A Practical Certificate and Identity Based Encryption Scheme and Related Security Architecture

Tomasz Hyla and Jerzy Pejaś

West Pomeranian University of Technology in Szczecin
Faculty of Computer Science and Information Technology, Poland
{thyla,jpejas}@wi.zut.edu.pl

Abstract. Group encryption schemes based on general access structures can be used to build advanced IT systems, which store and manage confidential documents. The paper proposes a reference architecture of public key cryptography infrastructure required to implement CIBE-GAS scheme. The CIBE-GAS scheme is a certificate-based group-oriented encryption scheme with an effective secret sharing scheme based on general access structure and bilinear pairings. The security architecture required to implement the scheme must be compliant with common standards and technical specifications, e.g. X.509 certificate format and XML-encryption standard for messages. In order to encrypt arbitrary-length messages, we also suggest a new CIBE-GAS-H scheme with a key encapsulation mechanism based on the techniques of Bentahar *et al.*, and combined with one-time symmetric-key encryption.

Keywords: group encryption, general access structures, security architecture, pairing based cryptosystem.

1 Introduction

The certificate and ID-based group encryption scheme with bilinear pairings allows to design the cryptographic access control mechanisms for protection of sensitive information. Due to these mechanisms, the information can be stored in the network in an encrypted form and decrypted only by authorised users [1, 2]. However, the development of a system that uses such mechanisms requires consideration of many additional architectural problems. These problems include trust and certification models selection and choice of proper access structures.

Public keys certification models correspond to different methods of public keys' management, including theirs generation, certification, distribution and revocation. The architecture consists of a set of well-defined components, their functions and relations between them, including the trust relationship. The primary purpose of trust models is creation of trust relationship between any entities within the same or between different key management architectures (KMA). The trust models are based on the certification models with distinguished local trust authorities (TA).

An access structure is a rule that defines how to share a secret, or more widely, who has an access to particular assets in IT system. Access structures can be classified

K. Saeed et al. (Eds.): CISIM 2013, LNCS 8104, pp. 190–205, 2013.

as structures with and without threshold. Although threshold access structures are frequently used, the non-threshold structures (called also as general access control) are more versatile.

ID-based cryptosystems (IBC) had received considerable interest to cryptographic researchers since A. Shamir's work [3]. However, the question was how to construct effectively such systems. After slightly more than a decade ago, in 2001, Boneh and Franklin [4] proposed the first practical cryptographic IBE scheme based on bilinear pairings. Since then many extensive researches have been done, but only limited results of them have been implemented into commercial products. This is mainly due to the low commercial maturity of ID-based cryptography schemes measured by the number of available products and standards.

However, the current state of the commercialising IBC schemes and developing standards is slowly changing. We know three practical implementations of identity based cryptographic techniques:

- a commercial product for encrypted e-mail based on Boneh-Franklin IBE scheme (Voltage Security Inc, http://www.voltage.com);
- an application for secure e-mail encryption based on the Sakai-Kasahara ID- based key encapsulation mechanism (Trend-Micro, www.trendmicro.com/us);
- a smart-card implementation of IBE based on the Boneh-Franklin scheme (Gemalto, http://www.gemalto.com/press/gemplus/2004/id_security/02-11-2004-Identity-Based_Encryption.htm).

Among standards, there are:

- a draft standard of IEEE P1363.3/D1 for *Identity-based Public-key Cryptography Using Pairings* [5];
- RFC 6508 *Sakai-Kasahara Key Encryption (SAKKE)* [6];
- RFC 5091 *Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems* [7];
- RFC *5408 Identity-Based Encryption Architecture and Supporting Data Structures* [8].

1.1 Our Contributions

In this paper we introduce a new practical CIBE-GAS-H scheme (the modification of our previous CIBE-GAS scheme [1]). The new CIBE-GAS-H scheme is designed to work with arbitrary length messages while original CIBE-GAS scheme works with limited length messages only. We also propose reference security architecture to implement CIBE-GAS-H scheme, together with analysis of security issues related to implementation of the scheme.

We propose a hybrid system, which merges traditional PKI solutions with our CIBE-GAS-H scheme in order to achieve the good scalability, comparable with traditional X.509 based architectures. Combining these two cryptosystems in a single framework has advantages of the traditional PKI and ID-based public key cryptography and, for example, allows to authenticate both users of PKI domain and ID-based

domain. The framework defines data structures that can be used to implement the proposed hybrid trust system. These structures are required to support on-line interactions between CIBE-GAS users and PKI/TA management entities. We define messaging system, describing how the components work together, and data structures that support the system operation (with the extension of the standard X.509 certificate).

1.2 Paper Organisation

The paper is organized as follows. Section 2 contains description of basic bilinear pairings and short description of our CIBE-GAS scheme introduced in [1]. Section 3 introduces CIBE-GAS-H scheme, which is an extension of CIBE-GAS scheme for arbitrary length messages. Section 4 contains description of architecture for CIBE-GAS-H scheme and related security issues. Moreover, the section provides description of current standards that can be used in CIBE-GAS-H system architecture. The paper ends with summary and conclusions.

