IEEE Draft P1363.3

Identity Based Public Key Cryptography Based On Pairings

Daniel Schliebner

14. Dezember 2009

Gliederung

About The Headline

Identity Based Public Key Cryptography

- Is a type of **public-key cryptography**.
- Difference: the **public-key** is some unique information `ID` about the **identity** of a user.
- Proposed by Adi Shamir for the first time in 1984 (see [ASha84] page 47–53).
- **Paring**: mapping between groups (correct definition later).
- Used for: **encryption (IBE)**, key encapsulation, signatures (and combinations).

About The Headline

Identity Based Public Key Cryptography

- Is a type of **public-key cryptography**.
- Difference: the **public-key** is some unique information `ID` about the **identity** of a user.
- Proposed by Adi Shamir for the first time in 1984 (see [ASha84] page 47–53).
- **Paring**: mapping between groups (correct definition later).
- Used for: **encryption (IBE)**, key encapsulation, signatures (and combinations).

About The Headline

Identity Based Public Key Cryptography

- Is a type of **public-key cryptography**.
- Difference: the **public-key** is some unique information `ID` about the **identity** of a user.
- Proposed by Adi Shamir for the first time in 1984 (see [ASha84] page 47–53).
- **Paring**: mapping between groups (correct definition later).
- Used for: **encryption (IBE)**, key encapsulation, signatures (and combinations).

About The Headline

Identity Based Public Key Cryptography

- Is a type of **public-key cryptography**.
- Difference: the **public-key** is some unique information `ID` about the **identity** of a user.
- Proposed by Adi Shamir for the first time in 1984 (see [ASha84] page 47–53).
- **Paring**: mapping between groups (correct definition later).
- Used for: **encryption (IBE)**, key encapsulation, signatures (and combinations).

About The Headline

Identity Based Public Key Cryptography

- Is a type of **public-key cryptography**.
- Difference: the **public-key** is some unique information `ID` about the **identity** of a user.
- Proposed by Adi Shamir for the first time in 1984 (see [ASha84] page 47–53).
- **Paring**: mapping between groups (correct definition later).
- Used for: **encryption (IBE)**, key encapsulation, signatures (and combinations).

About The Paper

IEEE Draft P1363.3

- **IEEE**: Institute of Electrical and Electronics Engineers.
- **P1363**: Standardization project for public-key cryptography.
- **.3**: Specifications for identity-based public-key cryptography using pairings.
- Chair of working group (as of Oct. 08): William Whyte (NTRU Cryptosystems Inc).

About The Paper

IEEE Draft P1363.3

- **IEEE**: Institute of Electrical and Electronics Engineers.
- **P1363**: Standardization project for public-key cryptography.
- **.3**: Specifications for identity-based public-key cryptography using pairings.
- Chair of working group (as of Oct. 08): William Whyte (NTRU Cryptosystems Inc).

About The Paper

IEEE Draft P1363.3

- **IEEE**: Institute of Electrical and Electronics Engineers.
- **P1363**: Standardization project for public-key cryptography.
- **.3**: Specifications for identity-based public-key cryptography using pairings.
- Chair of working group (as of Oct. 08): William Whyte (NTRU Cryptosystems Inc).

About The Paper

IEEE Draft P1363.3

- **IEEE**: Institute of Electrical and Electronics Engineers.
- **P1363**: Standardization project for public-key cryptography.
- **.3**: Specifications for identity-based public-key cryptography using pairings.
- Chair of working group (as of Oct. 08): William Whyte (NTRU Cryptosystems Inc).

Pairing: a mathematical formalism

Definition

Let $(G_1, +), (G_2, +), (G_3, \cdot)$ be groups of prime order $p \in \mathbb{N}$. A pairing is a $\mathbb{Z}_p$–bilinear map

$$e : G_1 \times G_2 \longrightarrow G_3$$

between $\mathbb{Z}_p$–modules for which the following holds:

1. e is non-degenerated
   (i. e. $\forall P \neq 0_{G_1} \in G_1, Q \neq 0_{G_2} \in G_2 : \ e(P, Q) \neq 1_{G_3}$).
2. e is computable in an efficient manner.

Now: ID-based cryptography using the example of ID-based encryption.

Identity Based Encryption (IBE)

The Participants

Participants are:

- **Sender** A (want's to send message m).
- **Receiver** B (has an identification `ID`, e. g. `bob@domain.com`).
- A Trusted Third Party – the Private Key Generator (**PKG**).

