NIST Internal Report
NIST IR 8450

Overview and Considerations of Access Control Based on Attribute Encryption

Vincent C. Hu

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8450

NATIONAL INSTITUTE OF
STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Internal Report
NIST IR 8450

Overview and Considerations of Access Control Based on Attribute Encryption

Vincent C. Hu
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8450

September 2023

U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

Certain commercial equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

NIST Technical Series Policies

Publication History

Approved by the NIST Editorial Review Board on 2023-08-29

How to Cite this NIST Technical Series Publication:

Hu VC (2023) Overview and Considerations of Access Control Based on Attribute Encryption. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) NIST IR 8450. https://doi.org/10.6028/NIST.IR.8450

Author ORCID iDs

Vincent C. Hu: 0000-0002-1648-0584

Contact Information

ir8450-comments@nist.gov

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

Encryption technology can be incorporated into access control mechanisms based on user identities, user attributes, or resource attributes. Traditional public-key encryption requires different data to have different keys that can be distributed to users who satisfy perspective access control policies along with the encrypted version of the data. However, some distributed or pervasive system environments wish to avoid the public-key encryption's all-or-nothing data access limitation when considering their performance requirements. Attribute-based encryption incorporates access control policies and attributes with encryption and decryption functions and a one-to-many authorization scheme that requires fewer keys than public-key encryption. It also utilizes collusion-resistance, which provides a more efficient and flexible attribute-based access control mechanism that supports high-performance systems (e.g., cloud, IoT, disrupt-tolerant networks, wireless sensor networks, mobile ad-hoc networks, and public search service systems).

Keywords

access control; attribute-based access control; attribute-based encryption; authorization; encryptions; identity-based encryption; public-key encryption.

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof-of-concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Patent Disclosure Notice

NOTICE: ITL has requested that holders of patent claims whose use may be required for compliance with the guidance or requirements of this publication disclose such patent claims to ITL. However, holders of patents are not obligated to respond to ITL calls for patents and ITL has not undertaken a patent search in order to identify which, if any, patents may apply to this publication.

As of the date of publication and following call(s) for the identification of patent claims whose use may be required for compliance with the guidance or requirements of this publication, no such patent claims have been identified to ITL.

No representation is made or implied by ITL that licenses are not required to avoid patent infringement in the use of this publication.

Table of Contents

List of Figures

List of Tables

Acknowledgments

The author would like to express his thanks to Lily Chen, Isabel Van Wyk, and Jim Foti of NIST for their reviews of both the public comment version as well as the final publication.

Executive Summary

Traditional public-key encryption (PKE) requires different data to have different keys that can be distributed to users who satisfy access control policies along with the encrypted version of the data. With user-specific keys, communication complexity is linear to the number of users, and pre-distributed keys are neither bound to the attributes of users and data nor to the respective access control policy. If access policies or attributes change dynamically (especially in real time), keys need to change as well, which could cause inefficient performance in the system. Combining cryptography with access control mechanisms can avoid the PKE's all-or-nothing limitation of keys and improve performance. Encryption technology that is typically used for key exchange, digital signature, and certification can be incorporated into access control mechanisms based on user identities, user attributes, and resource attributes.

Attribute-based encryption (ABE) incorporates access control policies and attributes into encryption and decryption functions for public-key cryptography protocols through broadcasting. Fewer keys are used for ABE than for traditional PKE, which allows it to be an efficient and flexible attribute-based access control method.

The main features of ABE access control include:

- A one-to-many authorization scheme
- Fine-grained access control based on user (i.e., subject) or resource (i.e., object) attributes
- Message sending without obtaining public-key certificates from public-key infrastructure
- Data decryption without evaluating permissions from access control policy
- Collusion-resistance so that a user who holds multiple keys cannot combine different keys to access a resource that is only allowed by one key

The fine-grained and collusion-resistant features of ABE support the physical resources and performance demands of systems like the cloud, Internet of Things (IoT), disrupt-tolerant networks, wireless sensor networks, mobile ad hoc networks, and public search service systems.

1. Introduction

Traditional public-key encryption (PKE) requires different data to have different keys that – along with the encrypted version of the data – can be distributed to users who satisfy access control policies. With user-specific keys, the communication complexity is linear to the number of users, and pre-distributed keys are neither bound to the attributes of users and data nor to the respective access control policy. Therefore, if access policies or attributes change dynamically (especially in real time), then keys need to change as well, which could cause the system's performance to become inefficient [GOLIC]. Combining cryptography with access control mechanisms can help avoid the PKE's all-or-nothing limitation of keys and lead to more efficient performance. To that end, encryption technology that is typically used for key exchange, digital signatures, and certification can be incorporated into access control mechanisms that are based on user identities, user attributes, and resource attributes.

Attribute-based encryption (ABE) [GPSW] incorporates access control policies and attributes into encryption and decryption functions for public-key cryptography protocols through broadcasting. ABE encrypts only once by using a public key according to attributes associated with the access control policy. Only users hold the correct private decryption keys, which satisfies the access policies for decrypting data. ABE's fine-grained access control mechanism is based on user (i.e., subject) attributes or data (i.e., resource) attributes. Thus, the size of ABE encrypted data and the resulting communication complexity for key distribution are linear in the number of attributes, not users. Broadcasting enables ABE to utilize fewer keys than traditional PKE schemes, which allows it to be an efficient and flexible attribute-based access control method.

The main features of ABE access control include:

- A one-to-many authorization scheme.
- Fine-grained access control based on user (i.e., subject) attributes or resource (i.e., object) attributes.
- Message sending without obtaining public-key certificates from public-key infrastructure.
- Data decryption without evaluating permissions from access control policy.
- Collusion-resistance so that a user who holds multiple keys cannot combine different keys to access data that are only allowed to be accessed by one key.

These fine-grained and collusion-resistant features support the physical resources and performance demands of systems like the cloud, the Internet of Things (IoT), disrupt-tolerant networks, wireless sensor networks, mobile ad hoc networks, and public search service systems [ELT][SW].

This document is organized as follows:

- Section 1 is the introduction.
- Section 2 provides an overview of the fundamental theories that ABE is built on, including elliptic-curve cryptography, bilinear pairing, and bilinear pairing for elliptic curve cryptography.
- Section 3 introduces identity-based encryption (IBE).

- Section 4 illustrates ABE algorithms for Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and Key-Policy Attribute-Based Encryption (KP-ABE).
- Section 5 describes considerations for the applications of ABE from the perspectives of security, performance, access control policies, and support models.
- Section 6 is the conclusion.

2. Fundamental Theories

The underlying function of ABE is primarily based on public-private key cryptography calculated in bilinear pairings on elliptic curve groups. This section outlines the fundamental theories of elliptic curve, elliptic-curve cryptography, bilinear group, bilinear pairing, and elliptic-curve cryptography for ABE.

2.1. Elliptic Curve

An elliptic curve is so named for being described by cubic equations (used for calculating the circumference of an ellipse), which is of the form $y^2 = x^3 +ax +b$ ($y^2 + axy + by = x^3 +cx^2 + dx + e$), where all of the coefficients are real numbers that satisfy some conditions [ROBI][SP800-186]. However, elliptic curve is not an ellipse but, rather a cubic (x^3) formed by quadratic curves. The basic specifications for elliptic curves are:

1. Single elliptic curve points at infinity – or zero point – is denoted by “0”, which does not satisfy an elliptic curve equation but is needed for addition as the additive identity, 0 = -0. For any point P on an elliptic curve, $P + 0 = P$. All vertical lines intersect the curve at infinity (0), and if three points on an elliptic curve lie on a straight line, their sum is 0.
2. The negative of a point P is the point with the same x coordinate but the negative of the y coordinate of the elliptic curve’s x-y coordinate. That is, if $P = (x, y)$, then $-P = (x, -y)$, and these two points can be joined by a vertical line such that $P + (-P) = P - P = 0$, a point that adds a negative of itself will become an infinity point (as shown in **Fig. 1**). Any non-vertical line will intersect the curve in three places at most [MATA].

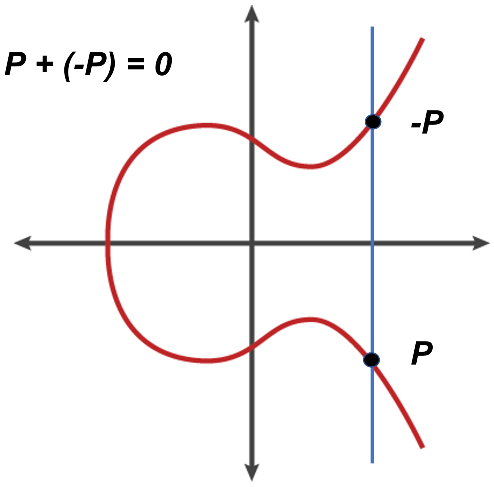

Fig. 1. $P + (-P) = P - P = 0$ in an elliptic curve

3. Add distinct points P and Q in an elliptic curve if $P \neq 0$ and $P \neq Q$ (as shown in **Fig. 2**), where $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$. If $R = P + Q = (x_R, y_R)$, then $x_R = s^2 – x_P - x_Q$ and $y_R = -y_p + s(x_P - x_R)$, where $s = (y_P - y_Q)/(x_P - x_Q)$.

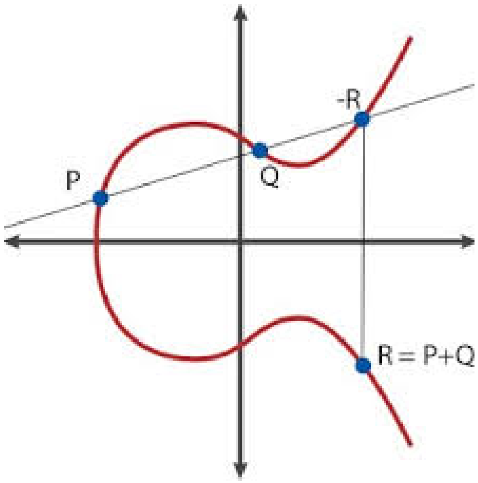

Fig. 2. *P* + *Q* in an elliptic curve

4. Doubling a point (also called a *dot* function) P ($P + P = 2P$) uses P's tangent line to find the second point in the curve, which will generate a new point -R and reflect -R from x axis to give a new point R such that from 3 above, if $y_P \neq 0$, $2P = R$ then replaces the Q with P and replaces s with $s = (3x_P^2 + a)/ (2y_P)$ for the elliptic curve: $y^2 = x^3 + ax + b$. Multiplying (also called dot, map, reflect) n (an integer) to a point P, $X = nP$ means that $P + P +\ldots+ P$ (n times), and the nP can be calculated by adding a doubling operation combined. For example, $5P$ = Double(Double P) + P (i.e., 2^2+1= 5). For an elliptic curve point P, two integers n and m, $m(nP) = n(mP)$, which is the same as the operation in a finite field $(g^y)^x = (g^x)^y$, where g is an element in a finite field and x, y are integers.