2 Background

2.1 Bilinear Groups and Security Assumptions

A pairing $\hat{e}$ is defined as a bilinear map between elements of two finite, cyclic and additive groups G_1 and G_2 to a third finite cyclic group G_T defined multiplicatively. Both of G_1 and G_2 are of prime order q, as it is in the case of G_T. In practice, pairing $\hat{e}$ allows to solve certain problem in one group, even if the problem is said to be hard in another group.

Depending upon the structure of the group G_2, a bilinear pairing can be classified as one of the following three types [9, 10]:

- **Type 1:** $G_2 = G_1$.
- **Type 2:** $G_2 \neq G_1$, but there is an efficiently computable isomorphism φ from G_2 to G_1.
- **Type 3:** $G_2 \neq G_1$, and there are no efficiently computable isomorphism φ from G_2 to G_1.

Note that since G1 and G2 are both cyclic groups of the same prime order, they are certainly isomorphic.

Here we simply consider symmetric pairings (i.e. the case of Type 1) in prime-order groups, using notations similar to those presented by Al-Riyami, S., et al. [11].

Definition 1. Let $(G_1, +)$ and $(G_T, \cdot)$ be two cyclic groups of some prime order $q>2^k$ for security parameter $k \in N$. The bilinear pairing is given as $\hat{e}: G_1 \times G_1 \rightarrow G_T$ and must satisfy the following three properties:

1. **Bilinearity:** $\hat{e}(aP, bQ) = \hat{e}(abP, Q) = \hat{e}(P, abQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$ and all $a, b \in Z_q^*$; this can be restated in the following way: for $P, Q, R \in G_1$, $\hat{e}(P+Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$ and $\hat{e}(P, Q+R) = \hat{e}(P, Q)\hat{e}(P, R)$.
2. **Non-degeneracy:** some $P, Q \in G_1$ exists such that $\hat{e}(P, Q) \neq 1_{G_2}$; in other words, if P and Q are two primitive elements of G_1, then $\hat{e}(P, Q)$ is a generator of G_2.
3. **Computability:** given P, Q$\in G_1$, an efficient algorithm computing $\hat{e}(P, Q)$ exists.

Note that a pairing $\hat{e}$ is symmetric, since $\hat{e}(aP, bP) = \hat{e}(P, P)^{ab} = \hat{e}(bP, aP)$. The most commonly used pairings arise from the theory of elliptic curves, where G_1 is subgroup of points on an elliptic curve over a finite field, whereas G_T is a subgroup of the multiplicative group of a finite field.

2.2 One-Time Symmetric-Key Encryption

A one-time symmetric-key encryption (*SKE*) scheme consists of two deterministic polynomial time secret key (*SK*) algorithms, E_{SK} and D_{SK}, where key, message and ciphertext spaces are given by $K_{SK}(\kappa)$, $M_{SK}(\kappa)$, $C_{SK}(\kappa)$ for some security parameter $\kappa \in Z^+$. A deterministic encryption algorithm E_{SK} takes a message $M \in M_{SK}(\kappa) = (0,1)^*$ and a key $K \in K_{SK}(\kappa)$ as inputs, and outputs a ciphertext $C = E_{SK}(K, M)$. Another deterministic algorithm D_{SK} is a decryption algorithm that takes a ciphertext C and a key K as inputs and outputs a message $M = D_{SK}(K, C)$ or $\perp$, when some error has occurred.

We assume that the scheme is sound, i.e. for all M we have $D_{SK}(K, E_{SK}(K, M)) = M$ and the key length $|M|$ is a polynomial function of the security parameter κ.

We do not define concrete one-time symmetric-key encryption scheme $SKE = (E_{SK}, D_{SK})$, but we assume that this scheme fulfils the security requirements given in [12, 13] and is secure against passive attacks (standard algorithms like AES, Blowfish or chaos-based ciphers, e.g. [14] can be used).

2.3 Full Certificate-Based Encryption Scheme with General Access Structure

In this Section first we review our Certificate-Based Encryption Scheme with General Access Structure (CIBE-GAS) [1]. This group encryption algorithm is intended to encrypt short plaintext messages M, which are a bit strings of length p. In Section 3 extension of CIBE-GAS scheme allowing to encrypt arbitrary length messages is presented.

Definition 2 (CIBE-GAS scheme). Assume that are given: *n*–element set containing all shareholders $U = \{u_1, u_2, ..., u_n\}$, *m*–element access structure[1] $\Gamma = \{A_1, A_2, ..., A_m\}$, dealer $D \notin U$ and combiner $Com \in U$. Eight probabilistic algorithms specify the original certificate-based encryption scheme with general access structure (CIBE-GAS).

- **Setup** $(1^\kappa) \rightarrow (s, params)$
  Trusted Authority (TA) runs this algorithm. The algorithm takes a security parameter 1^κ as an input and returns the master private key $s \in_R Z_q^*$, the master public key P_0 and the system parameter *params*:

$$params = \{G_1, G_2, \hat{e}, q, P, P_0, H_1, H_2, H_3, H_4, H_5, H_6\} \tag{1}$$

  where P – the primitive element of G_1, $P_0 = sP$ - the public key, $H_1 : \{0,1\}^* \times G_1 \times G_1 \rightarrow G_1^*$, $H_2 : \{0,1\}^* \times G_1 \times G_1 \rightarrow Z_q^*$, $H_3 : G_2 \times \{0,1\}^* \rightarrow Z_q^*$, $H_4 : \{0,1\}^p \times \{0,1\}^p \rightarrow Z_q^*$, $H_5 : G_2 \rightarrow \{0,1\}^p$ and $H_6 : \{0,1\}^p \rightarrow \{0,1\}^p$ are secure hash functions. TA runs the algorithm and, after completion, keeps secret the master private key s, while P_0 and *params* are publicly accessible by all users in the system.

