CR01: Advanced Cryptographic Primitives

Lecture Notes

Benoît Libert
Scribe: Benjamin Hadjibeyli

September 8, 2014

Outline

1. Identity-Based Encryption (IBE) and bilinear maps
   - 3-partite key agreement
   - Short signatures
   - IBE (in the random oracle model)
2. IBE and signatures in the standard model
3. Applications of IBE: chosen-ciphertext security, forward-security encryption…
4. Attribute-Based Encryption (ABE)
5. Bilinear maps in composite order groups: homomorphic encryption for Disjunctive Normal Forms (DNFs) or inner-products
6. Learning-With-Errors (LWE) and public-key encryption
7. IBE from LWE
8. ABE from LWE
9. Fully Homomorphic Encryption (FHE)
10. Multi-Linear maps

1 Identity-Based Encryption and bilinear maps

1.1 Generalities

1.1.1 Diffie-Hellman Problems and the Diffie-Hellman Key Agreement Protocol

The Diffie-Hellman key exchange is a protocol allowing two parties to obtain shared cryptographic keys over a public communication channel.

Let $G = \langle g \rangle$ be a cyclic group of prime order $q > 2^\lambda$, where g is a generator of G and $\lambda \in \mathbb{N}$ the security parameter. Since G is generated by g, for any element $h \in G$, there exists a unique $x \in \mathbb{Z}_q$ such that $h = g^x$: x is called the discrete logarithm of h according to the generator g.

From now on, $x \xleftarrow{R} S$ denotes the fact that x is chosen uniformly at random from the set S. Finding the discrete logarithm can be described as a computational problem.

Definition 1 (Discrete Logarithm Problem)**.** *In a cyclic group $G = \langle g \rangle$ of order $q > 2^\lambda$, the Discrete Logarithm problem is, given $g \in G$ and $h = g^x$ with $x \xleftarrow{R} \mathbb{Z}_q$, to compute the exponent $x \in \mathbb{Z}_q$. We say that the Discrete Logarithm assumption holds in G if, for any probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$, we have*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{DL}}(\lambda) := \Pr[h = g^x \mid x \leftarrow \mathcal{A}(G, g, h) : h \xleftarrow{R} G] \in \mathsf{negl}(\lambda),$$

where the probability is taken over all coin tosses and $\mathsf{negl}(\lambda)$ denotes the set of negligible functions of $\lambda \in \mathbb{N}$.

Related to the problem of computing discrete logarithms are the so-called Diffie-Hellman problems. There are two important variants: we now present two assumptions based on the fact that these two problems are hard.

Definition 2 (Computational Diffie-Hellman assumption (CDH))**.** *The Computational Diffie-Hellman (CDH) assumption holds in G if no Probabilistic Polynomial-Time algorithm (PPT) can compute g^{ab} given (g, g^a, g^b), with $a, b \xleftarrow{R} \mathbb{Z}_q$. We say that the CDH assumption holds in G if, for any PPT algorithm $\mathcal{A}$, it holds that*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{CDH}}(\lambda) := \Pr[h = g^{ab} \mid h \leftarrow \mathcal{A}(G, g, g^a, g^b) : a, b \xleftarrow{R} \mathbb{Z}_q] \in \mathsf{negl}(\lambda),$$

where the probability is taken over all coin tosses.

Definition 3 (Decisional Diffie-Hellman assumption (DDH)). *The Decision Diffie-Hellman (DDH) assumption holds in G if the distributions*

$$D_0 = \{(g, g^a, g^b, g^{ab}) \mid a, b \overset{R}{\leftarrow} \mathbb{Z}_q\}$$

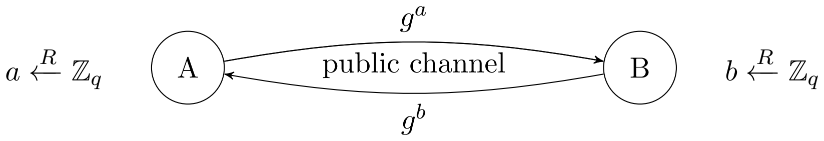
and

$$D_1 = \{(g, g^a, g^b, g^c) \mid a, b, c \overset{R}{\leftarrow} \mathbb{Z}_q\}$$

are computationally indistinguishable, i.e. for all PPT $\mathcal{A}$,

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{DDH}}(\lambda) = \big| \Pr_{x \leftarrow D_0}[\mathcal{A}(x) = 1] - \Pr_{x \leftarrow D_1}[\mathcal{A}(x) = 1]\big|$$
$$= \left|\Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = 1 \mid a, b \overset{R}{\leftarrow} \mathbb{Z}_q] - \Pr[\mathcal{A}(g, g^a, g^b, g^c) = 1 \mid a, b, c \overset{R}{\leftarrow} \mathbb{Z}_q]\right| \in \mathsf{negl}(\lambda),$$