5. Order of a point P on the elliptic curve is defined to be the smallest integer n such that $nP = 0$.

6. Elliptic curve cryptography (ECC) uses elliptic curves over a finite field. F_z: {0 … z-1} is a set of points (x, y) that satisfy $y^2 = x^3 + ax + b$ mod z, where z is a prime number > 3, and $a, b, x, y \in F_z$. For example, an elliptic curve $y^2 = x^3 + 7$ mod 11, when $x = 1$, $y^2 = 8$ mod 11, but there is no integer y, satisfy $y^2 = 8$ mod 11. When $x = 2$, $y^2 = (8 + 7)$ mod 11 = 4 mod 11, $y = 2$, or y = 9 can satisfy the formula, so points (2, 2) and (2, 9) are in the elliptic curve. When $x = 3$: $y^2 = (27 + 7)$, mod 11 = 1 mod 11, $y = 1$, or 10. Continually, one can conclude that points (2, 2), (2, 9), (3,1), (3, 10), (4, 4), (4, 7), (5, 0), (6, 5), (6, 6), (7, 3), and (7, 9) are in the elliptic curve over the finite field defined by mod 11.

2.2. Elliptic-Curve Cryptography

Elliptic-curve cryptography (ECC) [SP800-56A][FIPS186-5] was invented by Neal Koblitz and Victor Miller in 1985 [MMSC] and standardized in Institute of Electrical and Electronics Engineers (IEEE) P1363a [IE]. The primary advantage of using elliptic-curve-based cryptography is that ECC uses a shorter key/parameter than Rivest–Shamir–Adleman (RSA)'s PKE to achieve the same security strength [MY]. This property addresses performance issues for systems that need to handle many encryption sessions at the same time, such as wireless communication devices, smart cards, web servers, and applications. These systems need security but lack the power, storage, or computational capability required for RSA's PKE cryptographic scheme. For example, Bitcoin and Ethereum use *secp256k1* elliptic curve to generate private- and public-key pairs [MOBI] for their blockchain implementations. Discrete logarithm problem (DLP) (i.e., given two points P and Q on an elliptic curve, find an integer n such that $Q = nP$) on an elliptic curve is a hard problem. However, ECC is more difficult to explain when compared to traditional RSA's PKE cryptographic scheme [ROBI]. As ECC gains popularity, more applications are using it, such as Internet Key Exchange (IKE), Transport Layer Security (TLS), Tor, iMessage, Bitcoin, and Ethereum [LXYS].

The international consortium Standards for Efficient Cryptography Group (SECG) [DANI] developed commercial standards for efficient and interoperable ECC. SECG published a document with a recommend set of parameters refereed by the tuple (p, a, b, G, n, h) called Elliptic Curve Domain Parameters to describe an elliptic curve used for ECC, where p is a prime number for defining the finite field such that $F_p = \{0 \ldots p\text{-}1\}$, and a and b (usually restricted by $4a^3 + 27 + b^2 \neq 0$) are the coefficients of the elliptic curve equation $y^2 = x^3 + ax + b$ [SP800-186]. G is the generator point. n is the order of the G generator (base) point (also called n torsion point), which determines the maximum value that can be turned into private key (ranging from 1 to n -1). h equals N/n called *cofactor* such that N is the order of the elliptic curve (the number of points in the elliptic curve). For example, the finite field F_{37} with $p = 37$ for the elliptic curve: $y^2 = x^3 - x + 3 \bmod 37$ (a = -1, b = 3) has order $N = 42$. For $n = 7 \in$ factors of N in $\{1, 2, 3, 6, 7, 14, 21, 42\}$, the point $P = (2, 3)$ is the base point G because $P \neq 0$, $2P \neq 0$, $3P \neq 0$, $6P \neq 0$, but $7P = 0$. According to Lagrange's theorem, the order of a subgroup (generated by G) is a factor of N. That is, $N = nh$. For any point P in the elliptic curve, $NP = 0$ (i.e., $n(hP) = 0$). Elliptic curves defined by parameter sets have been given IDs in the standards for easier identification. For example, s*ecp256k1* is EC $y^2 = x^3 + 7$ (used by Bitcoin or Ethereum) [SP800-186, MOBI].

For cryptographic usage, the elliptic curves are selected with a subgroup generated by the generator point G such that the order is a prime and large enough for targeted security strength. The steps are:

1. Select an elliptic curve pseudo randomly[1].
2. Calculate the order N of the elliptic curve[2].
3. If N has a prime factor n that is large enough to satisfy the required security strength, go to step 4. Otherwise, go back to step 1.
4. Compute the cofactor $h = N/n$.
5. Choose a random point P as a candidate generator G on the curve.
6. Compute $G = hP$.
7. If G is 0 (i.e., the subgroup has order 1), then go back to step 4. Otherwise, G is the generator (of a subgroup) with order n and cofactor h.

Note that this algorithm only works if n is a prime. If n were not a prime, then the order of G could be one of the divisors of n [CORB].

In ECC, a point $X = nG$ where n is an integer and G is the generator is used for the public key, and n is used as the private key. For example, the message from the sender to the receiver with

[1] ECC standards use recommended curves with already defined subgroups and generators in [SP800-186], Appendix C.3.1.

[2] Schoof's algorithm [SCHOOF] can be applied to find N, but it does not work for finding the order of a subgroup generated by a point.

the ciphertext $C_m = \{KG, M + KP_{receiver}\}$ can be decrypted by function $Decrypt\ (C_m)$: $M + KP_{receiver} - S_{receiver}(KG) = M + K(S_{receiver}G) – S_{receiver}(KG) = M$, where M is the message converted to an elliptic point, K is a random number, KG is a point in the elliptic curve that can be known by everyone sent through a non-encrypted channel, $P_{receiver}$ is the receiver's public key, $S_{receiver}$ is the receiver's private key such that $P_{receiver} = S_{receiver}G$, and "+" is elliptic curve points addition [ROBI]. ECC can also be applied to a digital signature, such as the Elliptic Curve Digital Signature Algorithm (ECDSA). Assume that the private key $Pr = d$ is an integer. The public key $Q = kG$ is an elliptic curve point. To sign a message m, compute $e = H(m)$, where H is a hash function, and assume that e is an integer such that $1< e < n$. Randomly select an integer k, $1 < k < n$ to compute $R= kG = (x_R, y_R)$, and then convert finite field element x_R to an integer r such that $1< r < n$. Compute $s = k^{-1}(e + r{\cdot}d) \bmod n$. The signature of m is (r, s). To verify the signature $Sig(m) = (r, s)$, a verifier computes $e = H(m)$. With the signature (r, s) and e, the verifier computes two values $u = e{\cdot}s^{-1} \bmod n$ and $v = r{\cdot}s^{-1} \bmod n$ with u and v and computes an elliptic point $R_1 = uG + vQ = (x_{R'}, y_{R'})$. Afterward, convert finite field element x_R to an integer r_1 such that $1< r_1 < n$. If $r = r_1$, then (r, s) is a valid signature. Otherwise, it is not a valid signature. This is shown in the following steps:

Parameters

G: A generator of the elliptic curve group over a finite field with order n, where n is a prime

d: Private key, an integer, $1< d <n$

Q: Public key, $Q = dG = G + G +\ldots+ G$ (d times)

Message to be signed

m: Message to be signed

Signing

1. Randomly select an integer k, $1 < k < n$, and compute $R= kG = (x_R, y_R)$.
2. Convert finite field element x_R to an integer r, such that $1< r < n$.
3. Compute $e = H(m)$. Assume that $e = H(m)$ is an integer $1< e < n$.
4. Compute $s = k^{-1}(e + r{\cdot}d) \bmod n$.
5. Output (r, s) as the signature of m.

Verifying

1. Compute $e = H(m)$.
2. Compute $u = e{\cdot}s^{-1} \bmod n$ and $v = r{\cdot}s^{-1} \bmod n$.
3. Compute $R_1 = uG + vQ = (\mathrm{x}_{R'}, \mathrm{y}_{R'})$.
4. Convert finite field element $x_{R'}$ to an integer r_1 such that $1< r_1 < n$.
5. If $r = r_1$, then (r, s) is a valid signature.

2.3. Bilinear Pair Mapping

Based on elliptic curve cryptography, bilinear pairing cryptography can be used for the new signature scheme [ST], identity-based encryption (IBE) [BF], and attribute-based encryption

(ABE) by applying bilinear pair mapping operations (i.e., bilinear pairing) on groups. For consistency of notation, this document will use G to denote a group, and elements in a group will be denoted by letters in lowercase. For instance, g will indicate a generator of G. In general, a group is defined by a set of elements and an operation on the group. Section 2.2 introduced a group that consisted of points on an elliptic curve with operation addition "+". A prime order subgroup with generator g is a cyclic group. That is, the group generated by g is $\{0, g, 2g, \ldots, (n-1)g\}$, where n is the order of G. It can define a mapping from integer group $\{0, 1, 2, \ldots, n-1\}$ to the cyclic group such that $f(x) = xg$ and $f(x + y) = xg + yg$. For an integer n, a group is called a cyclic group of order n if the group elements can be represented as $\{0, g, 2g, \ldots, (n-1)g\}$ and ng $=0$, where g is a generator G.

Let G_1 and G_2 be cyclic groups of the same order (e.g., G_1 and G_2 are cyclic additive groups generated by g whose order is a prime n). The bilinear pairing is a computable function e: $G_1 \times G_2 \rightarrow G_T$ that associates pairs of elements from G_1 and G_2 with elements in groups G_T, which is a group that contains the nth roots of unity [WF]. If (u, v) is a pair of elements such that $u \in G_1$, $v \in G_2$ are points of G_1 and G_2, respectively, then bilinear pairing function e takes u and v to produce a value in Group G_T. Bilinear pairing has the following properties when a, $b, c, d \in$ Z, and $u \in G_1$, $v \in G_2$, w is an element of G_1 or G_2 of cyclic additive groups:

- Computing $e(u, v)$ is efficient.
- $e(u + w, v) = e(u, v)e(w, v)$
- $e(u, w + v) = e(u, w)e(u, v)$
- $e(au, v) = e(u, av) = ae\ (u, v)$
- $e(au, bv) = e(abu, v)$
- $e(-u, v) = e(u, v)^{-1} = e(u, -v)$
- The mapping can also be $G_1 \times G_1 \rightarrow G_T$. In such cases, a pairing is called Symmetric: $e(u, v) = e(v, u)$ for all u, v.
- $e(au, bv) = \mathrm{e}(av, bu) = e(bu, av)$ when $G_1 = G_2$, and the mapping is symmetric.