- **SetSecretValue** $(params) \rightarrow (s_E, Pk_E)$
  The entity *E*, i.e. any shareholder $u_i \notin U$, dealer $D \notin U$ and combiner $Com \in U$ runs this algorithm. The algorithm takes as an input the *params* and outputs the secret value s_E and the public key Pk_E for *E*.

- **CertGen** $(s, params, ID_E, Pk_E) \rightarrow Cert_E$
  This algorithm takes as an input the master private key s, the system parameter *params*, and an entity *E*'s identity ID_E and *E*'s public key Pk_E. It outputs the certificate $Cert_E$. The TA runs this algorithm once for each entity.

- **SetPublicKey** $(params, ID_E, Pk_E, Cert_E) \rightarrow \{yes, no\}$
  This algorithm takes as an input a system parameter *params*, an entity *E*'s identity ID_E, *E*'s public key Pk_E and $Cert_E$, and returns the positive result if the certificate is valid or the negative result in opposite case. It is run by the entity, and in positive case the resulting public key Pk_E is widely and freely distributed.

---

[1] The set $\Gamma = \{A_j \mid j = 1, 2, \ldots, m\} \subseteq 2^U$ is an *access structure*, if any secret *x* can be reconstructed by gathering all the shares x_i secretly owned by u_i in A_j. All sets in access structure Γ are called *authorised* or *qualified sets*.

– **ShareDistribution** ($params, prm_D, prm_U, \Gamma$) $\rightarrow pubVal$
This algorithm is run by a dealer *D*. It takes as an input a system parameter *params*, dealer parameters prm_D, shareholders parameters prm_U and current access structure Γ, and returns public values for each authorised subset $A_j \in \Gamma$. We assume that:

$prm_U = (Cert_U, ID_U) = \{ (Cert_{u_i}, ID_{u_i}) \mid u_i \in U \}$
$prm_D = \{ s_d, ID_d, Pk_d, Cert_d \}$
$pubVal = (\beta, f(1), Y, Y_{-1}, (d_1,\ldots,d_m), (\gamma_1,\ldots,\gamma_m), (k_{1,1},\ldots,k_{1,m}; \ldots, k_{n,1},\ldots,k_{n,m}))$

– **Encryption** ($M, ID_d, Pk_d, F, pubVal, params; \sigma$) $\rightarrow C$

On input of an limited length message $M \in \{0,1\}^p$, the dealer *D*'s identity ID_d, his public key Pk_d, the filter *F*, which superimposed on the access structure Γ allows only privileged groups with indexes from F to decrypt information *M*, publicly known parameters *pubVal*, the system *params* and possibly some randomness $\sigma \in \{0,1\}^p$, this algorithm outputs a ciphertext $C = (C_1, C_2, C_3, C_4, C_5, C_6)$. This algorithm is run when the dealer creates a new encrypted message. The ciphertext is calculated as follows:

$$r = H_4(\sigma, M),\ C_1 = r(H_2(ID_d, Pk_d)P + X_d) \tag{2}$$

$$C_2 = \sigma \oplus H_5(\hat{e}(P,Y)^r),\ C_3 = m \oplus H_6(\sigma),\ C_4 = \hat{e}(P, f(1)P)^r \tag{3}$$

$$C_5 = \{ v_k = \hat{e}(P, \gamma_k P)^r, \forall k \in F \subseteq 2^m \},\ C_6 = rY_{-1} \tag{4}$$

– **SubDecryption** ($C, ID_{u_{i_j}}, Cert_{ID_{u_{i_j}}}, s_{i_j}, A_j, params, pubVal$) $\rightarrow \delta_{i_j,j}$

Every shareholder (with an identity $ID_{u_{i_j}}$ and a certificate $Cert_{ID_{u_{i_j}}}$) from the privileged subset $u_{i_j} \in A_j \in \Gamma$, where $A_j = \{u_{1_j}, u_{2_j}, \ldots\}$, runs this algorithm and partially decrypts ciphertext *C* using his share s_{i_j}. The decrypted value $\delta_{i_j,j}$ is sent to a combiner.