The DDH assumption is the basis for the Diffie-Hellman key-agreement protocol, which allows two remote parties A and B to agree on a common random key while only communicating through a public channel. The key K should only be computable by A and B and look random to any adversary who passively observes the protocol run. This protocol consists in the following exchange: A picks $a \overset{R}{\leftarrow} \mathbb{Z}_q$ at random and computes g^a which is sent to B. Likewise, B picks $b \overset{R}{\leftarrow} \mathbb{Z}_q$ and computes g^b which is sent to A.


g^a
$a \overset{R}{\leftarrow} \mathbb{Z}_q$
A
public channel
B
$b \overset{R}{\leftarrow} \mathbb{Z}_q$
g^b

Then, A and B compute $K = g^{ab} = (g^a)^b = (g^b)^a$. To everyone but A and B, the key K is pseudo-random (i.e., computationally indistinguishable from a random element of the same group) under the DDH assumption. Importantly, the protocol is only secure against passive adversaries, who are not allowed to inject their own messages in the public channel.

1.1.2 ElGamal Encryption

The ElGamal key encryption scheme is based on the Diffie-Hellman key exchange protocol. Recall that a public-key encryption scheme is a tuple $(\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ of algorithms where only the last algorithm is deterministic. The ElGamal encryption scheme can be seen as a non-interactive Diffie-Hellman key exchange. It proceeds as follows:

- $\mathsf{KeyGen}(\lambda)$: choose a group G of prime order $q > 2^\lambda$, a generator g of G and $x \overset{R}{\leftarrow} \mathbb{Z}_q$.

  Then, output the public key PK and the secret key SK, which are defined as

  $$PK = \{G, q, g, X = g^x\} \text{ and } SK = \{x\}.$$

- $\mathsf{Encrypt}(M, PK)$: to encrypt $M \in G$, choose $r \overset{R}{\leftarrow} \mathbb{Z}_q$, and then compute

$$C = (C_1, C_2) = (g^r, M \cdot X^r).$$

- $\mathsf{Decrypt}(SK, c)$: compute and output $M = C_2/C_1^x = c_2/g^{rx} = C_2/X^r$.

The ElGamal encryption scheme can be shown to satisfy the notion of semantic security under the DDH assumption.

Definition 4 (Indistinguishability under chosen-plaintext attacks (IND-CPA))**.** *An encryption scheme is said to be semantically secure (or IND-CPA secure) iff no PPT adversary $\mathcal{A}$ has non-negligible advantage in the following game:*

1. *The challenger generates $(PK, SK) \leftarrow \mathsf{KeyGen}(\lambda)$ and gives the public key PK to the adversary $\mathcal{A}$;*
2. *$\mathcal{A}$ chooses M_0, M_1 of same length;*
3. *The challenger chooses $b \overset{R}{\leftarrow} \{0,1\}$ and gives $c = \mathsf{Encrypt}(M_b, PK)$ to $\mathcal{A}$;*
4. *$\mathcal{A}$ outputs $b' \in \{0,1\}$ and wins iff $b' = b$.*

The scheme is secure if the adversary $\mathcal{A}$ cannot win with significantly better probability than $1/2$. In particular, it must hold that

$$\mathbf{Adv}_{\mathcal{A}}(\lambda) = \left| \Pr[b = b'] - \frac{1}{2} \right| = \frac{1}{2} \left| \Pr[b' = 1|b = 1] - \Pr[b' = 1|b = 0] \right| \in \mathsf{negl}(\lambda),$$

where the probabilities are taken over the random coins of the challenger and those of the adversary.

Note: Algorithm $\mathsf{Encrypt}$ has to be probabilistic, otherwise, $\mathcal{A}$ could just run it and verify if $C = \mathsf{Encrypt}(M_0, PK)$.