If G_1 and G_2 are cyclic multiplication groups, then:

- Computing $e(u, v)$ is efficient.
- $e(u, v)^a = e(u^a, v) = e(u, v^a)$
- $e(u^a, v^b)\ e(u^c, v^d) = e(u, v)^{ab+cd}$ [QIAU]
- $e(-u, v) = e(u, v)^{-1} = e(u, -v)$
- $e(uw, v) = e(u, v)e(w, v)$
- The mapping can also be $G_1 \times G_1 \rightarrow G_T$. In such cases, a pairing is called Symmetric: $e(u, v) = e(v, u)$ for all u, v.
- $e(u^a, v^b) = e(u^b, v^a) = e(u, v)^{ab}$ when $G_1 = G_2$, and the mapping is symmetric [BETH].

- If $e(u, u)^k = 1$, then k is either 0 or a multiple of the order of the group when $G_1 = G_2$, and the mapping is symmetric [HUBWIZ].

For both addition and multiplication groups above:

- Non-degenerate property $e(u, v) \neq$ identity for some u, v, which ensures that if non-identical elements are selected for e, then the result of the pairing function will not be the identity of the target group. For example, assume that 0 is the identity. Then $e(u, v) = 0$ for all points v if and only if $u = 0$, and $e(u, v) = 0$ for all points u if and only if $v = 0$. Note that a degenerate property maps everything to the identity 0. That is, $\exists\ u \neq 0$, $v \neq 0$, $e(u, v) = 0$.
- Skew-symmetric: $e(u, v) = -e(v, u)$ when $G_1 = G_2$

2.4. Bilinear Pairing for Cryptography

Pairing-based cryptography [MD] applies bilinear pairing, which establishes the relationship between cryptographic groups for solving Decisional Diffie-Hellman problems. Weil and Tate pairings [MEFF] were first used in an effort to break ECC. The idea was to reduce the discrete logarithm problem in elliptic curves to a discrete logarithm problem in finite fields (called a MOV reduction) [BETH]. Bilinear pairing for ECC is based on the properties that add, double, and multiply (double means adding the same element, and multiply with an integer k means adding the same element k times) elliptic curve points to form an abelian group such that the bilinear pairing e: $G_1 \times G_2 \to G_T$ is defined by G_1, G_2 are subgroups of points on elliptic curves over a prime field F_p, and G_T is a subgroup of the multiplicative group of a finite field that contains the nth (n is the order or the number of points in the elliptic curve) of unity in a prime field (usually 12 degrees of extension[3] of a prime field). These values are not points. G_1, G_2, and G_T are all isomorphic to one another since they have the same order and are cyclic [BUTE][MPPRRC][IRON]. The bilinear pairing functions have the same properties as described in Section 2.3.

For this example, it is assumed that G_1 and G_2 are elliptic curve groups, but the notations are different from the curves. It uses g_1 as a point. It should be clear that private keys are integers. Message M must be an element in G_T. Moreover, it is assumed that the operation in G_1 and G_2 is "addition" and in G_T "multiplication".

For public-key encryption, an EC key pair used for bilinear pairing is a public key (PK) = private key (SK)g_1, an elliptic curve point, which means that the public key is just the private key times a fixed generator point g_1 in G_1. For example:

1. *Alice* generates a key pair (SK_A, PK_A). *Bob* generates (SK_B, PK_B), and both public keys are made available to public.
2. *Alice* can encrypt a message M to *Bob* by computing $Me(PK_B, SK_A\, g_2)$, where g_2 is a generator point in G_2. Note that $Me(PK_B, SK_A g_2) = Me(SK_B g_1, SK_A g_2) = M\, e(SK_A g_1, SK_B g_2) = Me(PK_A, SK_B g_2)$.

[3] Numbers that consist of 12 different values between 0 and prime – 1. Researchers conclude that the equivalent security of the degree 12 extension of a 256-bit prime field is under 100 bits [IRON].

3. *Bob* can recover *M* by computing $Me(PK_A, SK_Bg_2)\ e(PK_A, -SK_Bg_2) = M\ e(PK_A, (SK_B\text{-}SK_B)\ g_2) = Me(PK_A, 0) = M$.

Note that *M* must be an element in G_T, the operation in G_1 and G_2 is assumed to be addition, and G_T is assumed to be multiplication.

Bilinear pairing also works for message signatures. For example, *Alex* signs her message and sends it to *Bob* such that *Alex* generates SKg = public key *PK*, signature $C = SKH(M)$, where *SK* is *Alex*'s secret key, *g* is the generator of elliptic curve that is publicly known, *M* is the message *Alex* signed, and *H* is a hash function for hashing message *M* to another point in the elliptic curve. Bob receives *C*, *PK*, $H(M)$ and then calculates to check whether the pair mapping *e* of *g* and *C* equal the pair mapping of *PK* and $H(M)$ for Alex's signature of *M*: $e(g, C) = e(PK, H(M)) = e(g, SKH(M)) = e(SKg, H(M)) = e(PK, H(M))$. If so, the signature is verified.

In addition to public-key encryption, bilinear pairing is useful for functional encryption, which is a generalization of public-key encryption in which possessing a secret key allows one to learn a function of what the ciphertext is encrypting. It provides a mechanism for accessing the function of the data without revealing actual data values. For example, if *Alice* wants to prove to *Bob* that she knew the answer of $x + y$ without revealing the value of *x* and *y*, she can send xg_2 and yg_2 to Bob, who then calculates $A = e(g_1, xg_2)e(g_1, yg_2)$, where g_1 and g_2 are generator points of elliptic curve groups G_1 and G_2. Since *Bob* knows the value of $x + y$, he can check whether $e(g_1, g_2)^{x+y}$ is equal to *A* to prove that *Alex* indeed knows the value of *x* and *y* [SHINDE][BSW2011][BSW2012][BUCH].

Note that general ECC and bilinear pairing use different curves based on different security assumptions and have different trust models, as listed in **Table 1**.

Table 1. Elliptic curve used for general ECC and bilinear pairing.

	General ECC	**Pairing (IBE or ABE)**
Elliptic curve	Often use pre-defined Montgomery Curves or Edward curves. They do not have a small embedding degree. ECC cannot use supersingular curves.	Curves with embedding degree *k*, where *k* is small to make it pairing friendly. It can use supersingular curves.
Security assumptions	Discrete logarithm or Computational/Decisional Diffie-Hellman	Bilinear Diffie-Hellman (BDH) Problem
Trust models	PKI uses certificate authority (CA) as a trusted party, but CA does not access the private key.	Parameters need to be certified by a trusted third party (e.g., PKI). The private key for each party is generated by a key generator that accesses everyone's private key.

3. Identity-Based Encryption

Identity-based encryption (IBE) is a functional encryption proposed by Adi Shamir in 1984 [ADI] that requires a trusted key generator to publish a main[4] public key and retains the corresponding main private key (i.e., main key). The key generator allows any IBE user to generate a public key by combining the main public key with the user's identity value in text, such as an email address, name, or home address. The key generator also uses the main private key to generate the corresponding private key from the user's identity value. Thus, users may encrypt messages sent to other users without the prior distribution of a public key to other users. To decrypt or sign messages, the authorized user needs to obtain the appropriate private key from the key generator.

The Boneh-Franklin IBE encryption scheme [BF] applies the Weil pairing on elliptic curve over finite fields for setting up key management for public-key and private-key pairs from user identities for encrypting and decrypting messages, as constructed in the following.

The bilinear pairing function e: $G \times G \rightarrow G_T$, where g is the generator of G, and p is the order of G and G_T. The parameters are:

Identity $I \in \{0, 1\}^*$ for message sender

Message $M \in \{0, 1\}^m$

Hash function H: $\{0, 1\}^* \rightarrow G$

Extract function Q: $G_T \rightarrow \{0, 1\}^m$

Functions include:

Set up ( ) (by trusted key management server):

Return ($msk = Random\ (Z_p)$; $mpk = g^{msk}$). *msk* is the main secret key, which is for each public key of each access control system. *Random*( ) generates a random number.

Key generation(*mpk*, *msk*, *I*) (by trusted key management server for message receiver):

Return $sk = H(I)^{msk}$; *sk* is a private key for each identity.

Encryption(*mpk*, *I*, *M*) (for message sender):

$r = Random(Z_p)$; $R = g^r$; $K = e(mpk, H(I)^r)$; $W = Q(K) \oplus M$; $Return(R, W)$

Decryption(*mpk*, *sk*, *R*, *W*) (for message receiver):

$L = e(R, sk)$; Return $M = Q(L) \oplus W$; because $M = Q(L) \oplus Q(K) \oplus M$, and $L = K$ from the following:

$L = e(R, sk) = e(g^r, H(I)^{msk}) = e(g^{msk}, H(I)^r) = e(mpk, H(I)^r) = K$, where $H(\ )$ is a hash function. (Assume $H(I) = g^x$ for some x) [MIHIR].

[4] In an effort to address biased language, NIST has deprecated the use of "master" and replaced it with "main".

Further, IBE offers the capability to encode additional information into identities. For example, a sender can specify the expiration date of a message by appending a timestamp to the recipient's identity (e.g., through some formal protocol, like X.509). The receiver asks to retrieve the private key from the key manager (usually the key generator), who can evaluate the identity and decline the request if the expiration date has passed. Generally, embedding information in the identity provides an extra channel between the sender and the key manager with authenticity guaranteed in addition to the private key. The benefits of applying IBE can be demonstrated by an IBE email system:

- Senders can send mail to recipients who have not yet set up a public key.
- When sending email, there is no need for an online lookup to obtain the recipient's certificate.
- Senders can send email that can only be read at some specified time in the future.
- The system can proactively refresh the recipient's private key for a short time period [BONEH].

Note that the key generator can access the encrypted data for any receiver, and communications between the key generator and the receiver must be protected.

4. Attribute-Based Encryption

Attribute-based encryption (ABE) stems from IBE and is an encryption scheme that combines the principles of attribute-based access control [SP800-162] with the mechanisms of public-key cryptography. ABE allows data owners and data consumers to encrypt and decrypt data based on their attributes (e.g., organization, location, position) from which public and private keys are derived through a third-party key manager. ABE eliminates the need for public-key distribution and certification, and the authenticity of the public keys is implicitly guaranteed as long as the transport of the private keys to the corresponding user is secure. ABE is especially useful for the system environment that requires pre-distribution of authenticated keys due to technical limitations.

ABE has the following basic properties:

- Encryption time and ciphertext size are linear to the number of attributes involved.
- Collusion resistance means that it is impossible to decrypt any ciphertext for any new attribute set (CP-ABE) or new access policy (KP-ABE) by giving any number of randomized private keys.
- Randomized encryption prevents users from distinguishing repeated encryptions of the same message for privacy [GOLIC].

