Content-Based Encryption

Xiaofen Wang[1,2](✉) and Yi Mu[2](✉)

[1] Center for Cyber Security and Big Data Research Center,
University of Electronic Science and Technology of China,
Chengdu 611731, Si Chuan, China
xfwang@uestc.edu.cn

[2] Centre for Computer and Information Security Research,
School of Computing and Information Technology, University of Wollongong,
Wollongong 2500, Australia
ymu@uow.edu.au

Abstract. Content-centric networks have demonstrated an entirely new type of network topology, which offers a new way to distribute information in the data-driven network. Unlike the TCP/IP network topology, which is address-driven, content-centric networks do not require any address. Based on the content-to-consumer paradigm, content-centric networking architecture was proposed for the content to be provided efficiently with great convenience to users. As the content-centric network is not address-driven, when a data packet is delivered it cannot be encrypted with any encryption key of a node. Therefore, data confidentiality in content-centric network is a challenging problem. Motivated to solve this problem, we introduce a new cryptosystem for content-based encryption, where the encryption key is associated with the content. We propose a content-based encryption scheme (CBE), which is proven to be semantically secure in the random oracle model. We apply the CBE to construct a secure content delivery protocol in a content-centric network.

Keywords: Content-centric network · Content-based encryption · Chosen plain-text security

1 Introduction

In the traditional TCP/IP network, which is address-centric, the data packets need to tell *where* the content is. Therefore, the IP packets contain two addresses, one for the source and the other for the destination host. All the traffic on the Internet rely on these IP addresses. To address the security of TCP/IP network, conventional cryptography can be applied. In case of public-key cryptography, each host is usually equipped with a pair of public and private keys. In traditional public key infrastructure (PKI), the public key of a host is accompanied with a certificate. To simplify certificate management in traditional PKI, identity-based infrastructure [1,4] can also be applied in a TCP/IP network, where the public key of a host can be its IP address. The problem for the TCP/IP networking is

J.K. Liu and R. Steinfeld (Eds.): ACISP 2016, Part II, LNCS 9723, pp. 57–72, 2016.
DOI: 10.1007/978-3-319-40367-0_4

that it assumes there is end-to-end physical connectivity. However, end-to-end connectivity may not ever exist and links (contacts) may not be suitable for schedules. Therefore, if the target provider is unreachable or unable to provide the requested content, then the content acquisition in TCP/IP network will fail.

When users acquire a content, which could be a file, a music, a video, etc., in a network, they concern *what* they receive, where the location of the content might not be important. To replace *where* with *what* and to overcome the inherent problem in the TCP/IP network, the content-to-consumer paradigm was presented to replace the host-to-host paradigm. Therefore, Content-centric Networking [8,9,13] or Information-centric Networking [2,11], a new communication architecture built on named data, was introduced. Content-centric network has no notion of host at its lowest level, while a packet "address" names content (not location). In a content-centric network, the content is delivered to the intended consumers regardless of their addresses [8,9,11,13]. Therefore, it offers great advantage for content acquisition, as in the content-centric network, the content-centric mechanism is employed to seek the target content, and any node which holds the requested contents can provide contents. This is a distinct feature compared with the TCP/IP network, since in the TCP/IP network, even if an intermediate node between the source node and the destination node possesses the requested content, it cannot provide the content because only the target provider node can provide it [10]. Therefore, the content acquisition cost and latency might be increased. In the content-centric network, the consumer node can acquire the content in an optimal manner. The content can be provided by a nearest node instead of a further node if both nodes possess the content. Therefore, the content-centric network can greatly reduce the cost of content transmission.

As the content-centric network is not address-driven, in a public-key setting, a content cannot be encrypted with the destination node's public key for the confidentiality of the content. Therefore, different from the TCP/IP networking, the traditional public key infrastructure cannot be used to best suit content-centric networking. The ID-based encryption [3,12] is unsuitable for the content-centric network either. In 1984, Shamir [12] asked for a public key encryption scheme in which the public key can be an arbitrary string. Their motivation was to simplify certificate management in traditional public key crypto-systems. Boneh and Franklin [3] proposed the first practical and provably secure ID-based encryption scheme. In an ID-based encryption, the user's identity is used as the public key, which is usually the IP address in a network protocol, and the corresponding private key is extracted from the identity. As in this content-to-consumer paradigm, there is no notion of host, which means no "IP address" is used, the ID-based encryption cannot be used for the content-centric network when a provider node encrypts the content. The conventional symmetric encryption is neither a good choice to provide the content's confidentiality in a content-centric network, since the content provider and the content consumer need to share the same symmetric key. The obvious issue is key distribution, which requires users' addresses.

In this paper, we propose a new notion of *content-based encryption*, where the encryption key is directly associated with the content itself, and the corresponding private decryption keys, generated by a trusted party, are provided for the valid users who are potential content receivers. Any user who wants to acquire the content needs to obtain one of the associated private keys of the content. With the content-based encryption key, the content provider can encrypt the content. The ciphertext is then relayed by intermediate nodes to the corresponding consumer who acquires the content. The consumer can decrypt it with his private decryption key.

To better illustrate the applicability of our scheme to the content-centric encryption, in the paper, we also construct a secure content delivery protocol tailored for the content-centric network. We describe how a content can be delivered by the content provider and acquired by the consumer and how the confidentiality of the content is achieved.

Besides the content-centric network, the content-based encryption can be used in many other content sharing applications, e.g. secure multimedia content dispatching and selling. The content owner encrypts the content under the content-based public key. The consumer who has got one of the corresponding private keys can decrypt it and retrieve the content.

As a note, we noticed that Zhao and Zhuo [14] proposed a content-based encryption scheme for wireless H.264 compressed videos. However it is not relevant to our notion of content-based encryption.

Our Contribution. We propose a new notion of *content-based encryption* for the content-centric network. In this new encryption paradigm, the public encryption key is directly associated with the content name itself and the private keys of the content are derived secretly from the content name. The content encrypted with the public key can be decrypted by any user who holds a valid content-based private key. We present a concrete content-based encryption scheme and prove its semantic security under the random oracle model. Significantly, we are able to show the application of the proposed content-based encryption scheme when the content is delivered in the content-centric network.