Identity Based Encryption (IBE)

The Participants

Participants are:

- **Sender** A (want's to send message m).
- **Receiver** B (has an identification `ID`, e. g. `bob@domain.com`).
- A Trusted Third Party – the Private Key Generator (**PKG**).

Identity Based Encryption (IBE)

The Participants

Participants are:

- **Sender** A (want's to send message m).
- **Receiver** B (has an identification `ID`, e. g. `bob@domain.com`).
- A Trusted Third Party – the Private Key Generator (**PKG**).

Identity Based Encryption (IBE)

The Algorithms Of An IBE-Protocol

Algorithms of an IBE-Protocol are:

- **Setup**: run by the PKG. Returns:
  - $\mathcal{P}$ – a set of public parameters.
  - s – the master key (or secret server key).
- **Extract**$(\mathcal{P}, s, ID)$: run by the PKG (when B requests his private key). Returns:
  - K_{ID} – the private key corresponding to `ID`.
- **Encrypt**$(\mathcal{P}, ID, m)$: run by A. Returns:
  - c – the encrypted plaintext m.
- **Decrypt**$(\mathcal{P}, ID, c)$: run by B. Returns:
  - m – the decrypted ciphertext c.

Identity Based Encryption (IBE)

The Algorithms Of An IBE-Protocol

Algorithms of an IBE-Protocol are:

- **Setup**: run by the PKG. Returns:
  - $\mathcal{P}$ – a set of public parameters.
  - s – the master key (or secret server key).
- **Extract**$(\mathcal{P}, s, ID)$: run by the PKG (when B requests his private key). Returns:
  - K_{ID} – the private key corresponding to `ID`.
- **Encrypt**$(\mathcal{P}, ID, m)$: run by A. Returns:
  - c – the encrypted plaintext m.
- **Decrypt**$(\mathcal{P}, ID, c)$: run by B. Returns:
  - m – the decrypted ciphertext c.

Identity Based Encryption (IBE)

The Algorithms Of An IBE-Protocol

Algorithms of an IBE-Protocol are:

- **Setup**: run by the PKG. Returns:
  - $\mathcal{P}$ – a set of public parameters.
  - s – the master key (or secret server key).
- **Extract**$(\mathcal{P}, s, ID)$: run by the PKG (when B requests his private key). Returns:
  - K_{ID} – the private key corresponding to `ID`.
- **Encrypt**$(\mathcal{P}, ID, m)$: run by A. Returns:
  - c – the encrypted plaintext m.
- **Decrypt**$(\mathcal{P}, ID, c)$: run by B. Returns:
  - m – the decrypted ciphertext c.

Identity Based Encryption (IBE)

The Algorithms Of An IBE-Protocol

Algorithms of an IBE-Protocol are:

- **Setup**: run by the PKG. Returns:
  - $\mathcal{P}$ – a set of public parameters.
  - s – the master key (or secret server key).
- **Extract**$(\mathcal{P}, s, ID)$: run by the PKG (when B requests his private key). Returns:
  - K_{ID} – the private key corresponding to `ID`.
- **Encrypt**$(\mathcal{P}, ID, m)$: run by A. Returns:
  - c – the encrypted plaintext m.
- **Decrypt**$(\mathcal{P}, ID, c)$: run by B. Returns:
  - m – the decrypted ciphertext c.

Identity Based Encryption (IBE)

The Algorithms Of An IBE-Protocol (cont.)

Summarization:

- PKG runs **Setup**$() \longrightarrow (\mathcal{P}, s)$.
- PKG runs **Extract**$(\mathcal{P}, s, ID) \longrightarrow K_{\mathrm{ID}}$.
- A runs **Encrypt**$(\mathcal{P}, ID, m) \longrightarrow c$.
- B runs **Decrypt**$(\mathcal{P}, ID, c) \longrightarrow m$.

Identity Based Encryption (IBE)

Primitives

- **Primitives**: contain basic mathematical operations (building blocks for an IBE-protocol).
- **Generation**: used to extract a private key at a PKG.
- **Verification**: verification of K_{ID} by receiver B.
- **Encrypt** and **Decrypt**: used inside corresponding algorithms.

Identity Based Encryption (IBE)

Primitives

- **Primitives**: contain basic mathematical operations (building blocks for an IBE-protocol).
- **Generation**: used to extract a private key at a PKG.
- **Verification**: verification of K_{ID} by receiver B.
- **Encrypt** and **Decrypt**: used inside corresponding algorithms.