Popular distributed systems, such as cloud and IoT, make it possible for users to access dynamic resources in flexible environments. However, their growth and the ubiquity of mobile devices for data access have generated new security and performance challenges. Many studies have been conducted on ABE, such as applying it to distributed systems [HL] for its one-to-many cryptographic scheme as well as the capability to store, transmit, and retrieve high-dimensional data with low computational time and high security. This shows that ABE can address security and privacy issues in outsourced and pervasive data access environments [ZDXSLZ]. For example, for the large attribute universe of a cloud system, ABE allows data owners to compose access control policies based on their applications so that they can provide delegation capabilities to data users [BS]. However, the implementation of ABE requires complex support infrastructures – including key generation services and data storing services – to manage access structures and coordinate between clusters of users.

ABE is classified into two main schemes: ciphertext-policy ABE (CP-ABE) [BSW2007] and key-policy ABE (KP-ABE) [GPSW]. Selective security[5] of CP-ABE is more suited to user attributes, while adaptive or full security of KP-ABE is more suited to data (i.e., resource) attributes [GOLIC], as described in the following sections.

4.1. Ciphertext-Policy Attribute-Based Encryption

CP-ABE [BSW2007] enables data owners to define their own access policies over the user attributes and enforce those policies on data to be distributed. It provides a certain level of flexibility and scalability by removing the need for data owners to manage every individual access request and maintains an access control policy instead. Encryption and decryption of CP-

[5] For the challenger to private keys in the selective security model, the adversary has to commit the target attributes and declare the challenge message (ciphertext) before public parameters are set up. The selective security model is weaker than the fully secure (adaptive) model, which has no restrictions. Both are given a public key, several secret keys, and one challenge ciphertext [WSOE].

ABE are based on the policy specified over the attributes so that a user can gain access to data if they have appropriate attributes. For example, the attribute set {*student*, *professor*, *TA*, *RA*, *registration*} contains attributes for student records. To encrypt *student records*, the school administrator specifies a policy rule for permitting access to *student records*: *professor* OR (*student* AND *TA*) OR *registration*. Thus, users who have the attribute sets {*professor*} or {*student*, *TA*} can decrypt *student records*, but users who have the attribute sets {*TA*} or {*student*, *RA*} cannot. CP-ABE is a useful scheme for addressing the risks associated with data security in a cloud system that needs key management and data storing services [MHH][BCSES] to handle costumers with complex attribute structures.

Figure 3 shows the basic process steps of a CP-ABE scheme:

1. A trust authority generates public key *PK* and main key *MK* according to the applied attribute set and sends them to the key management service.
2. To access the data, requester *x* sends their attribute set *Ax* to the key management service.
3. The key management service sends the public key *PK* to the data owner and generates secret key *SKx* for the data requester according to their attributes.
4. Using the public key, the data owner generates ciphertext *CT* for the data (*message*) based on the rules of their access control policy and then uploads the data to the data resource service.
5. The requester decrypts ciphertext *CT* from the resource service by using their secret key and attributes.

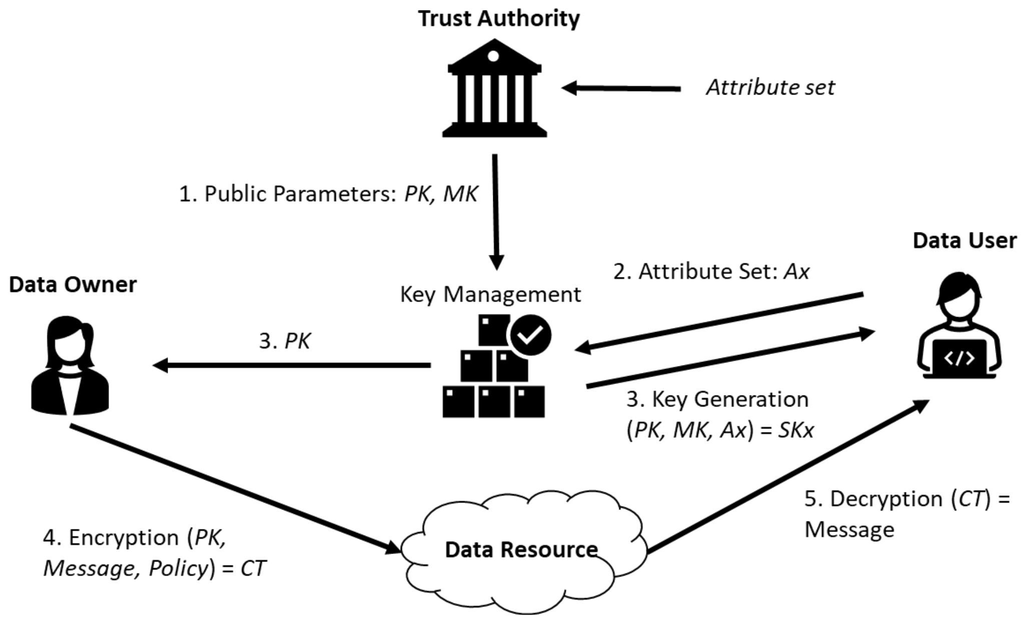

Fig. 3. Basic process steps of the CP-ABE scheme

The main secret key *MK* can decrypt all ciphertexts, which CP-ABE uses to derive user secret keys associated with different attributes. Formally, global attribute set $A = \{a_1 \ldots a_n\}$, where a_1,

… a_n are attribute elements. User *x* has the attribute set A_x, which elements may or may not be in *A*. Let *B* be the Boolean rule structure (i.e., access control policy). For example, $B = a_1$ AND a_2 OR (a_3 AND a_4) for the data of a data owner. Note that the fundamental CP-ABE can only be applied to the Boolean logic of a policy rule with a non-monochrome (i.e., including "NOT" gate) structure. Key generation function *Keygen* (*PK*, *MK*, A_x) = SK_x, where *PK* is the public key, and *MK* is the main key. Decryption function *Decry* (*CT*, SK_x) = *M*, where *CT* is the ciphertext, SK_x is the secret (private) key for the user *x*, and the message *M* is rendered if the function $B(SK_x)$ checks the SK_x against the policy *B* is satisfied. Otherwise, *M* is NULL.

Figure 4 shows an example structure of the access control rule and demonstrates the CP-ABE's algorithms for setup, encryption, key generation, and decryption functions.

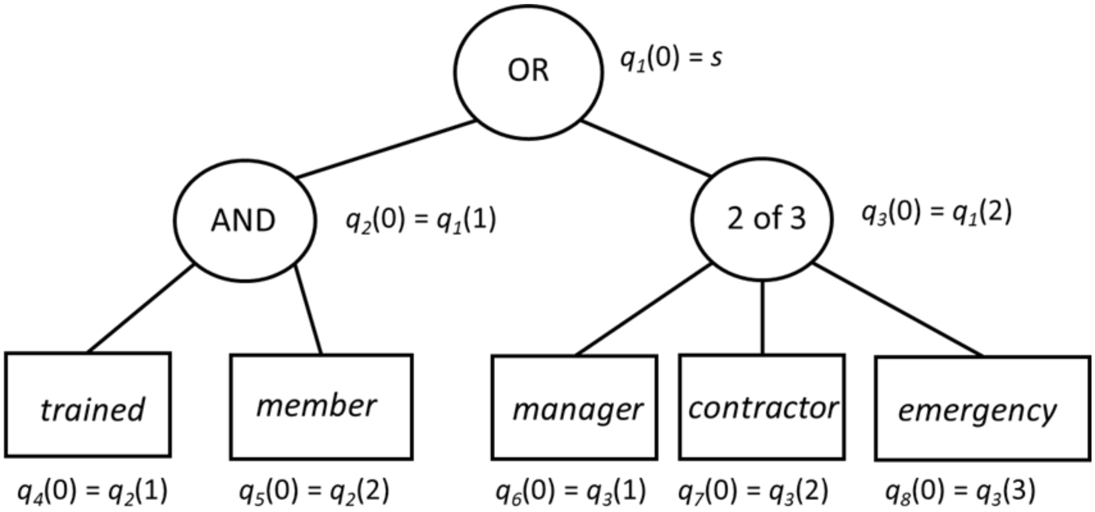

Fig. 4. Tree structure of an example access control policy

Setup function:

1. Main key *MK* = randomly chosen $\alpha, \beta \in Z_p$
2. Public key $PK = (G, g, g^{\beta}, e(g, g)^{\alpha}, g^{\frac{1}{\beta}})$, *G* is an elliptic curve group, *g* is the generator of the elliptic curve, $g^{\beta} = h$, and $g^{\frac{1}{\beta}} = f$ are for the delegation function (will not be discussed in this document).

Encryption function:

1. Let *T* be a tree that represents an access structure, as shown in **Fig. 4**. Each non-leaf node of the tree represents a threshold gate that is described by its children and a threshold value. If *n* is the number of children of a node *x* and k_x is its threshold value, then $0 < k_x \leq n$. The threshold value equals 1 for an OR gate (represented in a tree node of the Boolean operator on the node's children in the rule structure) and equals *n* for an AND gate with *n* elements or an *n*-out-of-*m* gate. Each leaf node *x* of the tree is described by an attribute and a threshold value $k_x = 1$.

2. Choose a polynomial q_i for each node q_1, q_2, …. q_8 for the tree structure that represents the access permission paths in the access control policy, as shown in **Fig. 4**. Set Polynomial degree $d_i(q_i)$ = Threshold value $k_i(q_i) – 1$ for each node q_i.

3. Choose random s such that root note $q_R(0) = s \in Z_p$, where p in Z is the order of the group G. For each node q_i, set $q_i(0) = q_j(n)$, where q_j is the parent node of q_i, and n is the sibling order from left to right. As shown in **Fig. 4**, $q_1(0) = s$, $q_2(0) = q_1(1)$, $q_3(0) = q_1(2)$, $q_4(0) = q_2(1)$, $q_5(0) = q_2(2)$, $q_6(0) = q_3(1)$, $q_7(0) = q_3(2)$, and $q_8(0) = q_3(3)$. According to 1 and 2 above, q_1 has degree 0, q_2 and q_3 has degree 1, and q_4, q_5, q_6, q_7, and q_8 have degree 0.

4. Encryption $(M, T, PK) = CT = \{T, Me(g, g)^{\alpha s}, C = h^s$, and for each leaf q_x:

   $C_x = g^{q_x(0)}$, $C_x' = H(l)g^{q_x(0)}$, where x is the sibling order, and l is a string of one of a leaf in $T\}$. For example:

   $C_4 = g^{q_4(0)}$, $C_4' = H(\text{"}trained\text{"})g^{q_4(0)}$

   $C_5 = g^{q_5(0)}$, $C_5' = H(\text{"}member\text{"})g^{q_5(0)}$

   $C_6 = g^{q_6(0)}$, $C_6' = H(\text{"}manager\text{"})g^{q_6(0)}$

   $C_7 = g^{q_7(0)}$, $C_7' = H(\text{"}contractor\text{"})g^{q_7(0)}$

   $C_8 = g^{q_8(0)}$, $C_8' = H(\text{"}emergency\text{"})g^{q_8(0)}$

   Where M is the message (data), T is the access control policy tree of attributes, as shown in **Fig. 4**. $e(\ ,\ )$ is a bilinear mapping function, and $H(\ )$ is a hash function mapping to a point in G.