Organization. We provide the definitions of content-based encryption and its security notion in Sect. 2. In Sect. 3, we introduce the preliminaries and the hard problem assumption. We then present our first construction CBE and its security proof in Sect. 4. In Sect. 5, we present an application of our scheme to show how it works in the content-centric network. We conclude this paper in Sect. 6.

2 Definitions

A content-based encryption scheme $\mathcal{E}$ is specified by four algorithms, namely **Setup**, **Encrypt**, **Key-Extract**, and **Decrypt**:

Setup(1^λ): it takes as input the security parameter λ and returns the system parameters params and master-key MK. params are publicly known, while MK is only known to the Private Key Generator (PKG).

Encrypt$(\mathsf{params}, \mathsf{C}, \mathsf{name})$: it is a randomized algorithm that takes as input the public parameters params, a content C and the unique name of the content name and outputs the ciphertext CT. Each content has a unique content name.

Key-Extract$(\mathsf{params}, \mathsf{MK}, \mathsf{name})$: it is a randomized algorithm that takes as input params, master-key MK, the unique name of the content C and outputs a set of private keys SK_i, $i = 1, \ldots, \bar{n}$, for an integer $\bar{n}$.

Decrypt$(\mathsf{params}, CT, SK_i)$: it takes as input a ciphertext CT, a private key SK_i, and the public parameters params and outputs the content C.

In the following, we slightly modify the definition of semantic security (IND-CPA) for a public key encryption scheme [7] and define a new semantic security model in content-based encryption where the adversary can obtain the decryption key associated with any content wrt content name name_j of her choice (other than the content name name being attacked).

We say that a content-based encryption scheme $\mathcal{E}$ is semantic secure against an adaptive chosen plaintext attack ($\mathsf{IND\text{-}name\text{-}CPA}$) if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage against the challenger in the following $\mathsf{IND\text{-}name\text{-}CPA}$ game.

Setup: the challenger takes a security parameter λ as input and runs the **Setup** algorithm. It gives the adversary the resulting system parameters params and keeps the master-key MK to itself.

Phase 1: $\mathcal{A}$ adaptively issues queries $q_1, \ldots, q_m$ where query q_i is one of:

Key Extraction queries $\langle \mathsf{name}_i \rangle$. The challenger responds by running algorithm **Key-Extract** to generate one private decryption key SK_i corresponding to the public key name_i. It sends SK_i to the adversary $\mathcal{A}$.

Challenge: once Phase 1 is over, it outputs two equal length contents $\mathsf{C}_0^*, \mathsf{C}_1^*$ on which it wishes to be challenged. The only constraint is that the adversary did not make any key extraction query of their corresponding content names name_0^* or name_1^* in Phase 1. The challenger picks a random bit $b \in \{0, 1\}$ and sets $CT^* = \mathsf{Encrypt}(\mathsf{params}, \mathsf{C}_b^*, \mathsf{name}_b^*)$. It sends the challenge ciphertext CT^* to the adversary $\mathcal{A}$.

Phase 2: $\mathcal{A}$ adaptively issues queries $q_{m+1}, \ldots, q_t$ key extraction queries as in Phase 1. The restriction is that the adversary cannot make any key extraction query for name_b^* $(b = 0, 1)$.

Guess: finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins the game if $b' = b$.

We refer to such an adversary $\mathcal{A}$ as an $\mathsf{IND\text{-}name\text{-}CPA}$ adversary. We define adversary $\mathcal{A}$'s advantage in attacking the scheme $\mathcal{E}$ as the following function of the security parameter λ: $\mathsf{Adv}_{\mathcal{E},\mathcal{A}}(\lambda) = \left|\Pr[b' = b] - \frac{1}{2}\right|$.

Definition 1. *A content-based encryption system $\mathcal{E}$ is semantically secure against an adaptive chosen plaintext attack if for any polynomial time* $\mathsf{IND\text{-}name\text{-}CPA}$ *adversary $\mathcal{A}$ the function* $\mathsf{Adv}_{\mathcal{E},\mathcal{A}}(\lambda)$ *is negligible. As shorthand, we say that $\mathcal{E}$ is* $\mathsf{IND\text{-}name\text{-}CPA}$ *secure.*

3 Preliminaries

3.1 Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of large prime order p. $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear map which satisfy the following properties:

- Bilinear. For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p^*$, we have $e(u^a, v^b) = e(u, v)^{ab}$;
- Non-degenerate. $e(g, g) \neq 1$, if g is a generator of $\mathbb{G}$;
- Computable. For any $u, v \in \mathbb{G}$, $e(u, v)$ can be computed efficiently.

3.2 Complexity Assumptions

The security of our encryption system is based on the truncated decision augmented bilinear Diffie-Hellman exponent assumption (truncated decision ABDHE) [6]. The truncated decision n-ABDHE problem is defined as follows.

Let n be an integer and $(p, \mathbb{G}, \mathbb{G}_T, e)$ be a bilinear map group system. Let g, g' be the generators of $\mathbb{G}$. For some unknown $a \in \mathbb{Z}_p^*$, given a vector of $n + 3$ elements $(g', g'^{(a^{n+2})}, g, g^a, g^{(a^2)}, \ldots, g^{(a^n)}) \in \mathbb{G}^{n+3}$ and an element $Z \in \mathbb{G}_T$ as input, decide whether $Z = e(g', g)^{(a^{n+1})}$ or not.

We define an algorithm $\mathcal{B}$ that outputs $b \in \{0, 1\}$ has advantage ε in solving truncated decision n-ABDHE problem if

$$\Big|\Pr\Big[\mathcal{B}(g', g'^{(a^{n+2})}, g, g^a, g^{(a^2)}, \ldots, g^{(a^n)}, e(g', g)^{(a^{n+1})}) = 0\Big] \\ - \Pr\Big[\mathcal{B}(g', g'^{(a^{n+2})}, g, g^a, g^{(a^2)}, \ldots, g^{(a^n)}, Z) = 0\Big]\Big| \geq \varepsilon$$

where the probability is over the random choice of generators g, g' in $\mathbb{G}$, the random choice of a in $\mathbb{Z}_p^*$ and the random choice of Z in $\mathbb{G}_T$.