Identity Based Encryption (IBE)

Primitives

- **Primitives**: contain basic mathematical operations (building blocks for an IBE-protocol).
- **Generation**: used to extract a private key at a PKG.
- **Verification**: verification of K_{ID} by receiver B.
- **Encrypt** and **Decrypt**: used inside corresponding algorithms.

Identity Based Encryption (IBE)

Primitives

- **Primitives**: contain basic mathematical operations (building blocks for an IBE-protocol).
- **Generation**: used to extract a private key at a PKG.
- **Verification**: verification of K_{ID} by receiver B.
- **Encrypt** and **Decrypt**: used inside corresponding algorithms.

Gliederung

BB_1IBE

- By **Dan Boneh** and **Xavier Boyen**.
- Security bases on **Bilinear-Diffie-Hellman** (BDH) Problem.
- As formulated in the previous section, the protocol consists of four algorithms: Setup, Extract, Encrypt and Decrypt.
- To this end, let G_1, G_2, G_3 be groups of prime order $p \in \mathbb{N}$ and $e : G_1 \times G_2 \longrightarrow G_3$ a pairing.
- Let $\mathrm{ID} \in \{0,1\}^*$ and plaintext $m \in \{0,1\}^n$.

BB_1IBE

- By **Dan Boneh** and **Xavier Boyen**.
- Security bases on **Bilinear-Diffie-Hellman** (BDH) Problem.
- As formulated in the previous section, the protocol consists of four algorithms: Setup, Extract, Encrypt and Decrypt.
- To this end, let G_1, G_2, G_3 be groups of prime order $p \in \mathbb{N}$ and $e : G_1 \times G_2 \longrightarrow G_3$ a pairing.
- Let $\mathrm{ID} \in \{0,1\}^*$ and plaintext $m \in \{0,1\}^n$.

BB_1IBE

- By **Dan Boneh** and **Xavier Boyen**.
- Security bases on **Bilinear-Diffie-Hellman** (BDH) Problem.
- As formulated in the previous section, the protocol consists of four algorithms: Setup, Extract, Encrypt and Decrypt.
- To this end, let G_1, G_2, G_3 be groups of prime order $p \in \mathbb{N}$ and $e : G_1 \times G_2 \longrightarrow G_3$ a pairing.
- Let $\mathrm{ID} \in \{0,1\}^*$ and plaintext $m \in \{0,1\}^n$.

BB_1IBE

- By **Dan Boneh** and **Xavier Boyen**.
- Security bases on **Bilinear-Diffie-Hellman** (BDH) Problem.
- As formulated in the previous section, the protocol consists of four algorithms: Setup, Extract, Encrypt and Decrypt.
- To this end, let G_1, G_2, G_3 be groups of prime order $p \in \mathbb{N}$ and $e : G_1 \times G_2 \longrightarrow G_3$ a pairing.
- Let ID $\in \{0,1\}^*$ and plaintext $m \in \{0,1\}^n$.

BB_1IBE

- By **Dan Boneh** and **Xavier Boyen**.
- Security bases on **Bilinear-Diffie-Hellman** (BDH) Problem.
- As formulated in the previous section, the protocol consists of four algorithms: Setup, Extract, Encrypt and Decrypt.
- To this end, let G_1, G_2, G_3 be groups of prime order $p \in \mathbb{N}$ and $e : G_1 \times G_2 \longrightarrow G_3$ a pairing.
- Let $\mathrm{ID} \in \{0,1\}^*$ and plaintext $m \in \{0,1\}^n$.

BB_1IBE

Setup

- PKG chooses a master key
  $s := (s_1, s_2, s_3) \in_R \mathbb{Z}_p^* \times \mathbb{Z}_p^* \times \mathbb{Z}_p^*$.
- PKG generates public parameter
  $\mathcal{P} := (Q_1, Q_2, R, T, V, G_1, G_2, e)$, where
  - Q_i is a generator of G_i, $i = 1, 2$, i. e. $\langle Q_i \rangle = G_i$,
  - $R := s_1 Q_1$,
  - $T := s_3 Q_1$,
  - $V := e(R, s_2 Q_2)$.

