Identity-Based Cryptosystems

Benoît Libert, Jean-Jacques Quisquater
Microelectronics Laboratory, Université catholique de Louvain, Louvain-la-Neuve, Belgium

Synonyms

IBE: Identity-based encryption; IBS: Identity-based signature

Background

Identity-based public key cryptography is a paradigm introduced by Shamir in 1984 [36]. His motivation was to simplify key management and remove the need for public key certificates as much as possible by letting the user's public key be the binary sequence corresponding to an information identifying him in a nonambiguous way (e-mail address, IP address combined to a user name, telephone number, etc). The removal of certificates allows avoiding the trust problems encountered in current public key infrastructures (PKIs): it is no longer necessary to bind a public key to its owner's name since those are one single thing, and it also simplifies key management since public keys are human-memorizable. These systems involve trusted authorities called private key generators (PKGs) that have to deliver private keys to users after having derived them from their identity information (users do not generate their key pairs themselves) using a master secret key. End users do not have to enquire for a certificate for their public key. The only things that still must be certified are the public keys of trusted authorities. This does not completely eliminate the need for certificates but, since many users depend on the same authority, this need is drastically reduced. Several practical solutions for identity-based signatures (IBS) have been devised since 1984 [15, 23, 34], but finding a practical identity-based encryption scheme (IBE) remained an open challenge until 2001 when elegant solutions were provided by Boneh and Franklin [8] and Cocks [13]. Other identity-based signatures were proposed after 2001 (e.g., [10, 26]).

Basically, an identity-based cryptosystem consists of four algorithms. First, a *Setup* algorithm, which is run by a PKG, takes as input a security parameter to output a public/private master key pair $(\mathsf{mpk}, \mathsf{msk})$ for the PKG. A key generation algorithm *Keygen* is also run by the PKG: it takes as input the PKG's master secret key msk and a user's identity ID to return the user's private key d_{ID}. In the case of identity-based encryption, the third algorithm is an encryption algorithm *Encrypt* that can be publicly run by anyone and takes as input a plaintext M, the recipient's identity, and the PKG's master public key mpk to output a ciphertext C. The last algorithm is then the decryption algorithm *Decrypt* that takes as input the ciphertext C and the private decryption key d_{ID} to return a plaintext M. In the case of identity-based signatures, the last two algorithms are the signature generation algorithm *Sign* that, given a message M, the PKG's public key and a private key d_{ID} generates a signature on M that can be verified by anyone thanks to the signature verification algorithm *Verify*. The latter takes as input the PKG's key mpk and the alleged signer's identity ID to return 1 or 0 depending on whether the signature is acceptable or not.

This chapter surveys some simple identity-based schemes that have appeared in the literature since Shamir's call for proposals in 1984. Some of the most famous identity-based signature schemes are fist described, and the chapter then gives an example of identity-based encryption scheme based on modular arithmetic.

Theory

This section presents simple examples of identity-based signatures. The first one is a generic construction that can be based on any signature scheme. The second one is the Guillou–Quisquater [23] signature scheme that builds on the RSA assumption. The next example is a scheme, proposed by Bellare, Namprembre, and Neven [2], which relies on the difficulty of computing discrete logarithms. The chapter finally outlines a simple identity-based encryption scheme due to Cocks [13].

Identity-Based Signatures

Syntactically, an identity-based signature consists of the following four algorithms.

Setup: is a probabilistic algorithm run by a private key generator (PKG) that takes as input a security parameter to output a master public key mpk and a master secret key msk which is kept secret.

Keygen: is a private key generation algorithm run by the PKG on input of params and the master key msk to return a private key d_{ID} associated with the identity ID.

Sign: is a (possibly probabilistic) algorithm that takes as input public parameters params, a message M and the signer's private key d_{ID}, and outputs a signature $\sigma = \mathbf{Sign}(d_{ID}, M)$.

Verify: is a deterministic verification algorithm that takes as input a purported signature σ, the master public key mpk and the signer's identity ID. It outputs 0 or 1.

The security of IBS schemes is formalized via a game between a challenger and an adversary $\mathcal{F}$. More precisely, the definition used in [2, 14] extends the standard notion

[22] of existential unforgeability under chosen-message attacks by requiring any probabilistic polynomial-time algorithm $\mathcal{F}$ to have negligible advantage in the following game.