Definition 2. *We say that the truncated decision (t, ε, n)-*ABDHE *assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ε in solving the truncated decision n-*ABDHE *problem in $\mathbb{G}$.*

4 Construction for Chosen Plaintext Security

We propose a content-based encryption system CBE that is secure against the chosen plaintext attack. In the construction, we assume each content denoted by C is associated with a unique identifier denoted by name. The public encryption key of each content is name, and its private decryption keys are generated based on its name. For each content, there is a unique encryption key, but multiple private decryption keys.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime order p, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be the bilinear map. The content-based encryption system CBE works as follows.

Setup. The PKG picks random generators $g, h, y \in \mathbb{G}$ and a random number $\alpha \in \mathbb{Z}_p^*$. It sets $g_1 = g^\alpha \in \mathbb{G}$. It also chooses two collision-resistant hash functions $H : \mathbb{G} \rightarrow \mathbb{Z}_p^*$ and $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$. The public parameters params and the master secret key MK are given by $\mathsf{params} = (g, g_1, h, y, H, H_1), \quad \mathsf{MK} = \alpha$.

Encrypt. To encrypt the content $\mathsf{C} \in \mathbb{G}_T$ using its unique identifier $\mathsf{name} \in \{0,1\}^*$, the sender generates a random number $z \in \mathbb{Z}_p^*$ and computes the ciphertext CT as follows: $U = (g_1 g^{-H_1(\mathsf{name})})^z$, $V = y^{-z}, W = e(g,g)^z$, $T = \mathsf{C} \cdot e(g,h)^{-z}$. The sender sends the ciphertext $CT = (U, V, W, T)$ to the users.

Key-Extract. For $i = 1, 2, \ldots, \bar{n}$, the PKG generates the secret key SK_i for a content C with the identifier name. The PKG generates a random number $r_i \in \mathbb{Z}_p^*$, and computes $R_i = g^{r_i}, \quad t_i = H(R_i), \quad S_i = (hy^{r_i} g^{-t_i})^{\frac{1}{\alpha - H_1(\mathsf{name})}}$. For $i = 1, 2, \ldots, \bar{n}$, the PKG outputs the private decryption key $SK_i = (R_i, S_i)$, and sends it to the user U_i.

Decrypt. To decrypt the ciphertext $CT = (U, V, W, T)$, the user U_i who holds the decryption key $SK_i = (R_i, S_i)$, firstly computes $t_i = H(R_i)$ and decrypts the ciphertext to obtain the content: $\mathsf{C} = T \cdot e(U, S_i) \cdot e(V, R_i) \cdot W^{t_i}$.

Correctness. Assuming the ciphertext is well-formed for name:

$$\begin{aligned} & e(U, S_i) \cdot e(V, R_i) \cdot W^{t_i} \\ &= e(g^{z(\alpha - H_1(\mathsf{name}))}, (hy^{r_i} g^{-t_i})^{\frac{1}{\alpha - H_1(\mathsf{name})}}) \cdot e(y^{-z}, g^{r_i}) \cdot e(g,g)^{zt_i} \\ &= e(g,h)^z \cdot e(g,y)^{zr_i} \cdot e(g,g)^{-zt_i} \cdot e(g,y)^{-zr_i} \cdot e(g,g)^{zt_i} = e(g,h)^z, \end{aligned}$$

as required. Therefore, the content C can be recovered.

Remark. In our construction, the PKG generates multiple different secret keys corresponding to each content. These secret keys are securely distributed to multiple users (at registration, for example). An authorized user who holds a private decryption key can recover the content. In CBE, without the knowledge of the master key MK, the authorized users cannot collude to generate a new valid secret key of the same content.

CBE is proved IND-name-CPA secure under the truncated decision n-ABDHE assumption.

Theorem 1. *Assume the truncated decision (t, ε, n)-ABDHE assumption holds for $(\mathbb{G}, \mathbb{G}_T, e)$. The proposed CBE scheme is (t', ε', q_n) IND-name-CPA secure where $q_n = n - 1$, $t' = t - \mathcal{O}(t_{H_1} \cdot n^2) - \mathcal{O}(t_H \cdot n) - \mathcal{O}(t_{exp} \cdot n^2)$, $\varepsilon' = \varepsilon + \frac{1}{p}$, t_{H_1} is the time required to compute the hash H_1, t_H is the time required to compute the hash H, and t_{exp} is the time required to compute the exponentiation in $\mathbb{G}$.*

Proof. Assume that $\mathcal{A}$ is an adversary that (t', ε', q_n)-breaks the $\mathsf{IND\text{-}name\text{-}CPA}$ security of CBE above. We can then construct an algorithm, $\mathcal{B}$, that solves the truncated decision n-ABDHE problem, as follows. $\mathcal{B}$ takes as input a random truncated decision n-ABDHE challenge $(g', g'^{a^{n+2}}, g, g^a, \dots, g^{a^n}, Z)$, where Z is either $e(g, g')^{a^{n+1}}$ or a random element of $\mathbb{G}_T$. $\mathcal{B}$ works as a challenger in the following procedure.

Setup. $\mathcal{B}$ generates a random polynomial $f(x) \in \mathbb{Z}_p[x]$ of degree n. It also randomly chooses $c, x^* \in \mathbb{Z}_p^*$. It sets $h = g^{f(a)}$ by computing from $g, g^a, \dots, g^{a^n}$. $\mathcal{B}$ sets $g_1 = g^a$ and $y = g_1^c g^{-cx^*} = g^{c(a-x^*)}$. It sends the public key (g, g_1, h, y) to the adversary $\mathcal{A}$. Since g, a, c and $f(x)$ are uniformly chosen at random, h and y are uniformly random, and the public key has a distribution identical to that in the actual attack.

Hash Query. $\mathcal{B}$ can make hash queries of H_1 and H, and maintains two hash lists L_1 and L_2 correspondingly.

H_1-query: $\mathcal{B}$ maintains a list L_1 of a tuple (name_i, x_i). The list is initially empty. Upon receiving a hash query for name_i, $\mathcal{B}$ looks up the list L_1 to find the hash value x_i of name_i and returns x_i to $\mathcal{A}$. If name_i is not on the list L_1, $\mathcal{B}$ randomly chooses $x_i \in \mathbb{Z}_p^*$ and adds a new tuple (name_i, x_i) to L_1. Then $\mathcal{B}$ returns x_i.