BB_1IBE

Setup

- PKG chooses a master key
  $s := (s_1, s_2, s_3) \in_R \mathbb{Z}_p^* \times \mathbb{Z}_p^* \times \mathbb{Z}_p^*$.
- PKG generates public parameter
  $\mathcal{P} := (Q_1, Q_2, R, T, V, G_1, G_2, e)$, where
  - Q_i is a generator of G_i, $i = 1, 2$, i. e. $\langle Q_i \rangle = G_i$,
  - $R := s_1 Q_1$,
  - $T := s_3 Q_1$,
  - $V := e(R, s_2 Q_2)$.

BB_1IBE

The primitives (knowing $\mathcal{P}$ and s)

Generation: `P-BB1-G`(M)

- $r_0 \in_R \mathbb{Z}_p^*$.
- $i := s_1 s_2 + r_0(s_1 M + s_3)$.
- **return** $(iQ_2, r_0 Q_2)$.

Encryption: `P-BB1-E`(r)

- $E_0 := rQ_1$.
- $E_1 := (rM)R + rT$.
- $B := V^r$.
- **return** (E_0, E_1, B).

Decryption: `P-BB1-D`$(E_0, E_1, (K_{0,M}, K_{1,M}))$

- **return** $e(E_0, K_{0,M}) \cdot e(E_1, K_{1,M})^{-1}$.

BB_1IBE

The primitives (knowing $\mathcal{P}$ and s)

Generation: `P-BB1-G`(M)

- $r_0 \in_R \mathbb{Z}_p^*$.
- $i := s_1 s_2 + r_0(s_1 M + s_3)$.
- **return** $(iQ_2, r_0 Q_2)$.

Encryption: `P-BB1-E`(r)

- $E_0 := rQ_1$.
- $E_1 := (rM)R + rT$.
- $B := V^r$.
- **return** (E_0, E_1, B).

Decryption: `P-BB1-D`$(E_0, E_1, (K_{0,M}, K_{1,M}))$

- **return** $e(E_0, K_{0,M}) \cdot e(E_1, K_{1,M})^{-1}$.

BB_1IBE

The primitives (knowing $\mathcal{P}$ and s)

Generation: `P-BB1-G`(M)

- $r_0 \in_R \mathbb{Z}_p^*$.
- $i := s_1 s_2 + r_0(s_1 M + s_3)$.
- **return** $(iQ_2, r_0 Q_2)$.

Encryption: `P-BB1-E`(r)

- $E_0 := rQ_1$.
- $E_1 := (rM)R + rT$.
- $B := V^r$.
- **return** (E_0, E_1, B).

Decryption: `P-BB1-D`$(E_0, E_1, (K_{0,M}, K_{1,M}))$

- **return** $e(E_0, K_{0,M}) \cdot e(E_1, K_{1,M})^{-1}$.

BB_1IBE

- There are three algorithms left.
- Using the three primitives `P-BB1-G`, `P-BB1-E`, `P-BB1-D` we can now formulate them.
- Therefore: consider three Hashfunctions

$$\begin{aligned} H_1 &: \{0,1\}^* \longrightarrow \mathbb{Z}_p^* \\ H_2 &: G_3 \longrightarrow \{0,1\}^n \\ H_3 &: G_3 \times \{0,1\}^n \times G_1 \times G_1 \longrightarrow \mathbb{Z}_p^* \end{aligned}$$

BB_1IBE

- There are three algorithms left.
- Using the three primitives `P-BB1-G`, `P-BB1-E`, `P-BB1-D` we can now formulate them.
- Therefore: consider three Hashfunctions

$$\begin{aligned} H_1 &: \{0,1\}^* \longrightarrow \mathbb{Z}_p^* \\ H_2 &: G_3 \longrightarrow \{0,1\}^n \\ H_3 &: G_3 \times \{0,1\}^n \times G_1 \times G_1 \longrightarrow \mathbb{Z}_p^* \end{aligned}$$

BB_1IBE

- There are three algorithms left.
- Using the three primitives `P-BB1-G`, `P-BB1-E`, `P-BB1-D` we can now formulate them.
- Therefore: consider three Hashfunctions

$$\begin{aligned} H_1 &: \{0,1\}^* \longrightarrow \mathbb{Z}_p^* \\ H_2 &: G_3 \longrightarrow \{0,1\}^n \\ H_3 &: G_3 \times \{0,1\}^n \times G_1 \times G_1 \longrightarrow \mathbb{Z}_p^* \end{aligned}$$

BB_1IBE

Extract

- $M := H_1(\mathrm{ID})$.
- $K_{\mathrm{ID}} := (K_{0,M}, K_{1,M}) \longleftarrow \texttt{P-BB1-G}(M)$.