Key generation function:

Choose $\gamma \in Z_p$, and for each attribute of a user (e.g., a user has attributes: A = {"*trained*", "*manager*", "*contractor*"}), choose $\gamma_{trained}$, $\gamma_{manager}$, $\gamma_{contractor} \in Z_p$.

Key generations $(A\ , MK) = SK = \{D = g^{\frac{(\alpha+\gamma)}{\beta}}\ ,\ D_l = g^{\gamma}H(l)^{\gamma_l}, D'_l = g^{\gamma_l}\ \}$ for all attributes the user has. For example:

$D_{trained} = g^{\gamma}H(\text{"}trained\text{"})^{\gamma_{trained}}$, $D'_{trained} = g^{\gamma_{trained}}$,

$D_{manager} = g^{\gamma}H(\text{"}manager\text{"})^{\gamma_{manager}}$, $D'_{manager} = g^{\gamma_{manager}}$,

$D_{contractor} = g^{\gamma}H(\text{"}contractor\text{"})^{\gamma_{contractor}}$, $D'_{contractor} = g^{\gamma_{contractor}}$,

where l is the string of one of a leaf in T.

(Note: MK contains α, β)

Decryption function:

Recursively go through the tree T to call *DecryptNode* (CT, SK, x). If the node x is a leaf node, then let $i = att(x)$ be a string of one of a leaf in T, and define as follows:

If $i \in S$ the set of all attributes in the tree, then

DecryptNode (CT, SK, x) =

$\frac{e(D_i,\ C_x)}{e(D'_i,\ C'_x)} = \frac{e(g^{\gamma}H(i)^{\gamma}i, g^{q_x(0)})}{e(g^{\gamma}i, H(i)^{q_x(0)})} = \frac{e(g^{q_x(0)}, g^{\gamma})e(H(i)^{\gamma}i, g^{q_x(0)})}{e(g^{q_x(0)}, H(i)^{\gamma}i)} = e(g,g)^{\gamma q_x(0)}$ [MUKH].

Note that all leaves are attributes.

For example:

$\frac{e(D_{manager}, C_6)}{e(D'_{manager}, C'_6)}$ = e(g, g)$^{\gamma q6(0)}$ (Note that $e(g,g)^{\gamma q_1(0)} = e(g,g)^{\gamma s}$)

For any leaf, return ⊥ (false) if it is not a user attribute.

If a node x is a non-leaf node, the algorithm proceeds such that for all nodes z that are children of x, it calls *DecryptNode*(*CT*, *SK*, *z*) and stores the output as F_z as the following.

Let S_x be an arbitrary k_x-sized set of child nodes z such that $F_z \neq \perp$. If no such set exists, then the node was not satisfied and returns ⊥. Otherwise, compute:

$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,s'_x}(0)}$, where i = index(z) is the order number of the child. That is, S'_x = $\{index\ (z), z \in S_x\}$.

$= \prod_{z \in S_x} (e\ (g,g)^{\gamma q_z(0)})^{\Delta_{i,s'_x}(0)}$

$= \prod_{z \in S_x} e(g,g)^{\gamma q_x(i) \Delta_{i,s'_x}(0)}$ (i.e., $\prod_{z \in S_x} (e\ (g,g)^{\gamma q_x(index(z))})^{\Delta_{i,s'_x}(0)}$ by construction)

$= e(g,g)^{\gamma q_x(0)}$ (using polynomial interpolation)

For example:

i = index(z) ∈ {1, 2, 3}, z ∈ {"*manager*", "*contractor*", "*emergency*"}

$F_{2of3} = F_{manager}{}^{\Delta_{1,(1,2,3)}(0)}\ F_{contractor}{}^{\Delta_{2,(1,2,3)}(0)} F_{emergency}{}^{\Delta_{3,(1,2,3)}(0)}$

$= (e(g,g)^{\gamma \cdot q_6(0)})^{\Delta_{1,(1,2,3)}(0)}\ (e(g,g)^{\gamma \cdot q_7(0)})^{\Delta_{2,(1,2,3)}(0)}\ (e(g,g)^{\gamma \cdot q_8(0)})^{\Delta_{3,(1,2,3)}(0)}$

$= e(g,g)^{\gamma(q_6(0)\Delta_{1,(1,2,3)}(0) + q_7(0)\Delta_{2,(1,2,3)}(0) + q_8(0)\Delta_{3,(1,2,3)}(0)}$

$= e(g,g)^{\gamma \cdot q_3(0)}$

Note that Lagrange coefficient $\Delta_{i,s}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$, $i \in Z_p$ and a set S of elements in Z_p. For example:

$\Delta_{1,(1,2,3)}(0) = \frac{0-2}{1-2}\frac{0-3}{1-3} = 3$, $\Delta_{2,(1,2,3)}(0) = \frac{0-1}{2-1}\frac{0-3}{2-3} = -3$, and $\Delta_{3,(1,2,3)}(0) = \frac{0-1}{3-1}\frac{0-2}{3-2} = 1$.

So,

$q_6(0)\Delta_{1,(1,2,3)}(0) + q_7(0)\Delta_{2,(1,2,3)}(0) + q_8(0)\Delta_{3,(1,2,3)}(0)$

$= 3q_6(0) - 3q_7(0) + q_8(0) = q_3(0)$ (i.e., $q_3(0) = 3q_3(1) - 3q_3(2) + q_3(3)$.

Since the algorithm started by simply calling the *DecryptNode* function on the root node R of the tree T, *DecriptNote* (*CT*, *SK*, *R*) = $e(g,g)^{\gamma \cdot q_R(0)} = e(g,g)^{\gamma s}$ if the tree is satisfied by S. Then calculate the following to retrieve the message M (note that $q_1 = q_R$, $g^{\beta s} = h^s = C$) [KB][BSW2007]:

$$\frac{Me(g,\ g)^{\alpha s}}{\frac{e(g^{\beta s}, g^{\frac{\alpha+\gamma}{\beta}})}{e(g,\ g)^{\gamma s}}} = \frac{Me(g,\ g)^{\alpha s}}{e(g,\ g)^{(\alpha+\gamma)s-\gamma s}} = M$$

An increasing number of organizations and individual users store their private data in open resources for sharing with others, such as cloud storage. Unlike traditional access control, data owners prefer to define their own access control policies rather than be controlled by a centralized access control policy. Thus, data owners encrypt their data on the open resource according to their defined access control policy so as not to compromise it. CP-ABE provides appropriate solutions to meet data owners' needs because it enables them to define access control policies and hide them by masking off attributes [HR].

4.2. Key-Policy Attribute-Based Encryption

Another variation of the ABE scheme is the KP-ABE [GPSW], wherein access control policies are associated with keys, and data are associated with attributes such that secret keys (private keys) are generated based on an access requester's attributes in the form of an access control policy. The ciphertext is labeled with a set of attributes so that decryption with a secret key works if and only if an attribute set built in ciphertext satisfies the structure of the access policy of the requester. Attribute sets can vary with each encryption.

Some access control models, such as multi-level and separation of duty security, are difficult to represent with straightforward Boolean formulas. In such cases, KP-ABE schemes can be defined to work with general Boolean circuits of attributes [TDN]. For example, to encrypt a secret document with attributes "*project_A*", "*project_B*", and " *project_C*" such that members involved in *project A*, *project B*, OR *project C* and *project A* OR *project D* can decrypt the document, but members involved in *project_A* AND *project_D* and *project_A* AND NOT *project_C* cannot decrypt it.

Figure 5 shows the basic process steps of KP-ABE functions:

1. The trust authority generates public parameters – the public key *PK* and main key *MK* – according to the applied attribute set and sends them to the key management service.
2. To access data, the requester *x* sends their attributes and access structure (policy) to the key management service.
3. The key management service generates secret key *SKx* using *PK* and *MK* for the requester according to their attributes and associated access structure and sends the public key *PK* to the data owner.
4. Data owners generate ciphertext for the data (i.e., message) based on the applied attribute set and public key *PK* and then upload the data to the data resource provider.
5. The requester retrieves and decrypts ciphertext from the data source provider using their secret key *SKx* and attributes associated with the access structure.

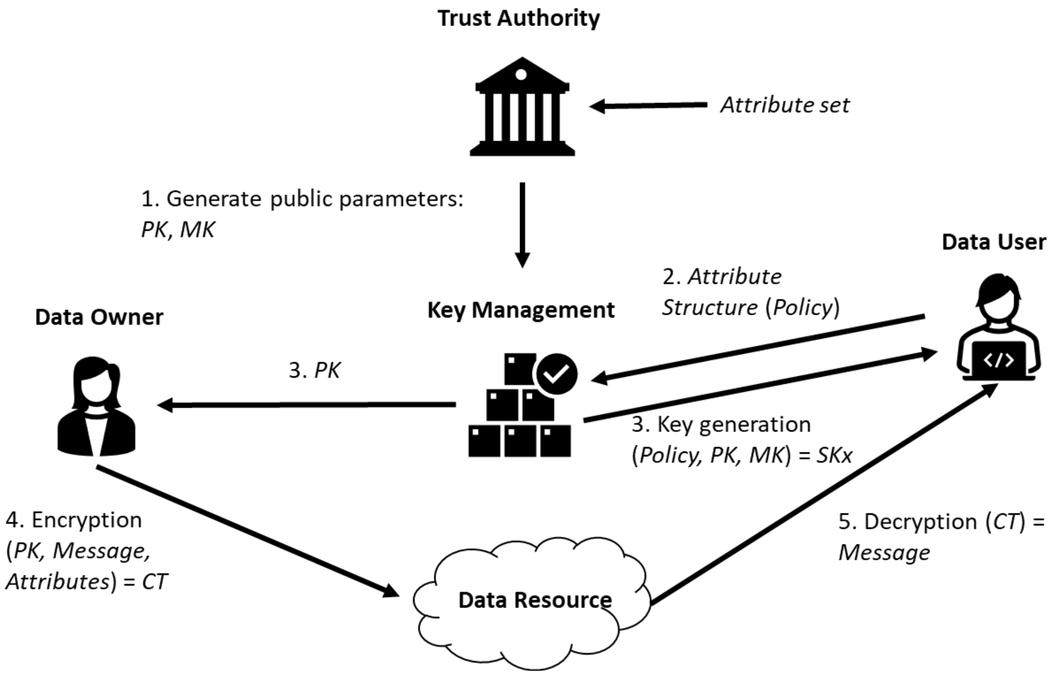

Fig. 5. Basic process steps of KP-ABE scheme

In general, the size of the public key of KP-ABE is linear to the total number of applied attribute sets. That is, the public key size is linear to the maximum number of attributes effectively used in encryption. However, it can be a fixed size in a random oracle large universe construction with hash function [GOLIC]. Using the example in **Fig. 4**, instead of a policy structure of data, it now represents an attribute structure of a data requester. The following demonstrates an example of KP-ABE's algorithms of setup, encryption, key generation, and decryption functions.