H-query: $\mathcal{B}$ maintains a list L_2 of a tuple (r_i, R_i, t_i). The list is initially empty. Upon receiving a hash query for $R_i = g^{r_i}$, $\mathcal{B}$ looks up the list L_2 to find the hash value t_i of R_i and returns t_i to $\mathcal{A}$. If (r_i, R_i) is not on the list L_2, $\mathcal{B}$ randomly chooses $t_i \in \mathbb{Z}_p^*$ and adds a new tuple (r_i, R_i, t_i) to L_2. Then $\mathcal{B}$ returns t_i.

Phase 1. $\mathcal{A}$ makes key extraction queries. $\mathcal{B}$ responds to a key extraction query for name_i as follows. Firstly $\mathcal{B}$ looks up L_1 to find a corresponding x_i. If $x_i = a$, $\mathcal{B}$ uses a to directly solve the truncated decision n-ABDHE problem. Otherwise, $\mathcal{B}$ randomly chooses $r_i \in \mathbb{Z}_p^*$ and computes $R_i = g^{r_i}$. It makes an H-query to obtain $H(R_i) = t_i$. Then $\mathcal{B}$ sets $S_i = g^{\frac{f(a)+acr_i-x^*cr_i-t_i}{a-x_i}}$ by computing from $g, g^a, \dots, g^{a^{n-1}}$. $\mathcal{B}$ sets the private decryption key for name_i as (R_i, S_i). This is a valid secret key for name_i, since $S_i = g^{\frac{f(a)+acr_i-x^*cr_i-t_i}{a-x_i}} = (hy^{r_i}g^{-t_i})^{\frac{1}{a-H(\mathsf{name}_i)}}$, as required.

Challenge. $\mathcal{A}$ outputs two equal length contents $\mathsf{C}_0^*, \mathsf{C}_1^* \in \mathbb{G}_T$ with unique identifiers name_0^* and name_1^* correspondingly. If $x^* = a$, $\mathcal{B}$ uses a to solve the truncated decision n-ABDHE problem directly. Otherwise, $\mathcal{B}$ generates a bit $b \in \{0, 1\}$, and computes a secret key $(R_b = g^{r^*}, S_b = (hy^{r^*}g^{-t^*})^{\frac{1}{a-x^*}})$ for name_b^* as in Phase 1. Let $f_2(x) = x^{n+2}$ and let $F_2(x) = \frac{f_2(x)-f_2(x^*)}{x-x^*}$, which is a polynomial of degree $n+1$. $\mathcal{B}$ sets $U^* = g'^{f_2(a)-f_2(x^*)}$, $V^* = g'^{-c(f_2(a)-f_2(x^*))}$, $W^* = Z \cdot e(g', \prod_{i=0}^{n} g^{F_{2,i}a^i})$, $T^* = \frac{\mathsf{C}_b^*}{e(U^*, S_b)e(V^*, R_b)W^{t^*}}$, where $t^* = H(R_b)$ and $F_{2,i}$ is the coefficient of x^i in $F_2(x)$. It returns $CT^* = (U^*, V^*, W^*, T^*)$ to $\mathcal{A}$ as the challenge ciphertext.

Let $s = (\log g^{g'})^{F_2(a)}$. If $Z = e(g', g)^{a^{n+1}}$, then $U^* = g^{s(a-x^*)} = (g_1 g^{-H(\mathsf{name}_b^*)})^s$, $V^* = y^{-s}$, $W^* = e(g, g)^s$ and $\mathsf{C}_b / T^* = e(U^*, S_b)e(V^*, R_b)$

$W^{t^*} = e(g,h)^s$ under randomness s. Since $\log g^{g'}$ is uniformly random, s is uniformly random. Therefore, (U^*, V^*, W^*, T^*) is a valid, appropriately distributed ciphertext to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ makes key extraction queries. $\mathcal{B}$ responds as in Phase 1.

Guess. Finally, $\mathcal{A}$ outputs its guess b'. If $b' = b$, $\mathcal{B}$ outputs 0; otherwise, it outputs 1.

Perfect Simulation. When $Z = e(g^{(a^{n+1})}, g')$, the public key and challenge ciphertext issued by $\mathcal{B}$ come from a distribution identical to that in the actual construction. Now we will show that the secret keys issued by $\mathcal{B}$ are appropriately distributed. Let $\mathcal{I}$ be a set consisting of a, the hash value $H(\mathsf{name}_b^*)$, and the hash value $H(\mathsf{name}_i)$ queried by $\mathcal{A}$; observe that $|\mathcal{I}| \leq n+1$. As $f(x)$ is a uniformly random polynomial of degree n, from $\mathcal{A}$'s view, the values $\{f(a_i) : a_i \in \mathcal{I}\}$ are uniformly random and independent. Therefore, the keys issued by $\mathcal{B}$ are appropriately distributed.

Probability Analysis. If $Z = e(g^{a^{n+1}}, g')$, then the simulation is perfect, and $\mathcal{A}$ will guess the bit b correctly with probability $\frac{1}{2} + \varepsilon'$. Otherwise, Z is uniformly random, thus the elements (U^*, V^*, W^*) are uniformly random and independently distributed in $\mathbb{G} \times \mathbb{G} \times \mathbb{G}_T$. In this case, the inequality $W^* \neq e(U^*, g)^{\frac{1}{a-x^*}}$ holds with probability $1 - \frac{1}{p}$. Since r^* is uniformly random and independent from $\mathcal{A}$'s view, t^* is random and independent. When the inequality $W^* \neq e(U^*, g)^{\frac{1}{a-x^*}}$ holds, the value of

$$e(U^*, S_b)e(V^*, R_b){W^*}^{t^*} = e(U^*, (hy^{r^*} g^{-t^*})^{\frac{1}{a-x^*}})e(V^*, g^{r^*}){W^*}^{t^*}$$

$$= e(U^*, h^{\frac{1}{a-x^*}})e(U^*, y^{\frac{1}{a-x^*}})^{r^*} e(V^*, g^{r^*})(W^*/e(U^*, g)^{\frac{1}{a-x^*}})^{t^*}$$

is uniformly random and independent from $\mathcal{A}$'s view. Therefore,

$$T^* = \frac{\mathsf{C}_b^*}{e(U^*, S_b)e(V^*, R_b){W^*}^{t^*}}$$

is uniformly random and independent, and (U^*, V^*, W^*, T^*) can impart no information regarding the bit b.