– **Decryption** ($C, ID_d, Pk_d, A_j, (\delta_{1_j,j}, \ldots, \delta_{|A_j|_j,j}), params, pubVal$) $\rightarrow M'$

This algorithm is run by a combiner $Com \in A_j$. On the input of a ciphertext *C*, a dealer's (ID_d, Pk_d), partially decrypted shares $\delta_{1_j,j}, \ldots, \delta_{|A_j|_j,j}$, system

parameters *params* and public values for authorised subset $A_j \in \Gamma$, the algorithm outputs the corresponding value of the plaintext *M* or the failure symbol ⊥. The decryption algorithm does as follows:

$$\Delta = \Delta_1^{\frac{d_j}{d_j - 1}} \cdot \Delta_2^{\frac{-1}{d_j - 1}},\ \Delta_1 = C_4,\ \Delta_2 = v_j \prod_{u_{ij} \in A_j} \delta_{i_j, j} \tag{5}$$

$$\sigma = C_2 \oplus H_5(\Delta),\ M = C_3 \oplus H_6(\sigma),\ r = H_4(\sigma, M) \tag{6}$$

If $C_1 \neq r\left(H_2(ID_d, Pk_d)P + X_d\right)$, then an algorithm raises an error condition and exits with ⊥, otherwise sets the plaintext to *M*.

This completes the high-level description of CIBE-GAS scheme. A workflow of all algorithms is depicted in Fig. 1. Each user (say, *E*) runs **SetSecretValue** algorithm that generates a private/public key pair by taking system parameters as an input. Then the user *E* sends the registration request (1) to the Registration Authority (RA) and asks the latter to issue the certificate (**CertGen** algorithm, see step (3)). RA examines the *E*'s information (ID_E, Pk_E) and initiates some process to verify the identifying information provided by the user. When the registration request has been approved, the RA sends the confirmation (2) to the Trusted Authority (TA). The TA checks confirmation, and if everything is correct it generates the certificate which binds together *E*'s (ID_E, Pk_E) and other information. At the end of this phase the certificate is sent to *E*. Before or after receiving the message (4) from TA (it is omitted here) the user *E* should provide the proof of his knowledge of the relevant private key s_E (so called proof of possession[2]). Next steps of CIBE-GAS scheme presented in Fig.1 are fully compliant with Definition 3 and are omitted here.

3 Extension for Arbitrary Length Messages

Here we extend our CIBE-GAS scheme to deal with arbitrary length messages. A simple and an efficient way to build an encryption scheme that has an unrestricted message length is to build a hybrid one. Loosely speaking, such a scheme is based on the well-known KEM-DEM framework [13, 15] using the key encapsulation mechanism (KEM) and data encapsulation mechanism (DEM). The KEM uses a public key encryption technique to derive and encrypt a shared key, while DEM uses the shared key in a symmetric key algorithm to encrypt the arbitrarily long message.

[2] There are numerous methods that can be used to show proof of possession. In the simplest one, the certificate issued by TA can be encrypted using *E*'s public key. Only the holder of the private key is able to decrypt the certificate to use it.

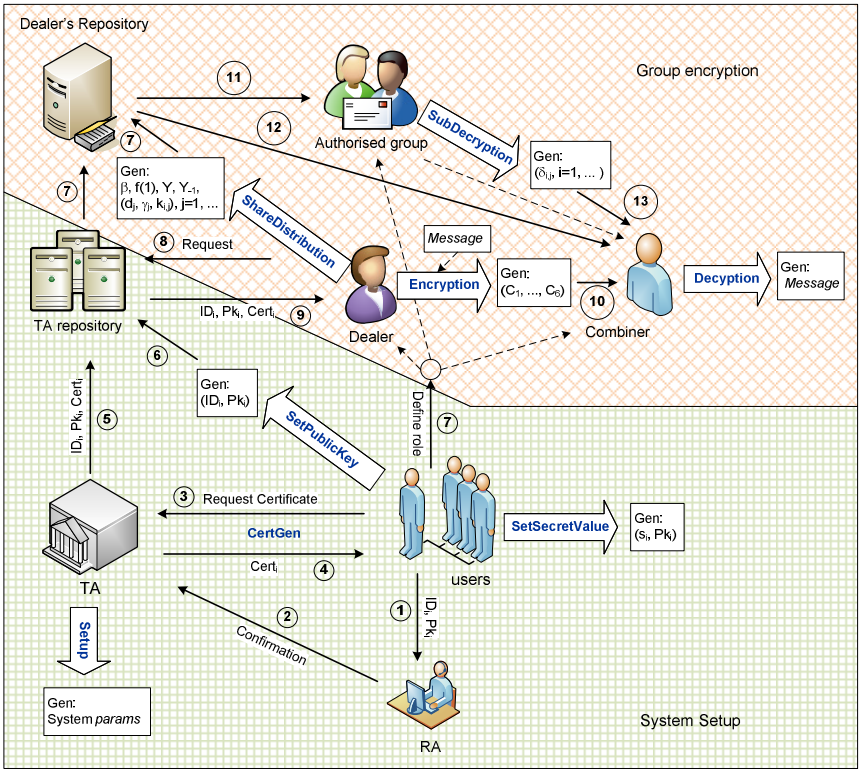

Fig. 1. Certificate and ID based group encryption scheme CIBE-GAS

Our extended encryption scheme is based on KEM-DEM framework given by L. Chen, *et al.* [12]. Following L. Chen, *et al.* formalisation of hybrid encryption, we assume that a hybrid construction *CIBE-GAS-H* = (*Setup-H*, *SetSecretValue-H*, *CertGen-H*, *SetPublicKey-H*, *ShareDistribution-H*, *Encryption-H*, *SubDecrytpion-H*, *Decryption-H*). For definitions of DEMs and their security definitions we refer to [12, 13, 15].

