Enhancement of FTP-NDN Supporting Nondesignated Receivers

Arijit Karati, Chun-I Fan(✉), and Ruei-Hau Hsu

Department of Computer Science and Engineering,
National Sun Yat-sen University, Kaohsiung 80424, Taiwan
{arijit.karati,cifan,rhhsu}@mail.cse.nsysu.edu.tw

Abstract. Recently, Fan et al. proposed the File Transfer Protocol Based on Re-Encryption for Named Data Network (FTP-NDN) in order to reduce the cost that affects simultaneous access of same video services. The authors designed an elegant network architecture to deal with secure file transmission to the unknown potential customers. The technique is shown to be secured under Decisional Bilinear Diffie-Hellman (DBDH) assumption and computationally efficient than other existing techniques. Although, the protocol achieves data confidentiality, it does not provide node authentication during transmission in the NDN. In this paper, we propose an authentication scheme using the bilinear pairing. Performance evaluation shows that the proposed technique can be incorporated with considerable computation overhead.

Keywords: Confidentiality · Authenticity · Named Data Networking (NDN)

1 Introduction

In the modern digital era, Internet plays a vital role in making a data driven economy by integrating a number of objects into our daily lives. The Internet is the most common surface nowadays where people share their sensitive data among themselves. In the neoteric society, the data are not limited to textual form, rather it is now in the form of multimedia. Most of the users communicate with another by the help of IP address-based network although it is inefficient in content distribution. In this architecture, a user may face several problems, e.g., simultaneous access of a same file by several users, host authentication, etc. In this scenario, a recent communication architecture called Named Data Network (NDN) [8], built on hierarchical named data, solves the aforementioned problems. Evolution of this data-centric network architecture from the host-centric network architecture (IP) was primarily introduced by [6] in 2009. The motivation of NDN is to focus on the data (named as content) that a user wants

The work is supported by Information Security Research Center, National Sun Yat-sen University, Taiwan.

C.-Y. Chang et al. (Eds.): ICS 2018, CCIS 1013, pp. 590–598, 2019.
https://doi.org/10.1007/978-981-13-9190-3_65

rather to provide a reference of a specific address (named as hosts) where the data is located. In the NDN architecture, a data owner generates a file that is to be shared over the network. Then, it sends the file by appending its signature in the network. On the other hand, anyone as a receiver can verify the owner's signature instead of authenticating hosts. The detailed process is further discussed in Fig. 1. However, security is the major concern in this architecture. Since it has several advantages over IP-address based communication, NDN is the prime focus to the researchers. In 2012, Etefia et al. [2] designed a NDN-based hierarchical network topology to support the military communications. The authors showed that the mobile communication can be merged with the NDN to acquire the other military goals. In the same year, Hamdane et al. [5] constructed a hierarchical identity based scheme for the NDN. Here, they appended a signature with the name of packet to enrich the security. After that, Grassi et al. [4] devised an efficient vehicular communication concept by introducing the NDN. In their construction, vehicles use the mobile networks to communicate with another vehicle. Recently, Fan et al. [3] designed a secure file sharing protocol, named as FTP-NDN, using bilinear pairings for the NDN. The authors showed that their protocol is also applicable for the nondesignated/unspecified users. Although, the work withstands many attacks, we notice that there is no node authentication during the public transmission of an encrypted file from the source to destination party. Hence, we improve the scheme to its next level by introducing the node authenticity feature. The proposed work is based on bilinear pairing and outputs satisfactory result.

Rest of the manuscript is structured as follows: Sect. 2 discusses some useful concepts to understand our work. Section 3 discusses Fan et al.'s designed FTP-NDN with its limitation(s). After that in Sects. 4 and 5, we discuss the proposed authentication technique for FTP-NDN and its performance analysis. Finally, Sect. 6 concludes the paper with some remarks.

2 Preliminaries

This section illustrates some backgrounds of the proposed authentication technique. Besides, Table 1 comprises a list of notations used in this article.