Assume that no $H_1(\mathsf{name}_i)$ equals a (which would only increase $\mathcal{B}$'s success probability). If Z is randomly sampled from $\mathbb{G}_T$,

$$\left| \Pr[\mathcal{B}(g', g'^{(a^{n+2})}, g, g^a, g^{(a^2)}, \ldots, g^{(a^n)}, Z) = 0] - \frac{1}{2} \right| \leq \frac{1}{p}.$$

When $Z = e(g^{a^{n+1}}, g')$,

$$\left| \Pr[\mathcal{B}(g', g'^{(a^{n+2})}, g, g^a, g^{(a^2)}, \ldots, g^{(a^n)}, Z) = 0] - \frac{1}{2} \right| \geq \varepsilon'.$$

Thus, for uniformly random g, g', a and Z, we have

$$\Big|\Pr[\mathcal{B}(g', g'^{(a^{n+2})}, g, g^a, g^{(a^2)}, \ldots, g^{(a^n)}, e(g',g)^{(a^{n+1})}) = 0] \\ - \Pr[\mathcal{B}(g', g'^{(a^{n+2})}, g, g^a, g^{(a^2)}, \ldots, g^{(a^n)}, Z) = 0]\Big| \geq \varepsilon' - \frac{1}{p}.$$

Time-Complexity. In the simulation, to respond $\mathcal{A}$'s key extraction queries for name_i, $\mathcal{B}$ needs to make n H_1-hash query, 1 H-hash query and to compute $g^{\frac{f(a)+acr_i-x^*cr_i-t_i}{a-x_i}}$, where $\frac{f(a)+acr_i-x^*cr_i-t_i}{a-x_i}$ is a polynomial of degree $n-1$. Therefore, each key extraction query needs to compute $\mathcal{O}(n)$ exponentiations in $\mathbb{G}$. Since $\mathcal{A}$ makes at most $n-1$ such queries, $t = t' + \mathcal{O}(t_{H_1} \cdot n^2) + \mathcal{O}(t_H \cdot n) + \mathcal{O}(t_{exp} \cdot n^2)$, where t_{H_1} is the time required to compute the hash H_1, t_H is the time required to compute the hash H, and t_{exp} is the time required to compute the exponentiation in $\mathbb{G}$.

This concludes the proof of Theorem 1.

By applying a technique due to Fujisaki-Okamoto [5], we can easily convert the IND-name-CPA secure content-based encryption scheme CBE into a chosen ciphertext secure content-based encryption system in the random oracle model.

5 Securing Content-Centric Network

We apply our content-based encryption scheme to a content-centric network and demonstrate the applicability of our scheme for a real-world application.

5.1 Content-Centric Network Architecture

The content-centric network consists of a trusted third party (TTP) and three types of nodes as shown in Fig. 1:

- TTP: it provides the unique identifier for each content and acts as a private key generator (PKG) that generates the private decryption keys for the content;
- Provider node: it is a node which provides the content uniquely identified by its name to the other nodes in the network;
- Consumer node: it is a node which is authorized to obtain the content provided by the Provider node;
- Intermediate node: it is a node resided between a Provider node and a Consumer node, and it aims to forward an Interest sent by a Consumer node or a Data (here, we refer content as Data) returned by a Provider node.

In Fig. 1, an example of the content-centric network is presented, where C_1, C_2 and C_3 are the Consumer nodes; E_1, E_2, E_3, E_4 and E_5 are the Intermediate nodes; P_1 and P_2 are the Provider nodes. Note that a Provider node could also be an Intermediate node or a Consumer node for another content; a Consumer node could also be a Provider node or an Intermediate node for another content;

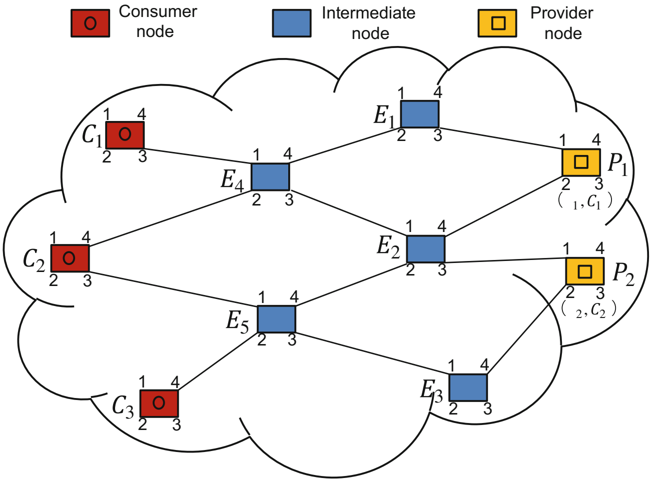


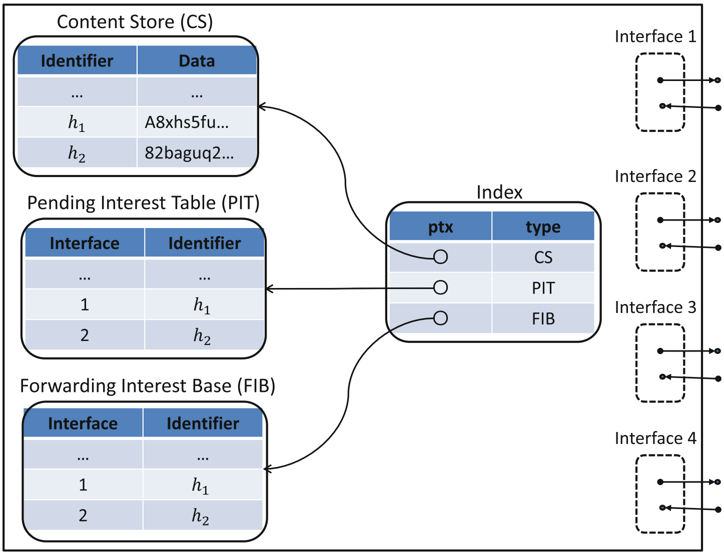
Fig. 1. Content-centric network architecture.

an Intermediate node could be a Provider node or a Consumer node for another content. Each node has multiple interfaces where the data comes or outputs. For simplicity, assume all the nodes including the Provider nodes, the Consumer nodes and the Intermediate nodes have four interfaces denoted as 1, 2, 3 and 4. Therefore, It allows multiple sources for data and can query them all in parallel.