It has been shown in [TY98] that the ElGamal encryption scheme provides IND-CPA security under the DDH assumption.

Theorem 1 (Tsiounis-Yung)**.** *The ElGamal encryption scheme is IND-CPA secure if and only if the DDH assumption holds in G.*

Proof. Suppose an adversary $\mathcal{A}$ with non-negligible advantage ε. Then, there is a DDH distinguisher $\mathcal{B}$. Algorithm $\mathcal{B}$ takes as input (g, g^a, g^b, T), where $a, b \in_R \mathbb{Z}_q$ and either $T = g^{ab}$ or $T \in_R G$.

- $\mathcal{B}$ defines PK such that $X = g^a$ and then runs $\mathcal{A}$ which chooses $M_0, M_1 \in G$.
- $\mathcal{B}$ picks $d \overset{R}{\leftarrow} \{0,1\}$ and computes $C = (C_1, C_2) = (g^b, M_d \cdot T)$.

  If $T = g^{ab}$, $C = (g^b, M_d \cdot g^{ab}) = (g^b, M_d \cdot X^b)$, so that C is a valid ElGamal encryption of M_d.

If $T \in_R G$, we can write $T = g^{ab+c}$ for some uniformly distributed $c \in_R \mathbb{Z}_q$, so that

$$C = (g^b, M_d \cdot T) = (g^b, M_d \cdot g^c \cdot X^b) = (g^b, M_{rand} \cdot X^b),$$

where $M_{rand} = M_d \cdot g^c$ is uniformly random in G. In particular, in $\mathcal{A}$'s view, M_d is perfectly hidden by g^c and C contains no information on $d \in \{0, 1\}$.

- $\mathcal{B}$ runs $\mathcal{A}$ which outputs $d' \in \{0, 1\}$:

  If $d' = d$, $\mathcal{B}$ outputs 1 (meaning $T = g^{ab}$)

  Otherwise, $\mathcal{B}$ outputs 0 (meaning $T \in_R G$)

Now, by the definition of $\mathcal{A}$'s advantage, we know that

$$\Pr[\mathcal{B} = 1 | T = g^{ab}] = \Pr[d' = d | T = g^{ab}] = \frac{1}{2} + \varepsilon$$

since C is a valid ElGamal encryption of M_d when $T = g^{ab}$. On the other hand, $\Pr[\mathcal{B} = 1 | T \in_R G] = \Pr[d' = d | T \in_R G] = \frac{1}{2}$ since, if $T \in_R G$, the ciphertext C contains no information on $d \in \{0, 1\}$ and $\mathcal{A}$ can only guess $d' = d$ with probability 1/2. So, $\mathcal{B}$ has a non-negligible advantage

$$\mathbf{Adv}_{\mathcal{B}}(\lambda) = |\Pr[\mathcal{B} = 1 | T = g^{ab}] - \Pr[\mathcal{B} = 1 | T \in_R G]| = \varepsilon$$

as a DDH distinguisher. □ □

Note that the proof does not hold anymore if we do not encode the message M as an element of G. For example, let $p = 2q + 1$, where p and q are both prime. Since $\mathbb{Z}_p^*$ is a multiplicative group of order $p-1$, we can choose G as the cyclic group $G_q \subset \mathbb{Z}_p^*$ of order q generated by $g = x^2 \bmod p$, where $x \in_R \mathbb{Z}_p^*$ is chosen at random. Hence, the ElGamal encryption scheme can be instantiated in the group $G_q = \langle g \rangle \subset \mathbb{Z}_p^*$ and computing the ciphertext as

$$(C_1, C_2) = \big(g^r \bmod p,\ M \cdot X^r \bmod p\big),$$

with $r \overset{R}{\leftarrow} \mathbb{Z}_q$. Then, suppose that we allow the message M to be in $\mathbb{Z}_p^* \backslash G_q$. An adversary could trivially win by simply choosing $M_0 = p - 1 = -1 \in \mathbb{Z}_p^* \backslash G_q$ and $M_1 \in G_q$ and, upon receiving the ciphertext (C_1, C_2), testing whether $C_2^q \bmod p = 1$. Note that $C_2^q \bmod p = 1$ if M_1 was encrypted. Otherwise, $C_2^q \equiv -1 \pmod{p}$.

