pejaś, "ID-based directed threshold signcryption scheme using a bilinear pairing", *Polish J. Envir. Stud.*, vol. 17, no. 4C, pp. 335–341, 2008.

[21] J. Pomykała and B. Źrałek, "Electronic signature, developement perspective", in *Electronic Signature and Biometric Authentication*, B. Hołyst and J. Pomykała, Eds. Warsaw: WSM, 2009 – in print (in Polish).



Jacek Pomykała (born in 1958) has obtained the M.Sc.degree in 1981 and Ph.D. degree in 1986 at the Faculty of Mathematics Informatics and Mechanics of University of Warsaw. His habilitation was done in 1997 at the Institute of Mathematics of Polish Academy of Sciences. He works in the Mathematical Institute at the Faculty of Mathematics Informatics and Mechanics of Warsaw University. His scientific interests are number theory, cryptology, computational complexity, security of computer systems. He was the invited speaker of many international conferences in the area of mathematics, computer science and security systems, the author of over 30 publications in the various international journals of these domains. He is an author of one book on the modeling and security of information systems, one of the editors of the book concerning the cryptographic and biometrics authentication and the special volume of the International Journal of Biometrics entitled "Digital signature and biometric in theory and practice". At present he is the leader of the international research seminar on Computational Number Theory and Cryptology at the Faculty of Mathematics Informatics and Mechanics of Warsaw University.

e-mail: pomykala@mimuw.edu.pl
Institute of Mathematics
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw
Banacha st 2
02-097 Warsaw, Poland