Setup function:

- Bilinear map function e: G_1 x $G_1 \rightarrow G_2$, G_1 has prime order p, and g is a generator of G_1.
- $U = \{a_1, a_2, \ldots, a_n\}$ is a set of applicable n attributes. For this example, a_1 = "*trained*", a_2 = "*member*", … from **Fig. 4**.
- t: $U \rightarrow Z_p$. Randomly choose $t_1, t_2, \ldots t_n \in Z_p$ from G_1, t_x for attribute x in U.
- Main key MK: Random $y \in Z_p$, $t_1, t_2, \ldots t_n$. and $Y = e(g, g)^y$
- Public key PK: Y, $T_1 = g^{t_1}$, $T_2 = g^{t_2}$, …., $T_n = g^{t_n}$

Encryption function:

$Encrypt(M, \gamma, PK) = C = (\gamma, MY^s, T_i^s \;\forall i \in \gamma)$, where random $s \in Z_p$, message $M \in G_2$, $\gamma \subseteq U$. For example, γ = {"*trained*", "*manager*", "*contractor*"} for a user.

Key generation function:

The algorithm is the same as CP- ABE, but it is applied to each data requester instead.

att(x): if x is a leaf node, then return the attribute associated with x

$num(x)$: the number of children of a node x

$K(x)$: threshold value, $0 <= K(x) <= num(x)$

$K(x) = 1$, for an OR gate

$K(x) = num(x)$, for an AND gate with n elements or an n-out-of-m gate.
$index(x)$: return node's index

- Choose a polynomial q_x for each node: $q_1, q_2, q_3, \ldots\ldots.. q_8$.
- $degree\ (q_x) = K(x)$ -1, $degree\ (q_1) = 0$, $degree\ (q_2) = 1$, $degree\ (q_3) = 1$, $degree\ (q_4) = 0$ ….. $degree\ (q_8) = 0$, as show in **Fig. 4**.
- Access Tree: set root note $q_1(0) = y$, and choose $degree\ (q_1)$ other points of the polynomial q_1 randomly to define it completely. For example, in **Fig. 4**, $q_1(0) = y \in Z_p$, $q_2(0) = q_1(1)$, $q_3(0) = q_1(2)$, $q_4(0) = q_2(1)$, $q_5(0) = q_2(2)$, $q_6(0) = q_3(1)$, $q_7(0) = q_3(2)$, and $q_8(0) = q_3(3)$.
- For each leaf node x, $i = att(x)$ generates:

  $D = \{D_x = g^{\frac{q_x(0)}{t_i}}$ for all attributes a user has$\}$. For example:

  $D = \{\ D_4 = g^{\frac{q_4(0)}{t_{trained}}},\ D_5 = g^{\frac{q_5(0)}{t_{member}}}, D_6 = g^{\frac{q_6(0)}{t_{manager}}}\ \}$

Decryption function:

Inputs:

$C = (\gamma, MY^s, T_i^s\ \ \forall i \in \gamma)$

Private Key : D

Access Tree: T

With inputs, define a recursive algorithm $DecryptNode\ (C, D, x)$ that takes a node x in the tree and outputs a group element of G_2 or $\perp$.

Let $i = att(x)$. If the node x is a leaf node,

$DecryptNode\ (C, D, x) = e(D_x, T_i^{\ s}\ ) = e(g^{\frac{q_x(0)}{t_i}}\ , g^{s \cdot t_i}\ ) = e(g, g)^{s \cdot q_x(0)}$.

For example, if $i \in \gamma$,

$e(D_6, T_{manager}{}^{s}\ ) = e(g^{\frac{q_6(0)}{t_{manager}}}\ , g^{s \cdot t_{manager}}\ ) = e\ (g, g)^{s \cdot q_6(0)}$

$e(D_7, T_{contractor}{}^{s}\ ) = e(g^{\frac{q_7(0)}{t_{contractor}}}\ , g^{s \cdot t_{contractor}}\ ) = e(g, g)^{s \cdot q_7(0)}$

If x is not an attribute in a leaf, then return $\perp$. If x is a non-leaf node, then proceed as follows:

For all nodes z that are children of x, call $DecryptNode(C, D, z)$, and store the output as F_z. Let S_x be an arbitrary k_x-sized set of child nodes z such that $F_z \neq \perp$. If no such set exists, then the node was not satisfied, and the function returns $\perp$.

Otherwise, compute:

$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,s'_x}(0)}$, where $i = index(z)$ is the index number of child node z, S'_x = (i*ndex*(z), Z $\in S_x$), and Lagrange coefficient $\Delta_{i,s}(x) = \prod_{j \in s, j \neq i} \frac{x-j}{i-j}$, $i \in Z_p$ and a set S of elements in Z_p.

= $\prod_{z \in S_x} (e(g,g)^{sq_z(0)})^{\Delta_{i,s'_x}(0)}$

= $\prod_{z \in S_x} (e(g,g)^{sq_{parent}(index(z))})^{\Delta_{i,s'_x}(0)}$ (by construction)

= $\prod_{z \in S_x} e(g,g)^{sq_x(i)\Delta_{i,s'_x}(0)}$

= $e(g,g)^{sq_x(0)}$ (using polynomial interpolation).

For example:

$(e(g,g)^{sq_6(0)})^{\Delta_{1,(1,2,3)}(0)}\ (e(g,g)^{sq_7(0)})^{\Delta_{2,(1,2,3)}(0)}\ \ e(g,g)^{sq_3(0)} = \mathrm{e}(g,g)^{sq_1(0)} = e(g,g)^{sy}$.

Hence, $q_6(0) = q_3(6)$, $q_7(0) = q_3(7)$, $q_3(0) = q_1(3)$.

If and only if the ciphertext satisfies the tree, then $DecryptNode(C, D, x) = e(g, g)^{sy}$. Since $MY^s = Me(g, g)^{ys}$, simply divide out $e(g, g)^{ys}$ to recover the message M [HALL][GPSW].

KP-ABE is also useful for searching encryption contents from categorized attributes. For example, searching a video from attribute set = {*a*, *b*, *c*, *d*, *e*}, where *a* is the title, *b* is the actors, *d* is the directors, and *e* is producer. Users can decrypt with search criteria – such as *a*, *b* OR *c*, *d* AND *e*, and *a* OR *e* – because they are all in the attribute set, but users cannot decrypt with search criteria *a* AND *f*, *d* AND (NOT *e*), *b* AND (NOT *c*), *f* because the attribute set cannot satisfy the search criteria defined by the attribute set [GOLIC].

5. ABE System Considerations

The ABE encryption scheme allows for higher data scalability, less computational time, low memory usage, and large-scale deployments of system platforms [KKB] compared to traditional PKE. However, for applications, it suffers from the drawbacks of low efficiency, less expressive access policies, and the use of random oracle models. Thus, the deployment and adoption of ABE have been slow. According to [ELT], ABE is absent from common data products and formats that are generated by widely-used commercial authoring products (e.g., Microsoft Word documents, Excel spreadsheets, PowerPoint slides) for lacking selective and fine-grained control over what is shared and with whom. In general, even with specific modifications or add-on applications (e.g., blockchain), implementation of ABE applications should consider security, performance, and access control policies and model supports.

5.1. Security

ABE provides confidentiality and data integrity when used in a public environment with a large scope (e.g., cloud) of users. However, relying only on user-specified attributes may create various security issues from the perspectives of key management processes and intentional threats or attacks.

5.1.1. Key Management

Secure communication: To distribute keys to users, a secure communication channel between a user and the key management service is required such that an Secure Sockets Layer (SSL)-like connection is a common solution for a large-scale ABE system. Hence, it is important for users to authenticate themselves through – for example – usernames, passwords, or public-key pairs managed on user devices.

Non-repudiation: Because the key management service generates private keys for users, it may decrypt without authorization. If the secret key is abused, it is difficult to judge whether the abused private key comes from the users or the key management service [WZZGZZ]. Therefore, ABE systems are difficult for non-repudiation. This may not be an issue for organizations that host their own key management service and are willing to trust their system administrators or that do not require non-repudiation. A caveat is that the key management service must be highly trusted.

User tracking: The problem of the basic ABE scheme is that there is no mechanism to identify the user who issued a key. The secret key does not contain the specific information of users, so it is impossible to identify the user who misuses the distributed key or shares their secret key with other users [WZZGZZ]. A tracking function might be required for higher security requirements. However, providing traceability may infringe on a user's privacy by exposing the user's identifier value when the key is issued by the attribute verification [HL] process of the key management service.

Key escrow: Because a user's private key is generated through the key management service, ABE has the capability of key escrow. However, such a capability can be a positive or negative feature depending on the usages, such as a private organization using it for security control while sacrificing the privacy of its users. Several variant ABE systems have been proposed that remove the escrow by replacing encryption or key generation processes with certificate-based encryption [CRAI], secure key issuing cryptography [BCEKJS], or certificateless cryptography [AP].

Key revocation: One of the major advantages of any identity encryption scheme is that a third party's secret key can be destroyed after all users have been issued keys and if there is only a finite number of users. This can also take place for ABE systems because it assumes that keys are always valid once issued, and there is no method for key revocation to handle secret keys due to expiry of embedded attributes, faulty access policies, or key compromise. Key revocation for ABE can be handled by including the expiry time and date among the attributes, periodic refreshing, and revocation lists [GOLIC].

5.1.2. Threats and Attacks

Compromised key management server: ABE relies on a key management service for the generation of cryptographic keys. If the key management service is compromised, data protected by the public-private key pair used are also compromised. Hence, a key management service is a high-value target for adversaries who wish to decrypt all ciphertexts. A countermeasure for this vulnerability is to frequently update the main private-public key pairs with new independent key pairs for all users. However, this complicates the key management process.

Collusion: CP-ABE users can infer other users' attributes through collusion with each other, generate another user's secret key with the inferred attributes, and share private decryption keys (and maybe attribute certificates if applicable). Therefore, a key management service must generate secret keys by applying various variables in addition to the user's attributes. If data leak through a collusion attack on resource providers, security technology is required so that only a legitimate user can decrypt and view the ciphertext [MIHIR].

Fully secure: Fully secure (i.e., adaptive) ABE is more advantageous than selectively secure ABE because it does not require adversaries to specify their target access policies or attribute lists until they receive the system public keys. General ABE schemes based on prime order groups for cryptography lack the proof of fully secure, so efforts in proof methods are needed to promote more secure and efficient designs. Existing fully secure ABE solutions are usually designed on composite-order groups or re-encryption[6] systems, and complex assumptions are involved in the security proof [ZDXSLZ][HL].