1. The challenger generates a master key pair $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathbf{Setup}(k)$ and hands mpk to the forger $\mathcal{F}$.
2. On polynomially-many occasions, the forger makes queries of the following two types:
   (a) Key generation queries: $\mathcal{F}$ chooses an arbitrary identity ID and the challenger replies by returning $d_{ID} \leftarrow \mathbf{Keygen}(\mathsf{msk}, ID)$.
   (b) Signing queries: $\mathcal{F}$ chooses a pair (ID, M). The challenger replies by computing $d_{ID} \leftarrow \mathbf{Keygen}(\mathsf{msk}, ID)$ and returning $\sigma = \mathbf{Sign}(d_{ID}, M)$.

   These queries can be made adaptively in that each one may depend on the answers to prior queries.
3. The forger outputs a triple $(ID^\star, M^\star, \sigma^\star)$ and wins if the three following conditions are satisfied:
   (a) $ID^\star$ was never queried for key generation.
   (b) The pair $(ID^\star, M^\star)$ was never queried for signature.
   (c) $\mathbf{Verify}(\mathsf{mpk}, ID^\star, \sigma^\star) = 1$.

Generic IBS from Any Signature

In [2], Bellare, Namprempre, and Neven pointed out that any digital signature can be made identity-based in a very simple manner using certification. This generic construction was previously implicitly described in [14] and goes as follows. Let $\Pi^{\mathsf{sig}} = (\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ be any ordinary (i.e., nonidentity-based) digital signature scheme providing existential unforgeability under chosen-message attacks [22]. Then, a secure IBS scheme $\Pi^{\mathsf{IBS}} = (\mathbf{Setup}, \mathbf{Keygen}, \mathbf{Sign}, \mathbf{Verify})$ can be obtained as follows.

Setup: Given a security parameter $k \in \mathbb{N}$, the algorithm generates a digital signature key pair $(pk, sk) \leftarrow \mathsf{Keygen}(k)$ and returns $(\mathsf{mpk}, \mathsf{msk}) = (pk, sk)$.

Keygen: To generate a private key for some user's identity ID, the PKG generates a fresh digital signature key pair $(pk_{ID}, sk_{ID}) \leftarrow \mathsf{Keygen}(k)$ and sets $d_{ID} = (sk_{ID}, pk_{ID}, \sigma_{ID})$, where $\sigma_{ID} = \mathsf{Sign}(\mathsf{msk}, ID \| pk_{ID})$ is a certificate binding the newly generated public key pk_{ID} to the identity ID.

Sign: To sign a message m using is private key $d_{ID} = (sk_{ID}, pk_{ID}, \sigma_{ID})$, the signer computes $\sigma_m = \mathsf{Sign}(sk_{ID}, m)$ and the identity-based signature is defined as the triple $\sigma = (\sigma_m, pk_{ID}, \sigma_{ID})$.

Verify: To verify an alleged signature $\sigma = (\sigma_m, pk_{ID}, \sigma_{ID})$ for message m under the identity ID, the verifier returns 1 if $\mathsf{Verify}(pk_{ID}, \sigma_m, m) = 1$ and $\mathsf{Verify}(\mathsf{mpk}, \sigma_{ID}, ID \| pk_{ID})$=1. Otherwise, it outputs 0.

The above construction was extended to provide hierarchical identity-based signatures [25] (i.e., IBS schemes involving a hierarchy of signers organized in a hierarchical setting) and identity-based signatures with special properties [18].

While simple and elegant, this construction leaves room for efficiency improvements, notably in terms of signature size since each signature comprises two ordinary digital signatures and a public key. Ideally, one would like to have an IBS scheme performing as well as ordinary digital signatures. The next subsections show two examples of such schemes that both derive from the Fiat–Shamir paradigm [15].

The Guillou–Quisquater IBS

This scheme is derived from a three round identification scheme. It was proposed in 1988 and consists of the following algorithms.

Setup: Given a security parameter k_0, the private key generator (PKG) picks two $k_0/2$-bit primes p and q and computes $n = pq$. It also picks a prime number $e \in \mathbb{Z}_{\varphi(n)}$ such that $\gcd(e, \varphi(n)) = 1$ and chooses cryptographic hash functions $H : \{0,1\}^* \rightarrow \mathbb{Z}_e$ and $G : \{0,1\}^* \rightarrow \mathbb{Z}_n$. The master public key is $\mathsf{mpk} = (n, e, G, H)$ while the master secret key is the pair $\mathsf{msk} = (p, q)$.