1.2 Pairing-Based Cryptography

Definition 5. *A pairing is a bilinear map $e : G_1 \times G_2 \to G_T$ for cyclic abelian groups G_1, G_2, G_T of order p (usually prime) such that:*

- $e(g^a, h^b) = e(g^b, h^a) = e(g, h)^{ab},\ \forall g \in G_1,\ \forall h \in G_2,\ \forall (a, b) \in \mathbb{Z}^2$*;*
- *If p is prime, then $e(g, h) = 1_{G_T} \iff g = 1_{G_1}$ or $g_2 = 1_{G_2}$.*

Concretely, when we will use a pairing, we will assume that it is efficiently computable.

Pairings are usually instantiated using elliptic curves: G_1 and G_2 are elliptic curves subgroups and G_T is a finite field subgroup.

In some cases $G_1 = G_2$, so that e is called symmetric. In these symmetric configurations, the DDH problem becomes easy in $G = G_1 = G_2$: given (g, g^a, g^b, g^c), we have the equivalence $c = ab \iff e(g^a, g^b) = e(g, g^c) = e(g, g)^{ab}$. Since e is efficiently computable, it is trivial to decide if $c = ab \bmod p$.

Also, the Discrete Logarithm problem is not any harder in the groups (G_1, G_2) (regardless of whether $G_1 = G_2$ or not) than in G_T: for example, if $g, g_1 \in G_1$, then $\log_g(g_1) = \log_{e(g,h)} e(g_1, h)$ for any $h \in G_2$, so that a Discrete Logarithm instance in G_1 is easily turned into a Discrete Logarithm instance in G_T. The same holds for a DL instance in G_2.

1.2.1 One-Round Tripartite Diffie-Hellman

A tripartite key-agreement protocol based on pairings has been described in [Jou00]. The goal is to have three parties A, B and C agree on a shared key K, which is only computable to the three of them and infeasible to predict for the rest of the world. Moreover, they want to do it by having each player A, B or C send only one message to the two other players.

First, we have to define a computational assumption related to pairings. For simplicity, we consider symmetric pairings $e : G \times G \to G_T$ (namely, configurations with $G = G_1 = G_2$), but the following definition can be generalized to the case $G_1 \neq G_2$.

Definition 6 (DBDH assumption). *The Decision Bilinear Diffie-Hellman (DBDH) assumption holds in (G, G_T) if the distributions*

$$D_0 = \{(g, g^a, g^b, g^c, e(g, g)^{abc} \mid a, b, c \xleftarrow{R} \mathbb{Z}_p\}$$

and

$$D_1 = \{(g, g^a, g^b, g^c, e(g, g)^d \mid a, b, c, d \xleftarrow{R} \mathbb{Z}_p\}$$

are computationally indistinguishable. The advantage of a distinguisher can be defined as the distance between two probabilities, analogously to the DDH case.

We can now describe the tripartite Diffie-Hellman key agreement protocol. The protocol only takes one round as each player only sends one message to the two other players

Let $e : G \times G \to G_T$ be a symmetric bilinear map, with a generator $g \in G$. The three parties proceed as follows.

1. A chooses $a \xleftarrow{R} \mathbb{Z}_p$ and sends g^a to B and C;
2. B chooses $b \xleftarrow{R} \mathbb{Z}_p$ and sends g^b to A and C;
3. C chooses $c \xleftarrow{R} \mathbb{Z}_p$ and sends g^c to A and B.

When the protocol ends, A, B and C are all able to compute the shared key

$$K = e(g,g)^{abc} = e(g^b, g^c)^a = e(g^a, g^c)^b = e(g^a, g^b)^c,$$

which is pseudo-random to everyone but A B or C. Namely, under the DBDH assumption, $K = e(g,g)^{abc}$ is computationally indistinguishable from a random element of G_T given (g, g^a, g^b, g^c).

The main interest of this tripartite key-agreement is that it needs only one round. However, this protocol is secure only against passive adversaries.

1.2.2 Short signatures

The standard security notion for signature scheme is called existential unforgeability under chosen-message attacks (EUF-CMA) or, more simply, security under chosen-message attacks (CMA).