The content-centric network communication is driven by the consumers of data. There are three content-centric network packet types, Route Establishing Request, Interest and Data. A Provider node which holds a content makes a Route Establishing Request to establish links among the nodes according to the content identifier. A Consumer node asks for an interested content by broadcasting its Interest over all available interfaces [9]. Any node which has received the Interest and has the data that satisfies it can respond with a Data packet (content chunk). Data is transmitted only in response to an Interest and consumes that Interest [9].

As shown in Fig. 2, the core content-centric network packet forwarding engine has three main data structures: FIB (Forwarding Information Base), CS (Content Store, i.e. buffer memory), and PIT (Pending Interest Table) [8]. The FIB is used to forward Interest packets toward content sources, i.e. the Provider nodes which have the matching Data. The CS is the same as the buffer memory of an IP router but it stores the received Data packet as long as possible. The PIT tracks Interests forwarded upstream toward content source(s), so returned Data can be sent downstream to its requester(s) [8]. After the PIT entries are used to forward a matching Data packet, they are erased immediately.

When an Interest arrives at an interface, if there is a matching entry in the CS, it will be returned from the same interface where the Interest comes. If there is no matching entry, the PIT is checked for an existing Pending Interest. If there is already a matching entry, the arrival Interface for the new Interest is added to the list in the corresponding PIT entry. If there is no already an



Content Store (CS)

Identifier	Data
...	...
h_1	A8xhs5fu...
h_2	82baguq2...

Pending Interest Table (PIT)

Interface	Identifier
...	...
1	h_1
2	h_2

Forwarding Interest Base (FIB)

Interface	Identifier
...	...
1	h_1
2	h_2

Index

ptx	type
○	CS
○	PIT
○	FIB

Fig. 2. Content-centric network forward engine model.

existing PIT entry, the FIB table is checked for forwarding information. If there is a corresponding entry, the Interest is forwarded accordingly, and the Interest and the arrival Interface are added to the PIT.

In content-centric network, the Data packet is not routed but simply follows the chain of PIT entries back to the original requester(s).

5.2 Secure Content Delivery in Content-Centric Network

The content-centric network presented in this section is built on our notion of content-based security for protection of the content. The proposed content-based encryption is applied to protect the content when it is acquired and transmitted over the content-centric network.

System Setup. To achieve the content confidentiality, the TTP executes the following steps to setup the system. It generates the master-key MK and the public system parameters params; chooses two collision-resistant hash functions $H_2 : \mathbb{Z}_p^* \rightarrow \{0,1\}^{l_1}$ and $H_3 : \mathbb{G}_T \rightarrow \{0,1\}^{l_2}$, where l_1, l_2 are positive integers. Then it publishes params and H_2, H_3, and keeps MK secret.

FIB Establishment. To establish the route among the three types of nodes, each node maintains a Forwarding Information Base (FIB) where each entry contains two fields: *Interface* and *Identifier*, as shown in Fig. 3. The routing establishment follows the next four steps:

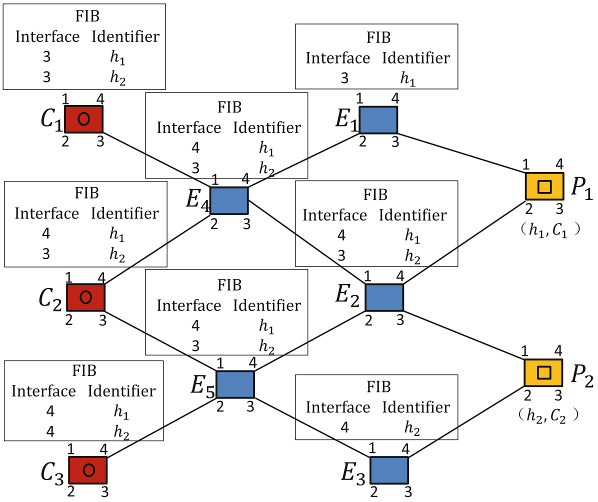


Fig. 3. FIB establishment.

1. If an original Provider node P_x wants to provide the content C_i, it sends the original name name_i of C_i and a unique tag ($\mathsf{tag}_i = H_3(\mathsf{C}_i)$) to the TTP. The TTP computes $N_i = H_1(\mathsf{name}_i \| \mathsf{tag}_i)$. The public key of C_i is set as $PK_i = N_i$. Then the TTP publishes (N_i, name_i). The Provider node P_x computes the identifier $h_i = H_2(N_i)$, and randomly selects a secure symmetric key k_i to compute C_i's ciphertext $e_i = \mathsf{SEnc}(\mathsf{C}_i, k_i)$ and $d_i = \mathsf{Encrypt}(k_i, PK_i)$ where SEnc is the symmetric encryption and $\mathsf{Encrypt}$ is the proposed content-based encryption.
2. The Provider node P_x generates a route establishing request $\mathsf{RER} = (h_i, T_i)$ where h_i is the header and T_i is the timestamp. P_x forwards the request RER to nearby nodes.
3. If an Intermediate node (or a Consumer node) receives this RER from interface j, the Intermediate node (or Consumer node) does the following operations:

   If there is no entry for h_i in the FIB of the Intermediate node (or Consumer node), it forwards the received request RER via each interface except the interface where RER arrived, and adds a new entry $[j, h_i]$ in its FIB where j is the interface the request arrived and h_i is the identifier; Otherwise, it discards the received RER;
4. Repeat Step 3 until all the Consumer nodes in the network receive the RER with identifier h_i and build an entry for h_i in their FIBs, as shown in Fig. 3.