Keygen: Given a user's identity ID, the PKG computes $I = G(ID) \in \mathbb{Z}_n^*$ and $a \in \mathbb{Z}_n^*$ such that $Ia^e \equiv 1 \pmod{n}$. The obtained $d_{ID} = a$ is returned to the user as a private key.

Sign: Given a message m, the signer does the following:

1. Pick a random $k \overset{R}{\leftarrow} \mathbb{Z}_n^*$ and compute $r = k^e \bmod n$.
2. Compute $\ell = H(m \| r) \in \mathbb{Z}_e$.
3. Calculate $s = ka^\ell \bmod n$.

The signature on m is the pair (s, ℓ).

Verify: To verify a signature (s, ℓ) on m:

1. Compute $I = G(ID)$ from the signer's identity ID.
2. Compute $u = s^e I^\ell \bmod n$.
3. Accept the signature if $\ell = H(m \| u)$.

To verify the consistency of the scheme, note that

$$u \equiv s^e I^\ell \equiv (ka^\ell)^e I^\ell \equiv k^e (a^e I)^\ell \equiv k^e \equiv r \pmod{n}.$$

Hence $u = r$ and then $H(m \| u) = H(m \| r)$.

For security reasons, the parameter k_0 should be at least 1024 or 2048 to avoid attacks trying to factor the modulus.

This signature scheme is derived from the Guillou–Quisquater identification protocol (GQ) using the Fiat–Shamir heuristic [15] that turns any 3-move identification scheme into a digital signature. In the scheme, the signer's private key is an RSA signature generated by the PKG on a message consisting of the user's identity: in other words, a GQ signature is actually a noninteractive proof of knowledge of an RSA signature.

The scheme can be proved existentially unforgeable (in the random oracle model [6]) provided it is hard to invert the RSA function. The first security proof can be traced back to the work of Pointcheval and Stern [32, 33] who showed how their "forking technique" yields security proofs for signature schemes derived from identification protocols. Their security proof was given in a model where the scheme was treated as an ordinary (i.e., nonidentity-based) signature. A security proof in the model of section "Identity-Based signatures" was provided by the general framework of Bellare et al. [2].

While the infeasibility of inverting the RSA function is sufficient to prove the security of the GQ signature scheme in the random oracle model, a stronger interactive assumption (introduced in [3]) is necessary to prove the security of the underlying interactive identification scheme. The security of the GQ identification scheme against active and concurrent attacks was established by Bellare and Palacio [5] in a traditional public key setting. Its security in the identity-based model was proved in [2].

The Bellare–Namprempre–Neven IBS

In 2004, Bellare et al. [2] described an identity-based signature based on the discrete logarithm problem. In the same way as the Guillou–Quisquater scheme can be seen as a proof of knowledge of an RSA signature, the Bellare et al. scheme can be viewed as a noninteractive proof of knowledge of a Schnorr [35] signature.

The Schnorr signature scheme makes use of a cyclic group $\mathbb{G}$ of prime order p. The signer holds a public key $X = g^x$, where $x \overset{R}{\leftarrow} \mathbb{Z}_p$ is the private key which is used to sign a message m as follows. The signer first computes $R = g^r$, for a randomly chosen $r \overset{R}{\leftarrow} \mathbb{Z}_p$, and sets $s = r + H(m||R)x \bmod p$, where $H : \{0,1\}^* \to \mathbb{Z}_p$ is a hash function modeled as a random oracle. The signature consists of (ℓ, s), where $\ell = H(m||R)$, and is verified by checking whether $\ell = H(m||g^s X^{-\ell})$. Alternatively, the signature can be (R, s) in such a way that the verifier can recompute $\ell = H(m||R)$ before checking whether $R = g^s X^{-\ell}$.

Setup: Given a security parameter k_0, the PKG chooses a cyclic group $\mathbb{G}$ of prime order $p > 2^{k_0}$ and a generator $g \in \mathbb{G}$. It picks $x \overset{R}{\leftarrow} \mathbb{Z}_p$ and sets $X = g^x$. It also chooses cryptographic hash functions $H : \{0,1\}^* \to \mathbb{Z}_p$ and $G : \{0,1\}^* \to \mathbb{Z}_p$. The master public key is $\mathsf{mpk} = (g, X, H, G)$ while the master secret key is $\mathsf{msk} = x$.