2.1 NDN Architecture

Named Data Networking (NDN) is a future Internet architecture invigorated by years of the technological research into the network usage. The concept of NDN is derived from the Content-Centric Networking (CCN) which was introduced by Van Jacobson in 2006. Figure 1 shows a typical file transfer protocol in the NDN. It is a network of several active nodes where a data consumer sends an interest packet to its nearest node in order to access a remote file, and the further communications are as follows:

Table 1. Notation table

Notation	Meaning
msk	KGC computed secure master key
$params$	KGC computed public parameters
ID_i	User/Node i's valid identity
sk_i, K_i	User i's secret key
$RK_{\{i,j\}}, K_i$	Node i's re-encryption key from Node i to j and secret key of Node i
p	Considerably large odd prime
G_1, G_2	Two same p-ordered cyclic groups, one additive and another multiplicative
P	Generator of group G_1
$k \in_R K$	k is chosen at random from set K
$\perp$	Unique symbol indicates *return nothing*
Z_p^*	$Z_p - \{0\}$

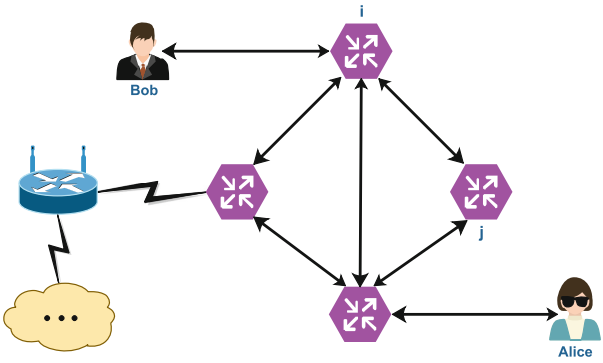

Fig. 1. Architectural overview of file transfer protocol in NDN

1. On receiving an interest packet, the nearest node finds whether it has file.
2. If it has file in its local cache, then
   – it answers the name and file as a data packet to the consumer.
3. Else, the node will request to its nearest nodes for file, and the process will continue until it touches the data source.
4. On receiving file, the node sends file to the requested consumer and keep a copy of file into its local cache temporarily for further same requests.

It may be noted that file can be provided by the nearest node to the consumer.

2.2 Identity-Based Encryption (IBE)

Novel concept of IBE was primarily introduced by Shamir [7] in 1984, however, the practical instantiation was drawn by Boneh [1] in 2001 using the bilinear pairings. The formal structure of IBE contains four following algorithms.

1. $(\mathsf{param}, \mathsf{msk}) \leftarrow \mathsf{Setup}(\mathsf{k})$: KGC runs the algorithm for security parameter k. It produces $params$, known to everyone and msk, known only to KGC.
2. $\mathsf{d_{ID}} \leftarrow \mathsf{Extract}(\mathsf{ID}, \mathsf{param}, \mathsf{msk})$: KGC runs the algorithm for input parameters $ID, param$, and msk. It produces user's private key d_{ID}.
3. $\mathsf{CT} \leftarrow \mathsf{Encrypt}(\mathsf{ID}, \mathsf{param}, \mathsf{m})$: A sender executes the algorithm for input parameters $ID, param$, and m. It outputs the ciphertext CT.
4. $\mathsf{m}/\perp \leftarrow \mathsf{Decrypt}(\mathsf{d_{ID}}, \mathsf{param}, \mathsf{CT})$: A receiver executes the algorithm for input parameters $d_{ID}, param$, and CT. It outputs the plaintext m or $\perp$.

2.3 Bilinear Map

Assume, $(P, Q) \in_R (G_1)^2$ and $(x, y) \in_R (Z_p^*)^2$ are chosen at random. An efficiently-computable map $e : G \times G \rightarrow G_T$ is said to be an admissible bilinear pairing if it holds the following properties:

1. Bilinear: Always produces $e(xP, yQ) = e(P, Q)^{xy}$.
2. Non-degenerate: Satisfies $e(P, P) = 1$, where $1 \in G_T$ is the identity element.
3. Computable: One efficient algorithm is always exists to compute $e(P, Q)$.

2.4 Intractable Problem

This section discusses two polynomial time (t) hard problems which are used in this work. We define $(x, y, z, u) \in_R (Z_p^*)^4$ chosen at random.