Integrity: Outsourcing servers for an ABE system requires trust so that the decrypted ciphertext is a legitimate message based on legitimate user attributes. Additionally, the message uploaded to the resource provider can be falsified, and it is unknown whether the value calculated by the outsourcing server is the correct value. Accordingly, it is necessary to verify whether the user's final decrypted value is the original message from the data owner [HL]. Specifically, verification processes are required to prove that the results from key management and the resource servers are properly computed.

Quantum resistant: ABE systems are insecure against quantum computer attacks. Many public-key encryption schemes, including ABE, require security enhancements to resist possible quantum attacks. Although lattice-based algorithms can resist quantum attacks, there are only a

[6] Proxy re-encryption (PRE) allows a proxy to convert a ciphertext encrypted under one key into an encryption of the same message under another key. The main idea is to place as little trust and reveal as little information to the proxy as necessary to allow it to perform its translations [UMAS].

few lattice-based ABE constructions that are selectively secure. In addition, lattice-based schemes lack practicability because they have only been considered secure for inefficiently large parameters. Thus, more attention should be paid to quantum-resistant ABE for better security assurance [ZDXSLZ][DKW][WWW].

5.2. Performance

A performance bottleneck of ABE is the high computation overhead due to the complexity of the embedded bilinear pairing algorithm and the requirement for large security parameters [OD] to cover a wider scope of attributes.

5.2.1. Computational Complexity

Most of the existing ABE schemes (e.g., revocable ABE, accountable ABE, policy-hiding ABE, ABE with policy updating, and multi-authority ABE) have a high order of computational complexity for typical cryptographic operations that are much greater than that of symmetric and traditional PKE [ZDXSLZ], including exponentiation, point multiplication, group arithmetic operations, and especially, the bilinear pairing calculation. Therefore, it may be more efficient to apply alternative schemes like non-bilinear pairing-based ABE schemes [KAB] for practical uses of ABE, especially in a resource-constrained system environment, such as IoT.

5.2.2. Keys and Ciphertext Size

Both CP-ABE and KP-ABE schemes have overhead issues with key size. In CP-ABE, the public key size can be fixed with a hash function or made linear to the number of attributes applied. In KP-ABE, the size of the public key is linear to the maximum number of attributes applied to the system [GOLIC]. The size of the ciphertext depends on the number of available attributes contained in the access structure, and it increases linearly with the number of attributes, which requires significant system storage and computation time for users to decrypt ciphertext. Therefore, it may be necessary to introduce assistant systems to accommodate the heavier computation (e.g., increase the computational efficiency with architecture options, such as proxy devices [MHH]), but a verification process is needed to prove that the results on the outsourcing server are properly computed [HL]. Further, CP-ABE is not efficient for modern enterprise environments when compared to KP-ABE due to the resource access policies needed for central management such that when a policy changes, secret keys need to be reestablished for users. In contrast, KP-ABE is made more flexible by its broadcast type of encryption [UMAS] for user policies.

5.2.3. Physical Limitations

The physical properties of ultra-low energy mobile devices [OD] include low processing power, a distributed nature, and a lack of standardization [RPRMK], which limit their capabilities to perform complex computations to support ABE’s (especially CP-ABE’s) encryption and decryption. These drawbacks hinder ABE adoption for advanced applications, such as IoT and cloud systems, due to the much greater heterogeneity and resource restrictions of their devices. Therefore, further investigation into the application of ABE is needed to decide device sizing against levels of computation, communication, and performance. Mobile computing for ABE has established its own paradigm, which has extended to researching whether ABE for mobile devices can be translated to the application of IoT [MHH].

Researchers are currently working on blockchain fundamentals and customizing blockchain-based ABE models for IoT applications to provide privacy and minimize computational overhead. For example, [QYLPYH] describes the use of a lightweight blockchain ABE to outsource decryption based on the blockchain, which can be extended to effectively reduce the burden of encryption computation on the user side. Blockchain technology can also provide integrity (i.e., the secret key does not contain specific information about users who may share their secret keys with other users) and the non-repudiation of data, as well as prevent the leaking of sensitive information from the ABE access structure [WZZGZZ].

5.3. Access Control Policies and Model Supports

In addition to functionalities like revocation, accountability, attribute privacy protection, policy updating, decentralization (multi-authorities), and key hierarchy for practical access control system deployments [ZDXSLZ], the applicable access policy structure for ABE is restricted to supporting non-monotone and stated policy rules [TKN]. For example, CP-ABE allows data owners to define their own access policies (structures) by attributes and, thus, support complex access control policy structure. However, by only associating attributes, decryption keys are organized logically as a static set. Users can only use all possible combinations of attributes in the set of keys issued to compose their policies, and it has restrictions for specifying policies, attribute managements (e.g., applying environment conditions and dynamic attributes), and the application of deny rules, which fails to satisfy the enterprise requirements of access control in terms of flexibility and dynamic requirements [BS].

In KP-ABE, the secret key and ciphertext relate to a set of attributes to offer fine-grained access control [BCSES] for which permission evaluation depends only on the resources' attributes. The resource provider (i.e., data owner or encrypted) cannot specify the access policy except by choosing descriptive attributes for permissions. This means that there is no choice but to trust the key issuer. Such accountability for user secret keys provides fine-grained access without flexibility or scalability [BS], making it unsuitable for certain applications unless supported by re-encryption techniques [GOLIC].

Further, from the perspective of full access, action capabilities – including write, modification, and execute privileges – are not straightforwardly implemented in ABE schemes and, thus, require other layers of operational support.

6. Conclusion

ABE supports fine-grained access control for encrypted data and is a cryptographic scheme that goes beyond the all-or-nothing approach of public-key encryption schemes. This document reviewed the interplay between cryptography and the access control of ABE, from fundamental theories on which the ABE scheme is based to the various main algorithms of IBE, CP-ABE, and KP-ABE, as well as considerations for deploying ABE systems.

Due to security, performance, and access control policy and model support considerations, the deployment and adoption of ABE have been slow. Few widely used commercial products (e.g., Microsoft Word, Excel, PowerPoint) use it to date. This shortcoming of selective and flexible access control may impact its adoption for government and commercial applications as well as applications for highly secure demanding areas (e.g., life sciences, healthcare, financial sectors) [ELT]. However, with additional exploration and the support of processing systems, a mature ABE technology can address these challenges.

References

[ADI] Adi S (1984) Identity-Based Cryptosystems and Signature Schemes. Lecture Notes in Computer Science. Vol. 196. Springer, pp 47–53. https://doi.org/10.1007/3-540-39568-7_5

[AP] Al-Riyami SS, Paterson KG (2003) Certificateless public key cryptography, ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, Proceedings. Lecture Notes in Computer Science. Vol. 2894. Springer. pp. 452–473. doi:10.1007/978-3-540-40061-5_29. Available at https://eprint.iacr.org/2003/126.pdf

[BCEKJS] Byoungcheon L, Colin B, Ed D, Kwangjo K, Jeongmo Y, Seungjae Y (2004) Secure key issuing in ID-based cryptography, 2004 ACSW Workshops – the Australasian Information Security Workshop (AISW2004), Vol. 32. Australian Computer Society. pp. 69–74. Available at https://dl.acm.org/doi/10.5555/976440.976449

[BCSES] Bagyalakshmi C, Samundeeswari ES (2018) A Survey on Attribute-Based Encryption Techniques in Data Security Using Cloud Environment, Journal of Advanced Research in Dynamical and Control Systems, Vol. 10, 03-Special Issue. Available at https://www.researchgate.net/publication/346095629_A_survey_on_attribute_based_encryption_techniques_in_data_security_using_cloud_environment

[BETH] Bethencourt J (2015) Intro to Bilinear Maps, Computer Sciences Department Carnegie Mellon University. Available at https://people.csail.mit.edu/alinush/6.857-spring-2015/papers/bilinear-maps.pdf

[BF] Boneh D, Franklin M (2001) Identity-Based Encryption from the Weil Pairing, Advances in Cryptology — CRYPTO 2001. CRYPTO 2001. Lecture Notes in Computer Science, vol 2139. Springer. https://doi.org/10.1007/3-540-44647-8_13

[BONEH] Boneh D, et. al (2002) IBE Secure E-mail, Stanford University. Available at https://crypto.stanford.edu/ibe/

[BS] Bagyalakshmi C, Samundeeswari ES (2018) A survey on attribute-based encryption techniques in data security using cloud environment, Journal of Advanced Research in Dynamical and Control Systems 10(03):926-931. Available at https://www.researchgate.net/publication/346095629_A_survey_on_attribute_based_encryption_techniques_in_data_security_using_cloud_environment

[BSW2007] Bethencourt J, Sahai A, Waters B (2007) Ciphertext-Policy Attribute-Based Encryption, 2007 IEEE Symposium on Security and Privacy (SP '07). https://doi.org/10.1109/SP.2007.11

[BSW2011] Boneh D, Sahai A, Waters B (2011) Functional Encryption: Definitions and Challenges, In: Ishai, Y. (eds) Theory of Cryptography. TCC 2011. Lecture Notes in Computer Science, vol 6597. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19571-6_16

[BSW2012] Boneh D, Sahai A, Waters B (2012) Functional encryption: a new vision for public-key cryptography, Communications of the ACM Volume 55 Issue 11 November 2012 pp 56–64. https://doi.org/10.1145/2366316.2366333

[BUCH] Buchanan B (2022) Pairing-based Cryptography, OBE presentation. Available at https://www.youtube.com/watch?v=4zu-kXIiXA4

[BUTE] Buterin V (2017) Exploring Elliptic Curve Pairings, *Medium*. Available at https://medium.com/@VitalikButerin/exploring-elliptic-curve-pairings-c73c1864e627

[CORB] Corbellini A (2015) Elliptic Curve Cryptography: finite fields and discrete logarithms. Available at https://andrea.corbellini.name/2015/05/23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/

[CRAI] Craig G (2003) Certificate-based encryption and the certificate revocation problem, Biham, Eli (ed.). Advances in Cryptology – EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, Proceedings. Lecture Notes in Computer Science. Vol. 2656. Springer. pp. 272–293. https://doi.org/10.1007/3-540-39200-9_17

[DANI] Daniel RLB (2009) SEC 1: Elliptic Curve Cryptography, https://www.secg.org Certicom Research Version 2.0. Available at https://www.secg.org/sec1-v2.pdf

[DKW] Datta P, Komargodski I, Waters B (2021) Decentralized Multi-Authority ABE for DNFs from LWE, A major revision of an IACR publication in EUROCRYPT 2021. Available at https://eprint.iacr.org/2020/1386.pdf

[ELT] Eldefrawy K, Lepoint T, Tam L (2022). In-App Cryptographically-Enforced Selective Access Control for Microsoft Office and Similar Platforms. Cyber Security, Cryptology, and Machine Learning. CSCML 2022. Lecture Notes in Computer Science, vol 13301. Springer, Cham. https://doi.org/10.1007/978-3-031-07689-3

[FIPS186-5] National Institute of Standards and Technology (2023) Digital Signature Standard (DSS). (U.S. Department of Commerce, Washington, DC), Federal Information Processing Standards Publication (FIPS) 186-5. https://doi.org/10.6028/NIST.FIPS.186-5