Keygen: Given a user's identity ID, the PKG chooses $r \overset{R}{\leftarrow} \mathbb{Z}_p$ at random and computes $R = g^r$ as well as $\ell = H(ID||R) \in \mathbb{Z}_p$. The private key is the pair $d_{ID} = (R, s)$ where $s = r + \ell x \bmod p$.

Sign: To sign a message m using $d_{ID} = (R, s)$, the signer proceeds as follows:

1. Pick a random $y \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute $Y = g^y$ as well as $S = g^s$.
2. Compute $c = G(m\|S\|Y) \in \mathbb{Z}_p$.
3. Calculate $z = y + cs \bmod p$.

The signature on m is (R, S, c, z).

Verify: To verify a signature (R, S, c, z) on m:

1. Compute $Y = g^z S^{-c}$ and reject the signature if $c \neq G(m||S||Y)$.
2. Compute $\ell = H(ID||R)$.
3. Accept the signature if $S = RX^{\ell}$.

Bellare et al. [2] proved the security of their scheme assuming that computing discrete logarithms is hard and when the hash functions H and G are modeled as random oracles.

A BNN signature can be seen as a noninteractive proof of knowledge of a Schnorr signature in the same way as GQ signatures prove knowledge of an RSA signature. The above scheme can also be seen as an optimization of the generic construction described in section "Generic IBS from Any Signature" it indeed provides slightly shorter signatures using suitable parameters such as carefully chosen elliptic-curve subgroups. The BNN IBS was recently further optimized [19] to provide shorter signatures and also inspired an IBS scheme [24] supporting partial signature aggregation.

Other IBS Schemes Based on Specific Number Theoretic Assumptions

The GQ and BNN identity-based signatures are far from being the only IBS systems where signatures consist of a noninteractive proof of knowledge of an ordinary signature. Indeed, many other proposals based on the discrete logarithm problem [7, 19, 28], RSA [28], factoring [15, 17, 29], or groups with bilinear maps [1, 10, 24, 26, 30] can be found in the literature (see [2] for a comprehensive survey of these). Recent works [11, 12] also investigated how to efficiently base IBS schemes on assumptions stemming from coding theory.

It has been reported that identity-based signatures can be endowed with specific additional properties. In many cases, these can be generically obtained [18] by extending the generic construction of section "Generic IBS from Any

Signature". Other IBS schemes supporting signature aggregation [20, 24] or multiple signers [4] were designed under specific assumptions.

In addition, an observation made by Naor (and reported in [8]) implies that identity-based signatures can also be derived from hierarchical identity-based encryption (HIBE) [21, 27]. Following this observation, Paterson and Schuldt [31] proved the security (without using the random oracle model) of a scheme derived from a 2-level extension of the Waters IBE [37] under the Diffie–Hellman assumption in groups with a bilinear map.

Identity-Based Encryption from Quadratic Residuosity: The Cocks IBE

An identity-based encryption scheme (IBE) consists of a tuple of algorithms $(Setup, Keygen, Encrypt, Decrypt)$, the first two ones of which have the same functionalities as in identity-based signatures. Algorithm *Encrypt* takes as input a plaintext m, the master public key mpk, and a receiver's identity ID to output a ciphertext $C = Encrypt(m, \mathsf{mpk}, ID)$. On input of C and the private key d_{ID}, the corresponding decryption algorithm outputs either a plaintext m or a special symbol $\perp$ indicating that the ciphertext is invalid.

The appropriate definition of security for IBE schemes was given by Boneh and Franklin [8].

Definition 1 *An IBE scheme is said to be adaptively chosen-ciphertext secure (IND-ID-CCA) if no probabilistic polynomial time (PPT) adversary has a non-negligible advantage in the following game.*

1. *The challenger runs the* **Setup** *algorithm on input of a security parameter k and sends the domain-wide parameters* mpk *to the adversary* $\mathcal{A}$.
2. *In a find stage,* $\mathcal{A}$ *starts probing the following oracles:*
   - *Key extraction oracle: Given an identity ID, this oracle returns the extracted private key* $d_{ID} = \mathbf{Keygen}(\mathsf{msk}, ID)$.
   - *Decryption oracle: Given an identity* $ID \in \{0,1\}^*$ *and a ciphertext C, it generates the private key* d_{ID} *associated to ID and returns either a plaintext* $M \in \mathcal{M}$ *or a distinguished symbol* $\perp$ *indicating that the ciphertext was not correctly formed.*