Definition 3 (*CIBE-GAS-H* scheme). For the same assumption as in Definition 2, the *CIBE-GAS-H* scheme consists of the following algorithms:

– **Setup-H** $(1^{\kappa}) \rightarrow (s, params)$
As in the previous CIBE-GAS scheme (see Definition 2). Additionally, the setup algorithm chooses a one-time symmetric key encryption scheme $SKE = (E_{SK}, D_{SK})$. The public parameters *params* are as follows:

$$params = \{G_1, G_2, \hat{e}, q, P, P_0, H_1, H_2, H_3, H_4, H_5, H_6, H_7, SKE\} \quad (7)$$

Comparing to CIBE-GAS scheme, this construction uses the new cryptographic hash function $H_7 : \{0,1\}^p \rightarrow \{0,1\}^{\kappa}$ for some $\kappa \in Z$; the κ parameter means the length of the resulting symmetric key $\mathcal{K}$.

- The algorithms **SetSecretValue-H**, **CertGen-H**, **ShareDistribution-H** and **SubDecryption-H** are the same as **SetSecretValue**, **CertGen**, **ShareDistribution** and **SubDecryption** respectively, as in the previous non-Hybrid CIBE-GAS scheme (see Definition 2).
- **Encryption-H** *(M, ID_d , Pk_d , F, pubVal, params)* $\rightarrow (c_K, c_M)$

  This operation is performed by the dealer. Given the dealer's identity ID_d , his public key Pk_d , the filter *F*, publicly known parameters *pubVal* and the system *params*, this algorithm outputs a ciphertext $C = (c_K, c_M)$, where c_K encapsulates the key *K* and c_M encapsulates the message *M*. We refer to such a construction as hybrid. The encryption algorithm with the key encapsulation mechanism (KEM) to calculate these values does as follows:

  (a) pick the random values $\sigma, m \in \{0,1\}^p$;
  (b) calculate encrypted key material c_K = **Encryption** *(m, ID_d , Pk_d , F, pubVal, params;* σ);
  (c) calculate session key $K = H_7(m)$;
  (d) calculate $c_M = E_{SK}(K, M)$;
  (e) output (c_K, c_M);
- **Decryption-H** $((c_K, c_M), ID_d , Pk_d , A_j ,(\delta_{I_{j,j}}, \ldots, \delta_{|A_j|_j,j}), params, pubVal) \rightarrow K'$

  This algorithm is run by the combiner once per encrypted values (c_K, c_M). Given (c_K, c_M), the dealer's (ID_d , Pk_d), partially decrypted shares $\delta_{I_{j,j}}, \ldots, \delta_{|A_j|_j,j}$, system parameters *params* and public values for authorised subset $A_j \in \Gamma$, the algorithm outputs the corresponding message *M* or the failure symbol $\perp$. The decryption algorithm works as follows:

  (a) calculate *m* = **Decryption** (c_K, ID_d , Pk_d , A_j , ($\delta_{I_{j,j}}, \ldots, \delta_{|A_j|_j,j}$), *params*, *pubVal*);
  (b) if (*m* == $\perp$), then return $\perp$; otherwise continue;
  (c) calculate session key $K = H_7(m)$;
  (d) decrypt message $M = D_{SK}(K, c_M)$;
  (e) return *M*.

4 CIBE-GAS-H System Architecture

In the following section we describe CIB-GAS-H architecture required to implement CIBE-GAS-H scheme (including original CIBE-GAS scheme). The component diagram shows basic system components and data flows between them. Next, we discuss security issues and describe messages exchanged between components.

4.1 Components

The architecture of CIBE-GAS system is based on SOA paradigm [16] and consists of five basic components. Fig. 2 presents a simplified component model (internal component structures and connection between interfaces are omitted to improve clarity of the figure). The most important component is CIBE-GAS library. This component consists of implemented algorithms from CIBE-GAS and CIBE-GAS-H schemes (ANSI C functions, based on PBC library [17]). This library component is directly used by other components (i.e. by Trusted Authority (TA), Dealer (DL) and Decrypter (DC) components) to perform cryptographic operations, which require to use CIBE-GAS-H scheme cryptographic operations.