Definition 7 (CMA). *A signature scheme (KeyGen, Sign, Verify) is EUF-CMA-secure iff no PPT adversaries has non-negligible advantage in the following game:*

1. *The challenger generates $(PK, SK) \leftarrow \mathsf{KeyGen}(\lambda)$ and PK is given to A.*

   A set $Q =$ is initialized.

2. *$\mathcal{A}$ makes signing queries:*

   - *$\mathcal{A}$ chooses a message M and obtains a signature $\sigma \leftarrow \mathsf{Sign}(SK, M)$ from the challenger;*
   - *The challenger updates Q and sets $Q := Q \cup \{M\}$.*

   Note that signing queries are made adaptively in that each query may depend on the answer to previous queries.

3. *$\mathcal{A}$ outputs a pair (M^*, σ^*) and wins if and only if $\mathsf{Verify}(M^*, \sigma^*, PK) = 1$ and $M^* \notin Q$.*

Here, $\mathcal{A}$'s advantage is its probability of success, taken over all random choices.

We now describe a pairing-based signature scheme, due to Boneh, Lynn and Shacham [BLS01], which provides short signatures. For the security level of AES-128, the scheme yields signatures of 256 bits. It can be proved CMA-secure under the CDH assumption in the random oracle model.

The signature scheme of Boneh, Lynn and Shacham (BLS) is described as follows.

- $\mathsf{KeyGen}(\lambda)$: choose groups (G, G_T) of prime order $p > 2^\lambda$ with a bilinear map $e : G \times G \to G_T$, a generator $g \xleftarrow{R} G$, a hash function $H : \{0,1\}^* \to G$, which will be modeled as a random oracle in the security analysis. Then, pick a group element $X = g^x \in G$, for a randomly chosen $x \xleftarrow{R} \mathbb{Z}_p$ which will be the private key. Define

  $$PK = \{(G, G_T), g, X = g^x, H\}$$

  and $SK = \{x\}$.

- $\mathsf{Sign}(SK, M)$: given $M \in \{0,1\}^*$, compute and output the signature

$$\sigma = H(M)^x \in G.$$

- $\mathsf{Verify}(M, \sigma, PK)$: to verify a candidate signature $\sigma \in G$ on a message M, decide if $(g, X, H(M), \sigma)$ is a Diffie-Hellman tuple. Namely, decide if $\log_{H(M)}(\sigma) = \log_g X$. This can be done by exploiting the easiness of DDH in G and testing the equality

$$e(\sigma, g) = e(H(M), X).$$

  If the latter equality holds, return 1. Otherwise, return 0.

1.3 Identity-Based Encryption (IBE)

Shamir suggested the concept of IBE in [Sha84]. The idea is that any easy-to-remember string can be a public key. The motivation for IBE schemes is to simplify key management and remove the need of public key certificates as much as possible: since a key is the identity of its owner, there is no need to bind them by a digital certificate and a public repository containing a list of user names and their associated public keys becomes useless since public keys are human-memorable. 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 called private key generators (PKGs) that have to generate private keys associated to users identities thanks to their secret key. This does not completely remove the need of certificates but, since many users depend on the same authority, this need is drastically reduced.

1.3.1 Definition

Definition 8 (Identity-Based-Encryption). *An IBE scheme is a tuple of algorithms (Setup, KeyGen, Encrypt, Decrypt) such that:*

- *Setup*(λ)*: given a security parameter* $\lambda \in \mathbb{N}$*, outputs a pair* (MPK, MSK) *(run by an authority called Private-Key Generator (PKG));*

- *KeyGen*(MSK, ID)*: given* MSK *and user's identity* ID*, outputs* d_{ID} *(run by the PKG);*

- *Encrypt*(MPK, M, ID)*: given* MPK*, a plaintext* M *and the receiver's identity* ID*, outputs a ciphertext* C*;*

- *Decrypt*(MPK, d_{ID}, c)*: given* MPK*, a private key* d_{ID} *and a ciphertext* C*, outputs* M *or an error symbol* $\perp$ *indicating a decryption failure.*

1.3.2 The Boneh-Franklin IBE

This scheme has been described in [BF01] and is the first example of IBE scheme.

The Boneh-Franklin IBE scheme can be described as follows:

- Setup(λ):
  1. Choose groups (G, G_T) of prime order $p > 2^\lambda$ with a bilinear map $e : G \times G \to G_T$ and a generator $g \overset{R}{\leftarrow} G$,
  2. Choose $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute $g_1 = g^\alpha \in G$,
  3. Choose a hash function $H : \{0,1\}^* \to G$ that will be modeled as a random oracle in the security analysis.