Assume that in the content-centric network architecture, the Provider nodes can provide totally m pieces of contents. Each content C_i ($i \in [1, m]$) is uniquely identified by h_i. As shown in Fig. 3, the Provider node P_1 owns content C_1 and it provides (h_1, Data_1) where h_1 is the identifier of content C_1 with public key N_1, $\mathsf{Data}_1 = (\mathsf{e}_1, \mathsf{d}_1)$ is the ciphertext of content C_1. The Provider node P_2 owns content C_2 and it provides (h_2, Data_2) where h_2 is the identifier of content C_2 with public key N_2, $\mathsf{Data}_2 = (\mathsf{e}_2, \mathsf{d}_2)$ is the ciphertext of content C_2.

As shown in Fig. 3, P_1 forwards a Route Establishing Request message $\mathsf{RER}_1 = (h_1, T_1)$ where the header is h_1 from all its interfaces to its nearby nodes. P_2 forwards a Route Establishing Request message $\mathsf{RER}_2 = (h_2, T_2)$ where the header is h_2 from all its interfaces to its nearby nodes.

When the Intermediate node E_1 receives RER_1 from interface 3, it creates a new entry $[3, h_1]$ where 3 indicates the coming interface and h_1 is the identifier in its FIB, and then it forwards RER_1 from all its interfaces except interface 3. Similarly, when the intermediate node E_2 receives RER_1 from interface 4, it creates a new entry $[4, h_1]$ in its FIB, and then it forwards RER_1 from all its interfaces except interface 4.

When the intermediate node E_4 receives RER_1 from interface 4, it creates a new entry $[4, h_1]$ in its FIB, and then it forwards RER_1 from all its interfaces except interface 4. Then, when E_4 receives an RER_1 with the same identifier h_1 from interface 3, it discards this request, since there is already an entry for h_1 in its FIB.

When the Intermediate node E_5 receives RER_1 from interface 4, it creates a new entry $[4, h_1]$ in its FIB, and then it forwards RER_1 from all its interfaces except interface 4.

The Consumer nodes C_1, C_2 and C_3 receive the request RER_1 from interface 3, 4, 4, respectively, and they create entries $[3, h_1]$, $[4, h_1]$, $[4, h_1]$ in their FIBs, respectively. With the same approach, the Consumers nodes C_1, C_2 and C_3 receive the request RER_2 from interface 3, 3, 4, respectively, and they create entries $[3, h_2]$, $[3, h_2]$, $[4, h_2]$ in their FIBs, respectively.

Content Acquisition. As shown in Fig. 4, to support content acquisition, all the Intermediate nodes maintain two tables: a Pending Interest Table (PIT) and a Content Store (CS). In the PIT, each entry consists of two fields: *Interface* and *Identifier*. Differing from FIB, the interface in PIT is the interface where the Interest message comes, while the interface in FIB is the interface where the Route Establishing Request RER comes. In CS, each entry consists of two fields: *Identifier* and *Data*.

If a Consumer node C_y wants to acquire the content C_i identified by h_i, it firstly checks whether h_i is in its FIB. If there is an entry for h_i, it acquires the corresponding private decryption key $SK_{i,y}$ from the TTP. The private decryption key is generated according to the content-based encryption scheme in Sect. 4. After that the Consumer node C_y acquires the content C_i according to the following steps:

1. C_y checks the entry for h_i in its FIB. Assume the entry is $[k, h_i]$. C_y then forwards an interest message $\mathsf{Interest} = (h_i, T_i')$ where h_i is the header and T_i' is the timestamp from the interface k to the nearby nodes;
2. If a node receives this Interest from interface j, the node does the following operations:

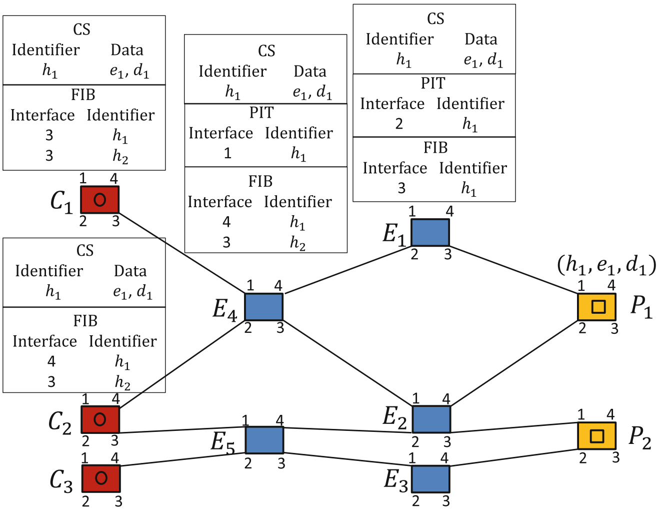


Fig. 4. Content acquisition.

- If there is no entry $[h_i, (e_i, d_i)]$ in the CS of the node, where h_i is the identifier and (e_i, d_i) is the data, the PIT is checked for an Interest entry with the same identifier. If there is already a matching entry, the arrival interface for the new Interest is added to the PIT list in the corresponding PIT entry. Otherwise, a new entry $[j, h_i]$ is added in the PIT where j is the interface the Interest comes and h_i is the identifier. The node forwards Interest from each interface except the interface where Interest comes, and then Step 2 is repeated;
- Otherwise, the node constructs a response data packet Data $= (h_i, e_i, d_i)$ where the header is h_i and the payload is (e_i, d_i), and forwards back Data from interface j;

3. If a node receives Data from the interface f, it checks if there is an entry for h_i in its CS. If no, creates a new entry $[h_i, (e_i, d_i)]$ and adds it to its CS. Otherwise, the new coming Data is not added to its CS. Then it checks its PIT. If there is an entry $[j, h_i]$ in its PIT, it forwards the response data packet Data back from the interface j according to the entry in PIT. After that it removes that entry in its PIT.
4. Repeat Step 3 until C_y receives the response data packet Data. Then, C_y decrypts d_i with the private decryption key $SK_{i,y}$ to obtain k_i, and decrypts e_i with k_i to obtain C_i.

Note that the response data packet Data includes two fields of CS entries, i.e. *Identifier* and *Data*.