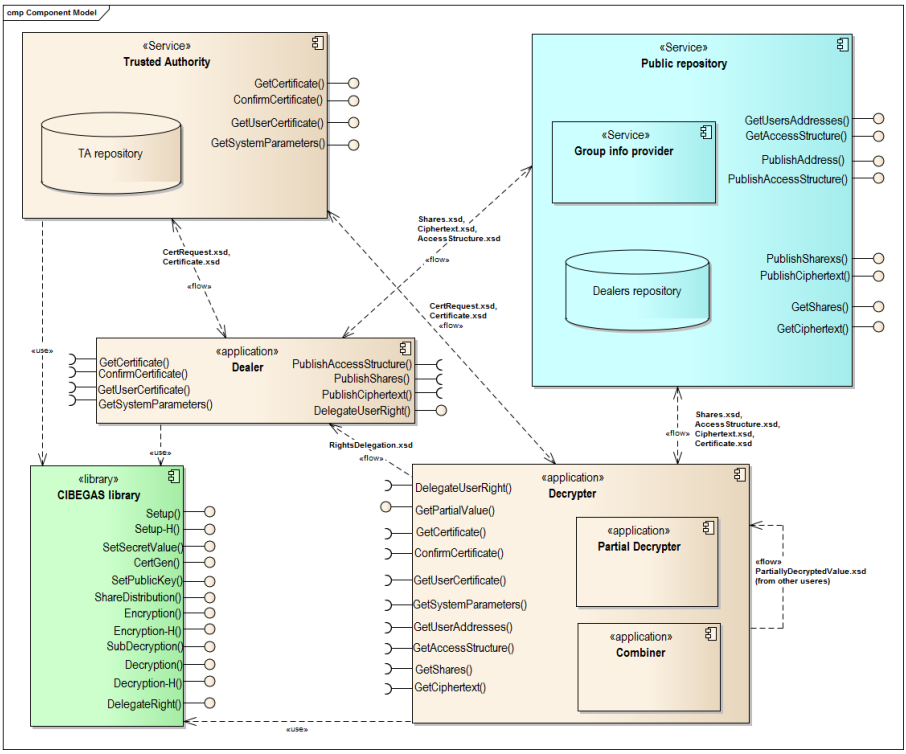

Fig. 2. Component model

The Trusted Authority (TA) component is responsible for management. It issues certificates and stores them in an internal trusted repository. Other TA interfaces provide user certificates and public system parameters. TA is an element of standard public key infrastructure. It means that TA should be considered subordinate to other TA and plays the role of intermediate certificate authority, which has a certificate issued by its predecessor, say CA_{prev}, in the trust model used. TA certificate issued by CA_{prev} contains system parameters (including TA's public key) and authorizes TA to issue certificates to end users.

Public Repository (PR) component stores information that can be kept in untrusted repository, i.e. if adversary gets any information from PR, he will not be able to decrypt any secret information. The PR stores public information required in the scheme (i.e. shares, access structures, ciphertext). Security analysis with description what can be made public is presented in further sections.

A Dealer component contains implementation of dealer specific functions from the scheme. The two most important functions are the shares and ciphertext creation. The dealer component gets all required information (e.g. user certificates and public system parameters) from TA (from its internal repository). The component sends ciphertext and shares to PR through appropriate interfaces. This component also can create and publish access structures. Additionally, it is possible that the dealer in on-line mode sends the shares or ciphertext directly to users (i.e. decrypting components). This variant of the architecture is not presented on the diagram.

A Decrypter is a component that contains all functions necessary to decrypt a document. It contains two major internal components. One is responsible for subdecryption process and the second for combining the partial decrypted values into a clear text. The decrypter component also handles requests from other decrypting components to subdecrypt the document.

4.2 Security Considerations

The approach to security issues presented in RFC 5408 [8] concerning IBE architecture is used to determine CIBE-GAS-H system security. The security of cryptographic algorithm comprising CIBE-GAS scheme against IND-CID-GO-CPA attacks was described in [1]. Due to this fact the following theorem can be formulated:

Theorem 1. The proposed CIBE-GAS-H encryption scheme is secure against adaptive chosen ciphertext attack IND-CID-GO-CCA2, assuming that (1) the hash function H_7 is modelled as random oracle and (2) the underlying CIBE-GAS scheme is an ID-OW-CPA secure encryption scheme.

The proof is similar to the proof of [13] and is omitted here.

We assume that the adversary is not able to access or change the cryptographic material of a TA. To achieve this goal the key material should be protected with FIPS 140-2 [18], Level 3 validated hardware that performs all key management, key storage, and key operations (such as digital signing and decryption) exclusively within hardware.

It is assumed also that authentication is done before communication between any two components (with some exceptions regarding Public Repository, described later). The authentication could be done using standard TLS protocol, for example.

The TA component belongs to some trusted zone of standard PKI. It is protected by physical, organizational and operational measures according to the best practices, e.g. [19]. Three basic reasons for binding TA with standard PKI are as follows (compare also [20]):

(a) resistance against Denial-of-Decryption (DoD) attack;
(b) solution of the public keys distribution problem for encryption schemes;
(c) PKI commercial maturity.

DoD attack is similar to Denial of Service (DoS). In DoD attack the adversary cannot gain any secret information, but any authorised user is also not able to decrypt this information and get the normal service. The adversary can succeed to launch this attack since there is no checking whether the public key is associated with the corresponding person or not.

The problem in distributing public keys for encryption schemes can be derived from DoD attack, and relies on the fact that the encrypter cannot correctly identify which public key to use from a range that are made available to him, knowing that choosing the wrong one will result in his message not getting through.

The certificate and ID-based public key cryptography is a new technology and at present that technology mainly exists in theory and is being tested in practice. This is in contrast to PKI-based cryptography, which has been an established and widespread technology for many years already. Therefore, the proper integration of a new encryption schemes with existing PKI architectures is required and should speed up their entering the market.