   $\mathcal{A}$ *can present her queries adaptively in the sense that each query may depend on the answers to previous ones. At some point, she produces two plaintexts* $M_0, M_1 \in \mathcal{M}$, *and a target identity* ID^* *for which she has not requested the private key in stage 2. The challenger computes* $C = \mathbf{Encrypt}(M_b, \mathsf{mpk}, ID^*)$, *for a random hidden bit* $b \overset{R}{\leftarrow} \{0,1\}$, *which is sent to* $\mathcal{A}$.
3. *In the guess stage,* $\mathcal{A}$ *asks new queries as in the find stage but is restricted not to issue a key extraction request on the target identity* ID^* *and cannot submit C to the decryption oracle for the identity* ID^*. *Eventually,* $\mathcal{A}$ *outputs a bit* b' *and wins if* $b' = b$.

$\mathcal{A}$*'s advantage is defined as* $Adv(\mathcal{A}) := |2 \times Pr[b' = b] - 1|$.

The above definition captures the chosen-ciphertext scenario where the adversary is granted access to a decryption oracle throughout the game. There exists a weaker definition, called *chosen-plaintext* security (or IND-ID-CPA for short), where no such decryption oracle is given to the adversary.

The IBE scheme proposed by Cocks in 2001 [13] is based on quadratic *residues* and on the properties of the Legendre and Jacobi symbols for Blum integers (i.e., composite integers $n = pq$, where p and q are primes such that $p \equiv q \equiv 3 \pmod 4$). It is made of the four algorithms depicted below.

Setup: The PKG picks prime numbers p and q such that $p \equiv q \equiv 3 \pmod 4$, computes their product $n = pq$, and chooses a hash function $H : \{0,1\}^* \to \mathbb{Z}_n^*$. The PKG's master secret key is defined to be $\mathsf{msk} = (p, q)$, and the master public key consists of $\mathsf{mpk} = (n, H)$.

Keygen: Given an identity ID, the PKG computes a sequence of hash values starting from ID until obtaining $a = H(H(H\ldots(ID))) \in \mathbb{Z}_n^*$ such that $\left(\frac{a}{n}\right) = 1$. For such a $a \in \mathbb{Z}_n^*$, either a or $-a$ is a square in $\mathbb{Z}_n^*$. It is easy to verify that $r = a^{\frac{n+5-(p+q)}{8}} \bmod n$ satisfies $a = r^2 \bmod n$ or $a = -r^2 \bmod n$ depending on whether $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = 1$ or $\left(\frac{a}{p}\right) = \left(\frac{a}{q}\right) = -1$. The obtained r is returned to the user as a private key.

Encrypt: The sender does not know which of a or $-a$ is a square in $\mathbb{Z}_n^*$ and one first considers the case $a = r^2 \bmod n$. The sender generates a symmetric transport key K and encrypts the plaintext M with it. Each bit x of that symmetric key is then encrypted before being sent to the receiver B. To do this, A encodes x in $\{-1, 1\}$ rather than in $\{0, 1\}$ and does the following.

1. Pick a random $t \in \mathbb{Z}_n^*$ such that $\left(\frac{t}{n}\right) = x$.
2. Compute $s = \left(t + \frac{a}{t}\right) \bmod n$ (since $\left(\frac{t}{n}\right) \neq 0$, t is coprime with p and q and thus invertible in $\mathbb{Z}_n$) and send it to B.

Since A does not know which of a or $-a$ is the square of B's decryption key, A has to repeat the above process for a new t and, this time, send $s = (t - a/t) \bmod n$. Hence,

$2|n|$ bits, where $|x|$ denotes the bitlength of x, have to be transmitted for each bit of the symmetric key.

Decrypt: B recovers x as follows. Given that

$$t(1+r/t)^2 \equiv t+2r+\frac{r^2}{t} \equiv t+2r+\frac{a}{t} \equiv s+2r \pmod{n},$$

B can compute $\left(\frac{s+2r}{n}\right) = \left(\frac{t}{n}\right) = x$ and recover x using his/her private key r thanks to the multiplicative properties of the Jacobi symbol. Once the symmetric key K is obtained in clear, the ciphertext can be decrypted.

For 128-bit symmetric keys, the scheme is fairly cheap from a computational standpoint: the sender's cost is dominated by 2 × 128 Jacobi symbol evaluations and 2 × 128 modular inversions. The receiver just has to compute 128 Jacobi symbols since he/she knows which of a or $-a$ is the square of his/her private key. The drawback of the scheme is its bandwidth overhead: for a 1024-bit modulus n and a 128-bit symmetric transport key, at least 2 × 16 Kb need to be transmitted if all the integers s are sent together.