  Define
  $$MPK = \{(G, G_T), g, g_1 = g^\alpha, H\}$$
  and $MSK = \alpha$.

- KeyGen(MSK, ID): to generate a private key for the identity ID, compute and output
  $$d_{ID} = H(ID)^\alpha \in G.$$

- Encrypt(MPK, M, ID): given a message $M \in G_T$, the master public key MPK and the receiver's identity $ID \in \{0,1\}^*$,
  1. Pick $r \overset{R}{\leftarrow} \mathbb{Z}_p$,
  2. Compute
  $$C = (C_1, C_2) = \Big(g^r, M \cdot e\big(g_1, H(ID)\big)^r\Big).$$

- Decrypt(MPK, d_{ID}, C): parse C as (C_1, C_2) and compute
  $$M = C_2 / e(C_1, d_{ID}).$$

The scheme is correct since

$$e(d_{ID}, g) = e(H(ID)^\alpha, g) = e(H(ID), g^\alpha) = e(H(ID), g_1) \tag{1.1}$$

(Note that each private key is a BLS signature delivered by the PKG on the identity ID). Hence, if we raise both members of (1.1) to the power $r \in \mathbb{Z}_p$, where r is the encryption exponent, we obtain the equality

$$e(d_{ID}, C_1) = e(H(ID), g_1)^r,$$

which explains why the decryption algorithm successfully recovers the message M.

We will see that, under the DBDH assumption, the Boneh-Franklin IBE satisfies the notion of semantic security for IBE schemes. In the context of identity-based encryption, the usual notion of semantic security must be strengthened by allowing the adversary to corrupt some identities before trying to threaten other identities. Allowing the adversary to corrupt a polynomial number of identites further gurantees that the scheme will be collusion-resistant: namely, a coalition made of an arbitrary (but polynomial) number of dishonest users of identities $ID_1, \ldots, ID_q$ will not be able to reconstruct the authority's master secret key MSK by pooling their private keys $d_{ID_1}, \ldots, d_{ID_q}$. The required collusion-resistance property is one of the main difficulties that make the construction of IBE schemes non-trivial.

Definition 9 ([BF01])**.** *An IBE scheme is semantically secure (or IND-ID-CPA secure) if no PPT adversary has non-negligible advantage in the following game:*

1. *The challenger generates $(MPK, MSK) \leftarrow \mathsf{Setup}(\lambda)$, gives MPK to A and initializes a set $Q := \emptyset$.*

2. *$\mathcal{A}$ corrupts identities of its choice and repeats the following kinds of queries:*
   - *$\mathcal{A}$ chooses an identity ID and obtains $d_{ID} \leftarrow \mathsf{KeyGen}(MSK, ID)$ from the challenger.*
   - *The challenger updates $Q := Q \cup \{ID\}$.*

   Note that each query may depend on the answer to previous queries.

3. *$\mathcal{A}$ chooses M_0, M_1 and a target identity $ID^* \notin Q$.*

4. *Challenger picks $d \overset{R}{\leftarrow} \{0, 1\}$ and returns $C^* = \mathsf{Encrypt}(MPK, M_d, ID^*)$.*

5. *$\mathcal{A}$ corrupts more identities under the restriction that ID^* can never be corrupted. Hence, it must hold that $ID^* \notin Q$ at then end of the game.*

6. *$\mathcal{A}$ outputs $d' \in \{0, 1\}$ and wins if and only if $d' = d$.*

The adversary's advantage is defined as the distance $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |Pr[d' = d] - \frac{1}{2}|$.

Bibliography

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, pages 213–229, 2001.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, pages 514–532, 2001.

[Jou00] Antoine Joux. A one round protocol for tripartite diffie-hellman. In *Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Leiden, The Netherlands, July 2-7, 2000, Proceedings*, pages 385–394, 2000.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, pages 47–53, 1984.

[TY98] Yiannis Tsiounis and Moti Yung. On the security of elgamal based encryption. In *Proceedings of the First International Workshop on Practice and Theory in Public Key Cryptography: Public Key Cryptography*, PKC '98, pages 117–134, 1998.