Public repository stores and provides encrypted documents and other public information required to decrypt documents. However, in CIBE-GAS-H (CIBE-GAS) scheme it is not possible to decrypt documents using only information from PR. Hence, it is not necessary to protect documents from unauthorized disclosure. The attack, which will result in modification of PR, will cause that the decryption of documents by authorized users might be impossible. Public repository does not require authentication, although authentication prevents unauthorized users (i.e. users which are not members of any group in the general access structure) from learning who is authorized to decrypt specific documents. Hence, further it is assumed that PR also requires authentication and provides information only to authorised users.

Dealer and decrypter components use secret keys internally. These keys must be protected from unauthorized disclosure. The components might use hardware components for key protection. The attacks compromising cryptographic keys or authentication between components will defeat the security of CIBE-GAS system.

During the design of CIBE-GAS system the following types of attacks where considered: passive monitoring of communication channels, masquerade as TA and denial of service. All message exchanges between components are protected by TLS protocol. CIBE-GAS system relies on TLS security mechanism established to prevent masquerade and passive monitoring (like in the architecture proposed in [21]). As the protection against DDoS attacks is generally very difficult, CIBE-GAS system relies on firewall, IDS or other network security mechanisms to protect against these kinds of attacks.

4.3 Messages

The system architecture is service oriented and XML messages are exchanged between components using SOAP protocol. These messages are designed for standalone CIBE-GAS system. If CIBE-GAS system is integrated with other system, then messages can be encapsulated into existing formats, e.g. XML Encryption [22].

Names of messages' XML schemas exchanged between components are presented on Fig.2. There are 8 basic definitions of xml messages (*CertRequest, Certificate, RightsDelegation, Ciphertext, Shares, AccessStructure, Users, PartiallyDecrypted-Value*).

Fig.3 contains content model view of XML schemas of *Ciphertext*, *Shares* and *AccessStructure* messages, which are necessary to decrypt a document. The *Ciphertext* message consists of cipher values c_K consisting of C_1, C_2, C_3, C_5, C_6 and C_4 for each privileged set *A*, which contains an encrypted symmetric key material and encrypted message c_M . *Shares* messages contain public share parameters. *AccessStructure* defines *A* sets. The *Ciphertext* message also contains *SharesID* and *AccessStrutureID* attributes which are references to related share parameters and access structure, respectively.

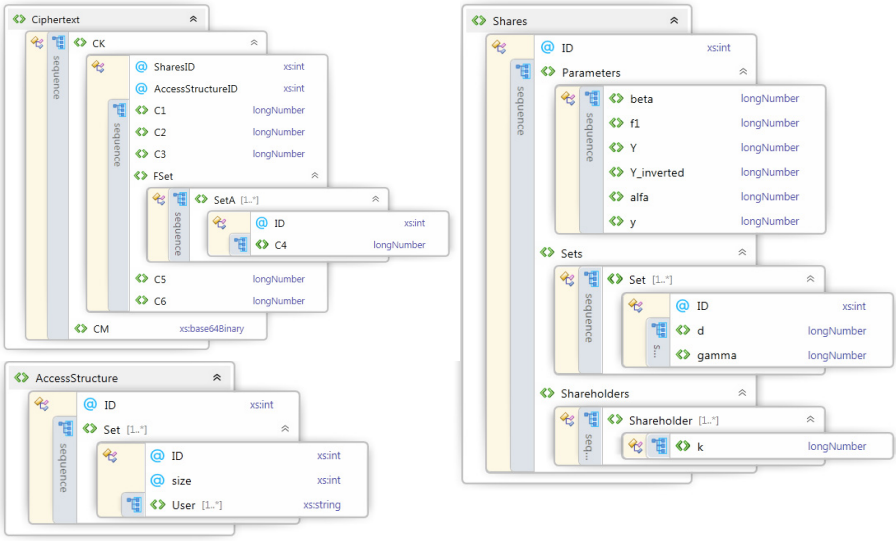

Fig. 3. Content model view of chosen messages

After downloading *Ciphertext* user downloads *Shares* and *AccessStructure*. Next step is determination of *A* set (or sets) to which the user belongs and for which *Ciphertext* contains corresponding C_4 value. When the user decides which *A* set he will use to decrypt document (playing combiner role), he sends to PR request in the message *UserRequest* and receives the message *Users* with IP addresses of other users from the chosen *A* Set. IP addresses are used to get from other users their partially decrypted values in on-line mode. These values are send using *PartiallyDecryptedValue.*

The *CertRequest* message (user name, public key(*X, Y*)) is used by users to request *Certificate* from TA. The *Certificate* message encapsulates user certificate in X.509 format (see section 4.4). Public system parameters (pairing parameters, hash functions, TA public values *P* and P_0) required to perform every operation using CIBEGAS library component are encapsulated inside user certificate. *RightsDelegation*

message is a special message send by a user to a dealer when the user wants to delegate his right to decrypt a document to another user.

4.4 Public Key Certificate in X.509 Format

Here we omit the contents of TA's certificate and show only how to extend the semantics of an end entity's X.509 certificate to apply CIBE-GAS-H scheme in practice. The CIBE-GAS-H scheme is different from the conventional public key algorithm and it means that public key and its algorithm identifier should be explicitly included in the certificate.