Cocks showed that his construction is secure (in the random oracle model) against chosen-plaintext attacks under the Quadratic Residuosity Assumption (i.e., the hardness of deciding whether or not a random integer a such that $\left(\frac{a}{n}\right) = 1$ is a square). Chosen-ciphertext security can be acquired via several generic transformations such as the one of Fujisaki and Okamoto [16].

While elegant, Cocks's construction is somewhat bandwidth-demanding. In addition, separately encrypting each bit of plaintext makes it difficult to turn the scheme into a chosen-ciphertext secure multi-bit encryption scheme. In 2007, Boneh et al. [9] showed how to construct a multi-bit quadratic-residuosity-based IBE scheme featuring much shorter ciphertexts. Other IBE systems avoiding the limitation of Cocks's proposal will be covered in the chapter dedicated to identity-based encryption.

References

1. Barreto PSLM, Libert B, McCullagh N, Quisquater JJ (2005) Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Advances in cryptology – Asiacrypt '05. Lecture notes in computer science, vol 3788. Springer, Heidelberg, pp 515–532
2. Bellare M, Namprempre C, Neven G (2004) Security proofs for identity-based identification and signature schemes. In: Advances in cryptology – Eurocrypt '04. Lecture notes in computer science, vol 3027. Springer, Heidelberg, pp 268–286
3. Bellare M, Namprempre C, Pointcheval D, Semanko M (2001) The power of RSA inversion oracles and the security of Chaum's RSA-based blind signature scheme. In: Financial cryptography 2001. Lecture notes in computer science, vol 2339. Springer, Heidelberg, pp 309–328
4. Bellare M, Neven G (2007) Identity-based multi-signatures from RSA. In: RSA conference cryptographers' track (CT-RSA '07). Lecture notes in computer science, vol 4377. Springer, Heidelberg, pp 145–162
5. Bellare M, Palacio A (2002) GQ and schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks. In: Advances in cryptology – Crypto '02, Lecture notes in computer science, vol 2442. Springer, Heidelberg, pp 162–177
6. Bellare M, Rogaway P (1993) Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM conference on computer and communications security, Fairfax, pp 62–73
7. Beth T (1988) Efficient zero-knowledge identification scheme for smart cards. In: Advances in cryptology – Eurocrypt '88. Lecture notes in computer science, vol 330. Springer, Heidelberg, pp 77–84
8. Boneh D, Franklin M (2001) Identity based encryption from the Weil pairing, SIAM J of Comput 32(3): 586–615, 2003. Earlier version in advances in cryptology – Crypto '01. Lecture notes in computer science, vol 2139. Springer, Heidelberg, pp 213–229
9. Boneh D, Gentry C, Hamburg M (2007) Space-efficient identity-based encryption without pairings. In: Proceedings of the FOCS '07, Providence, pp 647–657
10. Cha JC, Cheon JH (2003) An identity-based signature from gap Diffie-Hellman groups. In: Public Key Cryptography 2003 (PKC '03). Lecture notes in computer science, vol 2567. Springer, Heidelberg, pp 18–30
11. Cayrel PL, Gaborit P, Girault M (2007) Identity-based identification and signature schemes using correcting codes. In: Workshop of cryptography and coding 2007, Versailles
12. Cayrel PL, Gaborit P, Galindo D, Girault M (2009) Improved identity-based identification using correcting codes. In: Computing Research Repository (CoRR) abs/0903.0069
13. Cocks C (2001) An identity based encryption scheme based on quadratic residues. In: Proceedings of cryptography and coding. Lecture notes in computer science, vol 2260. Springer, Heidelberg, pp 360–363
14. Dodis Y, Katz J, Xu S, Yung M (2003) Strong key-insulated signature schemes. In: Public key cryptography 2003 (PKC '03). Lecture notes in computer science, vol 2567. Springer, Heidelberg, pp 130–144
15. Fiat A, Shamir A (1986) How to prove yourself: practical solutions to identification and signature problems. In: Advances in cryptology – Crypto '86. Lecture notes in computer science, vol 263. Springer, Heidelberg, pp 186–194
16. Fujisaki E, Okamoto T (1999) Secure integration of asymmetric and symmetric encryption schemes. In: Advances in cryptology – Crypto '99. Lecture notes in computer science, vol 1666. Springer, Heidelberg, pp 537–554
17. Fischlin M, Fischlin R (2002) The representation problem based on factoring. In: RSA conference cryptographers' track (CT-RSA '02). Lecture notes in computer science, vol 2271. Springer, Heidelberg, pp 96–113
18. Galindo D, Herranz J, Kiltz E (2006) On the generic construction of identity-based signatures with additional properties, In: Avances in cryptology – Asiacrypt '06. Lecture notes in computer science, vol 4284. Springer, Heidelberg, pp 178–193