K_{ID} is the private key for the receiver.

BB_1IBE

Extract

- $M := H_1(\mathrm{ID})$.
- $K_{\mathrm{ID}} := (K_{0,M}, K_{1,M}) \longleftarrow \texttt{P-BB1-G}(M)$.

K_{ID} is the private key for the receiver.

BB_1IBE

Encrypt

- $r \in_R \mathbb{Z}_p^*$.
- $(B, E_0, E_1) \longleftarrow \mathsf{P\text{-}BB1\text{-}E}(r)$.
- $Y := H_2(B) \oplus m$.
- $t := r + H_3(B, Y, E_0, E_1)$.
- $c := (Y, E_0, E_1, t)$.

c is the ciphertext. B is called **blinding factor**.

BB_1IBE

Encrypt

- $r \in_R \mathbb{Z}_p^*$.
- $(B, E_0, E_1) \longleftarrow \mathsf{P\text{-}BB1\text{-}E}(r)$.
- $Y := H_2(B) \oplus m$.
- $t := r + H_3(B, Y, E_0, E_1)$.
- $c := (Y, E_0, E_1, t)$.

c is the ciphertext. B is called **blinding factor**.

BB_1IBE

Encrypt

- $r \in_R \mathbb{Z}_p^*$.
- $(B, E_0, E_1) \longleftarrow \mathsf{P\text{-}BB1\text{-}E}(r)$.
- $Y := H_2(B) \oplus m$.
- $t := r + H_3(B, Y, E_0, E_1)$.
- $c := (Y, E_0, E_1, t)$.

c is the ciphertext. B is called **blinding factor**.

BB_1IBE

Encrypt

- $r \in_R \mathbb{Z}_p^*$.
- $(B, E_0, E_1) \longleftarrow \mathsf{P\text{-}BB1\text{-}E}(r)$.
- $Y := H_2(B) \oplus m$.
- $t := r + H_3(B, Y, E_0, E_1)$.
- $c := (Y, E_0, E_1, t)$.

c is the ciphertext. B is called **blinding factor**.

BB_1IBE

Encrypt

- $r \in_R \mathbb{Z}_p^*$.
- $(B, E_0, E_1) \longleftarrow \mathsf{P\text{-}BB1\text{-}E}(r)$.
- $Y := H_2(B) \oplus m$.
- $t := r + H_3(B, Y, E_0, E_1)$.
- $c := (Y, E_0, E_1, t)$.

c is the ciphertext. B is called **blinding factor**.

BB_1IBE

Decrypt

- $B \longleftarrow \texttt{P-BB1-D}(E_0, E_1, K_{\mathrm{ID}})$.
- $r := t - H_3(B, Y, E_0, E_1)$.
- **if** $\neg(B == V^r$ **and** $E_0 == rQ_1)$ **then exit** with „error“.
- $m := Y \oplus H_2(B)$.

m is the plaintext.

BB_1IBE

Decrypt

- $B \longleftarrow \texttt{P-BB1-D}(E_0, E_1, K_{\mathrm{ID}})$.
- $r := t - H_3(B, Y, E_0, E_1)$.
- **if** $\neg(B == V^r$ **and** $E_0 == rQ_1)$ **then exit** with „error“.
- $m := Y \oplus H_2(B)$.

m is the plaintext.

BB_1IBE

Decrypt

- $B \longleftarrow \texttt{P-BB1-D}(E_0, E_1, K_{\mathrm{ID}})$.
- $r := t - H_3(B, Y, E_0, E_1)$.
- **if** $\neg(B == V^r$ **and** $E_0 == rQ_1)$ **then exit** with „error“.
- $m := Y \oplus H_2(B)$.

m is the plaintext.

BB_1IBE

Decrypt

- $B \longleftarrow \texttt{P-BB1-D}(E_0, E_1, K_{\mathrm{ID}})$.
- $r := t - H_3(B, Y, E_0, E_1)$.
- **if** $\neg(B == V^r$ **and** $E_0 == rQ_1)$ **then exit** with „error“.
- $m := Y \oplus H_2(B)$.

m is the plaintext.