In the following, we will give an example. Assume the Consumer node C_1 wants to acquire the content with identifier h_1. It firstly acquires the corresponding decryption key $SK_{1,1}$ securely from the TTP. It searches for the entry for h_1 in its FIB. If there is an entry $[3, h_1]$, C_1 forwards an interest message $\mathsf{Interest} = (h_1, T_1')$ where the header is h_1 from interface 3. Node E_4 receives this $\mathsf{Interest}$ from interface 1. Since there is no entry $[h_1, e_1, d_1]$ in E_4's CS, E_4 cannot provide the data to C_1. E_4 creates an entry $[1, h_1]$ in its PIT. Then, E_4 checks the entry for h_1 in its FIB and finds the corresponding entry $[4, h_1]$. E_4 forwards this interest message $\mathsf{Interest} = (h_1, T_1')$ via interface 4. The Intermediate node E_1 receives this interest message $\mathsf{Interest}$ via the interface 2. However, there is no entry $[h_1, e_1, d_1]$ in E_1's CS either. Therefore, E_1 also creates an entry $[2, h_1]$ in its PIT. Then, E_1 checks the entry for h_1 in its FIB and finds the corresponding entry $[3, h_1]$. E_1 forwards this interest message $\mathsf{Interest} = (h_1, T_1')$ via interface 3. Finally, the Provider node P_1 receives the $\mathsf{Interest}$ from interface 1.

When P_1 receives the interest message $\mathsf{Interest} = (h_1, T_1')$, it constructs the response data packet $\mathsf{Data} = (h_1, e_1, d_1)$ where h_1 is the header and (e_1, d_1) is the payload. Then, it forwards this response data packet Data via interface 1. When E_1 receives this Data, it adds an entry $[h_1, e_1, d_1]$ in its CS where h_1 is the identifier and (e_1, d_1) is the ciphertext of the content C_1. E_1 forwards this response data packet Data from interface 2 based on its PIT, and then removes the entry $[2, h_1]$ from its PIT. When E_4 receives this response data packet Data, it also adds an entry $[h_1, e_1, d_1]$ in its CS. E_4 forwards Data from interface 1 based on its PIT, and then removes the entry $[1, h_1]$ from its PIT. Finally, the Consumer node C_1 receives this response data packet $\mathsf{Data} = (h_1, e_1, d_1)$, and it decrypts d_1 with the secret key $SK_{1,1}$ to obtain the symmetric key k_1 and decrypts e_1 with k_1 to acquire the content C_1.

If the Consumer node C_2 wants to acquire the content with the identifier h_1, it firstly acquires the corresponding secret key $SK_{1,2}$ securely from the TTP. It searches for the entry with the identifier h_1 in its FIB. As there is an entry $[4, h_1]$ in its FIB, C_2 forwards an interest message $\mathsf{Interest}' = (h_1, T_1'')$ where the header is h_1 via interface 4. Node E_4 receives this $\mathsf{Interest}'$ from interface 2. Since there is already an entry $[h_1, e_1, d_1]$ in E_4's CS, it forwards the response data packet $\mathsf{Data} = (h_1, e_1, d_1)$ via interface 2 directly. Then the Consumer node C_2 receives this response data packet $\mathsf{Data} = (h_1, e_1, d_1)$, and it decrypts d_1 with the secret key $SK_{1,2}$ to obtain the symmetric key k_1 and decrypts e_1 with k_1 to acquire the content C_1.

As the content is encrypted with a symmetric key and the symmetric key is encrypted under the content-based encryption system, any node without a valid decryption key cannot decrypt the ciphertext and obtain the content. As the Intermediate nodes store the ciphertext of the content transmitted via them, if an Intermediate node wants to acquire the content it can directly decrypt the ciphertext when it has obtained a valid decryption key. If a nearby Intermediate node has stored the ciphertext, the Consumer node need not acquire the content from the Provider node, but from the Intermediate node directly instead.

6 Conclusion

We presented the notion of content-based encryption tailored for content-centric networks and also defined its security models. We proposed a content-based encryption scheme and proved that it is semantic secure in the random oracle model under the truncated decision ABDHE assumption. We applied our content-based encryption to protect the content delivered in the content-centric network.

Acknowledgment. The first author thanks the partial support from the National Natural Science Foundation of China (No. 61502086). We thank Xiaonan Wang for useful discussions about content-centric networks.

References

1. Abe, M., Cui, Y., Imai, H., Kiltz, E.: Efficient hybrid encryption from ID-based encryption. Des. Codes Crypt. **54**(3), 205–240 (2010)
2. Amadeo, M., Molinaro, A., Ruggeri, G.: E-CHANET: routing, forwarding and transport in information-centric multihop wireless networks. Comput. Commun. **36**(7), 792–803 (2013)
3. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM J. Comput. **32**(3), 586–615 (2003)
4. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
5. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. J. Cryptology **26**(1), 80–101 (2013)
6. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
7. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. **28**(2), 270–299 (1984)
8. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.: Networking named content. In: Proceedings of the 2009 ACM Conference on Emerging Networking Experiments and Technology, CoNEXT 2009, pp. 1–12, Rome, Italy, 1–4 December 2009
9. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N., Braynard, R.: Networking named content. Commun. ACM **55**(1), 117–124 (2012)
10. Lee, E., Lee, E., Gerla, M., Oh, S.: Vehicular cloud networking: architecture and design principles. IEEE Commun. Mag. **52**(2), 148–155 (2014)
11. Polyzos, G.C., Ahlgren, B., Jacobson, V., Koponen, T., Sitaraman, R.K., Trossen, D.: Information-centric networking: state of advance. In: 2011 ACM SIGCOMM Workshop on Information-Centric Networking, ICN 2011, pp. 25–25, Toronto, ON, Canada, 19 August 2011
12. 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)
13. Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K.C., Crowley, P., Papadopoulos, C., Wang, L., Zhang, B.: Named data networking. Comput. Commun. Rev. **44**(3), 66–73 (2014)
14. Zhao, Y., Zhuo, L.: A content-based encryption scheme for wireless H.264 compressed videos. In: Wireless Communications and Signal Processing (WCSP), 2012 International Conference, Proceedings, pp. 1–6, Huangshan, China, 25–27 October 2012