19. Galindo D, Garcia FD (2009) A schnorr-like lightweight identity-based signature scheme. In: Progress in cryptology – Africacrypt '09. Lecture notes in computer science, vol 5580. pp 135–148
20. Gentry C, Ramzan Z (2006) Identity-based aggregate signatures. In: Public key cryptography 2006 (PKC '06). Lecture notes in computer science, vol 3958. Springer, Heidelberg, pp 257–273
21. Gentry C, Silverberg A (2002) Hierarchical ID-based cryptography. In: Advances in cryptology – Asiacrypt '02. Lecture notes in computer science, vol 2501. Springer, Heidelberg, pp 548–566
22. Goldwasser S, Micali S, Rivest R (1998) A digital signature scheme secure against adaptive chosen-message attacks. SIAM J Comput 17(2):281–308
23. Guillou L, Quisquater JJ (1998) A "Paradoxical" identity-based signature scheme resulting from zero-knowledge. In: Advances in cryptology – Crypto '88. Lecture notes in computer science, vol 403. Springer, Heidelberg, pp 216–231
24. Herranz J (2006) Deterministic identity-based signatures for partial aggregation. Comput J 49(3):322–330
25. Kiltz E, Mityagin A, Panjwani S, Raghavan B (2005) Append-only signatures. In: International colloquium automata, languages and programming (ICALP '05). Lecture notes in computer science, vol 3580. Springer, Heidelberg, pp 434–445
26. Hess F (2003) Efficient identity based signature schemes based on pairings. In: Proceedings of SAC '02. Lecture notes in computer science, vol 2595. Springer, Heidelberg, pp 310–324
27. Horwitz J, Lynn B (2002) Toward hierarchical identity-based encryption. In: Advances in cryptology – Eurocrypt '02. Lecture notes in computer science, vol 2332. Springer, Heidelberg, pp 466–481
28. Okamoto T (1992) Provably secure and practical identification schemes and corresponding signature schemes. In: Advances in cryptology – Crypto '92. Lecture notes in computer science, vol 740. Springer, Heidelberg, pp 31–53
29. Ong H, Schnorr CP (1990) Fast signature generation with a fiat shamir-like scheme. In: Advances in cryptology – Eurocrypt '90. Lecture notes in computer science, vol 473. Springer, Heidelberg, pp 432–440
30. Paterson KG (2002) ID-based signatures from pairings on elliptic curves. Available at http://eprint.iacr.org/2002/004/
31. Paterson KG, Schuldt J (2006) Efficient Identity-based signatures secure in the standard model. In: 11th Australasian conference on information security and privacy (ACISP '06). Lecture notes in computer science, vol 4058. Springer, Heidelberg, pp 207–222, 387–398
32. Pointcheval D, Stern J (1996) Security proofs for signature schemes. In: Advances in cryptology – Eurocrypt '96. Lecture notes in computer science, vol 1070. Springer, Heidelberg, pp 387–398
33. Pointcheval D, Stern J (2000) Security arguments for digital signatures and blind signatures. J Cryptol 13(3):361–396
34. Sakai R, Ohgishi K, Kasahara M (2000) Cryptosystems based on pairing. In: The 2000 symposium on cryptography and information security, Okinawa, Japan
35. Schnorr CP (1989) Efficient identification and signatures for smart cards. In: Advances in cryptology – Crypto '89. Lecture notes in computer science, vol 435. Springer, Heidelberg, pp 239–252
36. Shamir A (1984) Identity based cryptosystems and signature schemes. In: Advances in cryptology – Crypto '84. Lecture notes in computer science, vol 196. Springer, Heidelberg
37. Waters B (2005) Efficient identity-based encryption without random oracles. In: Advances in cryptology – Eurocrypt 2005. Lecture notes in computer science, vol 2567. Springer, Heidelberg, pp 114–127