[GOLIC] Golic J (2018) Attribute-based Encryption and Signatures, presentation. Available at https://www.youtube.com/watch?v=l0yCigNqv5w

[GPSW] Goyal V, Pandey O, Sahai A, Waters B (2006) Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data, CCS '06: Proceedings of the 13th ACM conference on Computer and communications security pp 89–98. Available at https://eprint.iacr.org/2006/309.pdf

[GW] Gong J, Wee H (2020) Adaptively Secure ABE for DFA from K-Lin and More, International Association for Cryptologic Research. Available at https://www.youtube.com/watch?v=I_-YzRYWSoE

[HALL] Hallsted S (2015) Attribute-based Encryption, Presentation on theme: "Attribute-Based Encryption." Available at https://slideplayer.com/slide/3246359/

[HL] Hwang Y, Lee I (2020) A Study on CP-ABE-Based Medical Data Sharing System with Key Abuse Prevention and Verifiable Outsourcing in the IoMT Environment, Special Issue of Internet of Medical Things in Healthcare Applications, Department of Computer Science and Engineering, Soonchunhyang University, Korea. https://doi.org/10.3390/s20174934

[HR] Nurmamat Helil N, Rahman K (2017) CP-ABE Access Control Scheme for Sensitive Data Set Constraint with Hidden Access Policy and Constraint Policy, Security and Communication Networks, vol. 2017, Article ID 2713595. https://doi.org/10.1155/2017/2713595

[HUBWIZ] hubwiz.com (2020) Bilinear Pairs. Available at http://blog.hubwiz.com/2020/06/04/bilinear-pairing/

[IE] IEEE Computer Society (2004) IEEE Standard Specifications for Public-Key Cryptography— Amendment 1: Additional Techniques. Available at https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1335427

[IRON] IRONCORE LABS (2018) Pairing-Based Transform Cryptography (Proxy Re-Encryption - PRE), Presented at DEF CON 26. Available at https://www.slideshare.net/IronCoreLabs/pairing-based-transform-cryptography-proxy-reencryption-pre

[KAB] Karati A, Amin R, Biswas G.P. (2016) Provably Secure Threshold-Based ABE Scheme Without Bilinear Map, Arab J Sci Eng 41, 3201–3213 (2016). https://doi.org/10.1007/s13369-016-2156-9

[KB] Kar D, Bezawada B (2018) Attribute-Based Encryption, Presentation on theme: “Attribute Based Encryption” – Presentation transcript, Colorado State University. Available at https://slideplayer.com/slide/13691307/

[KKB] Kavuri A, Kancherla GR, Bobba, B (2017) An Improved Integrated Hash and Attributed-based Encryption Model on High Dimensional Data in Cloud Environment, International Journal of Electrical and Computer Engineering, 7(2), 950. https://doi.org/10.11591/ijece.v7i2.pp950-960

[LXYS] Lin G, Xia Y, Ying C, Sun Z (2019) F2P-ABS: A Fast and Secure Attribute-Based Signature for Mobile Platforms, Security and Communication Networks Research Article of Hindawi. https://doi.org/10.1155/2019/5380710

[MATA] Matarazzo, L (2015) A Look Into Elliptic Curve Cryptography (ECC). Available at https://www.youtube.com/watch?v=5wDvlq-MrLg

[MEFF] Meffert D (2009) Bilinear Pairings in Cryptography, Radboud Universiteit Nijmegen Computing Science Department. Available at https://www.math.ru.nl/~bosma/Students/MScThesis_DennisMeffert.pdf

[MD] Moody D, Peralta R, Perlner R, Regenscheid A, Roginsky A, Chen L (2015) Report on Pairing-based Cryptography, Journal of Research of the National Institute of Standards and Technology, Volume 120 (2015). https://doi.org/10.6028/jres.120.002

[MPR] Maji HK, Prabhakaran M, Rosulek M(2011) Attribute-Based Signatures, Kiayias, A. (eds) Topics in Cryptology – CT-RSA 2011. CT-RSA 2011. Lecture Notes in Computer Science, vol 6558. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19074-2_24

[MHH] Moffat S, Hammoudeh M, Hegarty R (2017) A Survey on Ciphertext-Policy Attribute-based Encryption (CP-ABE) Approaches to Data Security on Mobile Devices and its Application to IoT, ICFNDS '17: Proceedings of the International Conference on Future Networks and Distributed Systems Article No.: 34. https://doi.org/10.1145/3102304.3102338

[MIHIR] Mihir B (2021) Invitation to Modern Cryptography – Identity-based Encryption, presentation for CSE207, UCSD Computer Science. Available at https://www.youtube.com/watch?v=kdf0u2TGgNg

[MMSC] Microprocessor and Microcomputer Standards Committee of the IEEE Computer Society (2004) IEEE Standard Specifications for Public-Key Cryptography—

Amendment 1: Additional Techniques, IEEE Computer Society. Available at https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1335427

[MOBI] Mobilefish.com (2016) Blockchain tutorial 11: Elliptic Curve key pair generation. Available at https://www.youtube.com/watch?v=wpLQZhqdPaA

[MPPRRC] Moody D, Peralta R, Perlner R, Regenscheid A, Roginsky A, Chen L (2015) Report on Pairing-based Cryptography, Journal of Research of the National Institute of Standards and Technology Volume 120. http://dx.doi.org/10.6028/jres.120.002

[MUKH] Mukhopadhyay D (2017) Attribute-Based Encryption (ABE), Department of Mathematics IIT Kharagpur NPTEL Online Certification Course. Available at https://www.youtube.com/watch?v=ZogQMKzoQdw

[MY] Mahto D, Yadav D K (2017) RSA and ECC: A Comparative Analysis, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017) pp. 9053-9061. Available at https://www.ripublication.com/ijaer17/ijaerv12n19_140.pdf

[OD] Odelu V, Das AK (2016) Design of a new CP-ABE with constant-size secret keys for lightweight devices using elliptic curve cryptography, Security and Communication Networks Security Comm. Networks 2016, Published online in Wiley Online Library. https://doi.org/10.1002/sec.1587

[QIAU] Qiau P (2020) Bilinear Pairing for Cryptograph Application, Hyperchain Technology Inc. Available at https://zhuanlan.zhihu.com/p/321902465

[QYLPYH] Authors: Qin X, Yang Z, Li Q, Pan H, Yang Z, Huang Y(2022) Attribute-based encryption with outsourced computation for access control in IoTs. ASSE' 22: 2022 3rd Asia Service Sciences and Software Engineering Conference, pp 66–73. https://doi.org/10.1145/3523181.3523191

[ROBI] Robinson E (2015) Elliptic Curve Cryptography, YSL Information Security – Public-Key Cryptography. Available at https://player.slideplayer.com/16/4898906/#

[RPRMK] Rahulamathavan Y, Phan RC, Rajarajan M, Misra S, Kondoz A (2017) Privacy-preserving Blockchain-based IoT Ecosystem using Attribute-based Encryption, 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). https://doi.org/10.1109/ANTS.2017.8384164

[SCHOOF] Schoof R (1995), Counting points on elliptic curves over finite fields, Journal de Theorie des Nombres de Bordeaux 7, 219-254. Available at http://www.numdam.org/item/JTNB_1995__7_1_219_0.pdf

[SHINDE] Shinde S (2020) Privacy Teaching Series: What is Functional Encryption? OpenMined Functional Encryption. Available at https://blog.openmined.org/privacy-teaching-series-what-is-functional-encryption

[SP800-162] Hu VC, Ferraiolo D, Kuhn R, Schnitzer A, Sandlin K, Miller R, Scarfone K (2014) Guide to Attribute-Based Access Control (ABAC) Definition and Considerations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-162, Includes updates as of August 02, 2019. https://doi.org/10.6028/NIST.SP.800-162

[SP800-186] Chen L, Moody D, Regenscheid AR, Robinson AY, Randall K (2023) Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve

Domain Parameters. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-186. https://doi.org/10.6028/NIST.SP.800-186

[SP800-56A] Barker EB, Chen L, Roginsky AL, Vassilev A, Davis R (2018) Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-56A, Rev. 3. https://doi.org/10.6028/NIST.SP.800-56Ar3

[ST] Tahat N, Shatnawi S (2022) New Signature Scheme Based on Elliptic Curve and Factoring Problems Using Chaotic Map, Journal of Applied Security Research. Available at https://www.tandfonline.com/doi/full/10.1080/19361610.2022.2041157

[SW] Sahai A, Waters B (2005) Fuzzy Identity-Based Encryption Cryptology, Proceedings of the 24th annual international conference on Theory and Applications of Cryptographic Techniques 2005. Pages 457–473. https://doi.org/10.1007/11426639_27

[TDN] Ţiplea FL, Dragan C, Nica A (2017) Key-Policy Attribute-Based Encryption from Bilinear Maps, International Conference for Information Technology and Communications. https://doi.org/10.1007/978-3-319-69284-5_3

[TKN] Tomida J, Kawahara Y, and Nishimaki R (2021) Fast, Compact, and Expressive Attribute-Based Encryption, Designs, Codes and Cryptography (2021) 89:2577–2626. https://doi.org/10.1007/s10623-021-00939-8

[UMAS] Umashankar SKA (2016) A Review on Attribute-Based Encryption (ABE) and ABE Types, Semantic Scholar - Computer Science. Available at https://www.semanticscholar.org/paper/A-Review-on-Attribute-Based-Encryption-(ABE)-and-Umashankar/c3910ecacc2a6c1ceb3c54eb493f2c8c801e9e25

[WF] Wall B, Frederickson C (2018) Implementing a Library for Pairing-based Transform Cryptography, DEF CON 26 Crypto and Privacy Village Conference video 23:09. Available at https://cryptovillage.org/implementing-a-library-for-pairing-based-transform-cryptography

[WSOE] Gong J, Wee H (2020) Adaptively Secure ABE for DFA from K-Lin and More, International Association for Cryptologic Research, Available at https://www.youtube.com/watch?v=I_-YzRYWSoE

[WWW] Water B, Wee H, Wu D J (2022) Multi-Authority ABE from Lattices without Random Oracles, A major revision of an IACR publication in TCC 2022. Available at https://eprint.iacr.org/2022/1194.pdf

[WZZGZZ] Wu A, Zhang Y, Zheng X, Guo R, Zhao Q, Zheng D (2019) Efficient and Privacy-preserving Traceable Attribute-based Encryption in Blockchain, Annals of Telecommunications 74:401–411. https://doi.org/10.1007/s12243-018-00699-y

[ZDXSLZ] Zang Y, Deng RH, Xu S, Sun J, Li Q, Zeng D (2020) Attribute-based Encryption for Cloud Computing Access Control: A Survey, Computing Surveys, Vol. 53, No. 4, Article 83. https://doi.org/10.1145/3398036