Gliederung

BB_1IBE

Security

Definition

Let $e : G_1 \times G_2 \longrightarrow G_3$ be a pairing and $P \in G_1, Q \in G_2$. The Bilinear-Diffie-Hellman (BDH) Assumption says, that if P, Q, aP, bP, aQ, cQ for $a, b, c \in \mathbb{Z}_p^*$ are given, then it is hard to compute $e(P, Q)^{abc}$.

BB_1IBE

Security (cont.)

The security depends on

- Hashfunctions H_1, H_2, H_3.
- The secure channel between the receiver and the PKG.
- The BDH Assumption (at which point?).

BB_1IBE

Security (cont.)

The security depends on

- Hashfunctions H_1, H_2, H_3.
- The secure channel between the receiver and the PKG.
- The BDH Assumption (at which point?).

BB_1IBE

Security (cont.)

The security depends on

- Hashfunctions H_1, H_2, H_3.
- The secure channel between the receiver and the PKG.
- The BDH Assumption (at which point?).

BB_1IBE

Security (cont.)

Definition

Let $q \in \mathbb{Z}_p^*$. Then the q–Bilinear-Diffie-Hellman-Inverse (q-BDHI) Assumption says, that if $(P, aP, a^2P, \ldots, a^qP, Q, aQ, \ldots, a^qQ)$ are given, it is hard to compute $(e(P,Q)^a)^{-1}$.

Definition

We say, that the $(t, q, \varepsilon)-$BDHI Assumption holds, if no $t-$time algorithm $\mathcal{A}$ has advantage ε (i. e. $P(\mathcal{A}(P, \ldots, a^qP, Q, \ldots, a^qQ)) \geq \varepsilon$) in solving the q-BDHI problem.

BB_1IBE

Security (cont.)

Definition

Let $q \in \mathbb{Z}_p^*$. Then the q–Bilinear-Diffie-Hellman-Inverse (q-BDHI) Assumption says, that if $(P, aP, a^2P, \ldots, a^qP, Q, aQ, \ldots, a^qQ)$ are given, it is hard to compute $(e(P,Q)^a)^{-1}$.

Definition

We say, that the $(t, q, \varepsilon)-$BDHI Assumption holds, if no $t-$time algorithm $\mathcal{A}$ has advantage ε (i. e. $P(\mathcal{A}(P, \ldots, a^qP, Q, \ldots, a^qQ)) \geq \varepsilon$) in solving the q-BDHI problem.

BB_1IBE

Security (cont.)

Definition

We say, that an IBE system is $(t, q_{\mathrm{ID}}, \varepsilon)$-selective identity, chosen plaintext secure (short: $(t, q_{\mathrm{ID}}, \varepsilon)$–`IND-sID-CPA` secure) iff for every `IND-sID-CPA` adversary $\mathcal{A}$, that makes at most q_{ID} chosen private key queries, there is $\mathrm{ADV}_{\mathcal{A}} < \varepsilon$, where $\mathrm{ADV}_{\mathcal{A}}$ is the advantage of $\mathcal{A}$, attacking the IBE system.

BB_1IBE

Security (cont.)

Theorem

Suppose the (t, q, ε)-BDHI Assumption holds for G_1 and G_2. Then BB_1IBE is (t', q_S, ε)–IND-sID-CPA secure for any $q_S < q$ and any $t' < t - \Theta(\tau q^2)$, where τ is the maximum time for an exponentiation in G_1, G_2.

Proof: see [BB04]. □

Advantages

Advantages

- IBE eliminates the need for a public key distribution infrastructure.
- No key agreement.
- Interesting features (e. g. encode additional information into the `ID`: for instance expirations dates).

Disadvantages

Disadvantages

- PKG may decrypt and/or sign any message without authorisation.
- A secure channel is required between the PKG and the receiver.

That's it. Thank you for your attention.

Literaturverzeichnis

- [P1363308] IEEE P1636.3/D1 Draft Standard for Identity-based Public-key Cryptography Using Pairings.
  Working Group of the Microprocessor Standards Committee
- [ASha84] Identity-Based Cryptosystems and Signature Schemes. Advances in Cryptology: Proceedings of CRYPTO 84.
  Adi Shamir
  Lecture Notes in Computer Science 7, 1984

[BB04] Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles.
D. Boneh, X. Boyen
Advances in Cryptology – Eurocrypt, 2004, Springer-Verlag (2004), pp. 223–238.

[Wiki09] Wikipedia (DE, EN)
As of: 14. Dezember 2009.