Fig. 4. X.509 certificate structure

The Fig.4 presents the syntax of X.509 certificate compliant with ASN.1. The CIBGAS entity's public key should be included in the *subjectPublicKeyInfo* basic field of the certificate. This field has two subfields: *algorithm* and *subjectPublicKey*. The value for *algorithm* field can be the public key algorithm identifier with its parameters included into *CIBEGASSysParams* structure. The value for *subjectPublicKey* is bit string of DER encoding of public key given in *CIBEGASPublicKey* structure.

The *signatureValue* field contains a digital signature computed upon the ASN.1 DER encoded *tbsCertificate*. For CIBE-GAS scheme, the value of this field is the Boneh *et al.*'s short signature [20, 23] of the certificate information of the *tbsCertificate* field.

5 Conclusions

The proposed CIBE-GAS-H system is designed to work with messages of arbitrary length. These messages can be encrypted using the rules contained in access structures. In practice, it is possible using access structures to describe any access rules (e.g. hierarchical or threshold ones). The proposed architecture visualises the properties of the system. The most important properties are security related. The system is divided into components based on security analysis. Only the successful attack on the security of TA will compromise the complete system. The architecture is also designed to be scalable. Users can be added and removed dynamically and the length of a message is practically not restricted.

In the paper, we show only how to combine the two level CIBE-GAS-H certification domain with traditional PKI. However, hierarchical identity based cryptography HIBC constructions (see [24]) can be used to extend the scheme and propagate down the trust to identities that are not registered at the trusted node (TA level).

Future work will be carried out simultaneously in two directions. In the first the risk analysis method will be applied to CIBE-GAS-H system [25, 26] and in the second one our method for long-term preservation of documents digital signatures [27] will be enhanced to support confidentiality using CIBE-GAS-H technique.

Acknowledgment. This scientific research work is supported by NCBiR of Poland (grant No PBS1/B3/11/2012) in 2012-2015.

References

1. Hyla, T., Pejaś, J.: Certificate-Based Encryption Scheme with General Access Structure. In: Cortesi, A., Chaki, N., Saeed, K., Wierzchoń, S. (eds.) CISIM 2012. LNCS, vol. 7564, pp. 41–55. Springer, Heidelberg (2012)
2. Sang, Y., Zeng, J., Li, Z., You, L.: A Secret Sharing Scheme with General Access Structures and its Applications. International Journal of Advancements in Computing Technology 3(4), 121–128 (2011)
3. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
4. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
5. IEEE-P1363.3. IEEE P1636.3TM/D1 draft standard for identity-based public-key cryptography using pairings (2008)
6. RFC 6508 Sakai-Kasahara Key Encryption, SAKKE (2012)
7. RFC 5091 Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems (2007)
8. RFC 5408 Identity-Based Encryption Architecture and Supporting Data Structures (January 2009)
9. Chatterjee, S., Sarkar, P.: Identity-Based Encryption. Springer, New York (2011)
10. Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Applied Mathematics 156(16), 3113–3121 (2008)

11. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
12. Chen, L., Cheng, Z., Malone–Lee, J., Smart, N.P.: Efficient ID-KEM based on the Sakai–Kasahara key construction. IEE Proceedings, Information Security 153(1), 19–26 (2006)
13. Bentahar, K., Farshim, P., Malone-Lee, J., Smart, N.P.: Generic constructions of identity-based and certificateless KEMs. Journal of Cryptology 21, 178–199 (2008)
14. Burak, D., Chudzik, M.: Parallelization of the Discrete Chaotic Block Encryption Algorithm. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Waśniewski, J. (eds.) PPAM 2011, Part II. LNCS, vol. 7204, pp. 323–332. Springer, Heidelberg (2012)
15. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33, 167–226 (2004)
16. Bell, M.: Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture. Wiley & Sons (2008) ISBN 978-0-470-14111-3
17. Lynn, B.: PBC Library Specification, `http://crypto.stanford.edu/pbc/` (retrieved 2013)
18. FIPS PUB 140-2: Security Requirements for Cryptographic Modules, National Institute of Standards and Technology (2001)
19. Souppaya, M., Wack, J., Kent, K.: Security Configuration Checklist Program for IT Products - Guidance for Checklist Users and Developers, NIST Special Publication SP 800-70 (May 2005)
20. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
21. Appenzeller, G., et al.: Identity-Based Encryption Architecture and Supporting Data Structures, RFC5408, IETF (2009)
22. Imamura, T., et al.: XML Encryption Syntax and Processing. W3C Recommendation (2002)
23. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 614–629. Springer, Heidelberg (2003)
24. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
25. El Fray, I., Kurkowski, M., Pejaś, J., Maćków, W.: A New Mathematical Model for Analytical Risk Assessment and Prediction in IT Systems. Control and Cybernetics 41(1), 241–268 (2012)
26. El Fray, I.: A Comparative Study of Risk Assessment Methods, MEHARI & CRAMM with a New Formal Model of Risk Assessment (FoMRA) in Information Systems. In: Cortesi, A., Chaki, N., Saeed, K., Wierzchoń, S. (eds.) CISIM 2012. LNCS, vol. 7564, pp. 428–442. Springer, Heidelberg (2012)
27. Hyla, T., El Fray, I., Pejaś, J., Maćków, W.: Long-term Preservation of Digital Signatures for Multiple Groups of Related Documents. IET Information Security 6(3), 219–227 (2012)