Definition 1 (Computational Diffie-Hellman (CDH)). *For an algorithm $\mathcal{A}$, computation of $Z = xyP$ from given (P, xP, yP) is infeasible in time t. The advantage ϵ of $\mathcal{A}$ to breach CDH is mentioned as* $\Pr[\mathcal{A}(xP, yP) = xyP] \geq \epsilon$

Definition 2 (Decisional Bilinear Diffie-Hellman (DBDH)). *For $\mathcal{A}$, decision of $u = xyz$ from $(P, xP, yP, zP, e(P, P)^{xyz})$ and $(P, xP, yP, zP, e(P, P)^u)$ is infeasible in time t. The advantage ϵ of $\mathcal{A}$ to breach DBDH is mentioned as* $|\Pr[\mathcal{A}(P, xP, yP, zP, e(P, P)^{xyz})] - \Pr[\mathcal{A}(P, xP, yP, zP, e(P, P)^u)]| \geq \epsilon$

3 Overview of Fan et al.'s Protocol

In this section, we explain the construction and limitation of the file transfer protocol by Fan et al. [3].

3.1 Details of the FTP-NDN

The protocol comprises six following algorithms:

1. $\mathsf{param} \leftarrow \mathsf{Setup}(\mathsf{k})$: For a security parameter k, KGC computes public parameter $param$ by executing the following tasks.
   (a) considers a bilinear pairing $e : G \times G \rightarrow G_T$ and a generator $P \in_R G$.

(b) chooses a symmetric key cryptosystem that considers an encryption ($SymEnc$) and a decryption ($SymDec$) functions with a l-bit key.
(c) selects three cryptographic hash functions as $H : \{0,1\}^* \rightarrow G$; $H_1 : G_T \rightarrow \{0,1\}^l$, and $H_2 : \{0,1\}^* \times G_T \rightarrow Z_p^*$.
(d) sets $msk = (s)$ and compute $P_{pub} = sP$ as the public key for $s \in_R Z_p^*$.
(e) Publishes the public parameter as

$$param = \{G, G_T, p, P, e, SymEnc, SymDec, P_{pub}, H, H_1, H_2\}$$

2. $\mathsf{sk_i} \leftarrow \mathsf{Extract(param, ID_i, s)}$: On receiving sufficient input parameters, KGC outputs the user's public key $pk_i = H(ID_i)$ and private key $sk_i = sH(ID_i)$.
3. $\mathsf{RK_{i,j}} \leftarrow \mathsf{ReKeyGen(param, ID_i, ID_j, s)}$: On receiving sufficient inputs, KGC generates a key as $RK_{i,j} = s\,(H(ID_j) - H(ID_i))$ to re-encrypt a ciphertext received by an NDN node i for another NDN node (or a user) j.
4. $\mathsf{CT_{i,0}} \leftarrow \mathsf{Enc(param, m, ID_i)}$: On receiving a plaintext message $m \in \{0,1\}^*$ along with sufficient inputs, the encryption process is performed as follows:
   (a) selects an element $K \in_R G_T$ and sets $r = H_2(m, K) \in Z_p^*$.
   (b) sets the ciphertext $CT_{i,0} = (C_1, C_2, U_{i,0})$ where $C_1 = SymEnc_{H_1(K)}(m)$, $C_2 = rP$, $U_{i,0} = K \cdot e\,(rH(ID_i), P_{pub})$.
5. $\mathsf{CT_{j,k+1}} \leftarrow \mathsf{ReEnc(param, CT_{i,k}, RK_{i,j})}$: On receiving $CT_{i,k} = (C_1, C_2, U_{i,k})$ with sufficient inputs, it calculates $U_{j,k+1} = U_{i,k} \cdot e(RK_{i,j}, C_2)$ and output the re-encrypted ciphertext as $CT_{j,k+1} = (C_1, C_2, U_{j,k+1})$. It can be noticed that $U_{j,k+1} = U_{i,k} \cdot e(RK_{i,j}, C_2) = K \cdot e(rH(ID_j), P_{pub})$.
6. $\mathsf{m}/\perp \leftarrow \mathsf{Dec(param, CT_{x,n}, sk_x)}$: On receiving sufficient inputs, the algorithm decrypts $m = SymDec_{H_1(K)}(C_1)$ where $K = U_{x,n} \cdot e(sk_x, C_2)^{-1}$. Finally it returns m if $C_2 \stackrel{?}{=} H_2(m, K)P$, otherwise, $\perp$.

3.2 Scope of Improvement in Fan et al.'s Protocol

Fan et al. showed that their protocol maintains confidentiality property under DBDH assumption, however, they didn't consider the authentication issue in between two adjacent nodes in the NDN. More specifically, an NDN node i forwards the ciphertext to its succeeding node $i+1$ without verifying the authenticity of ciphertext sent by its preceding node $i-1$. Therefore, an malicious node j can act like node $i-1$ and forward some unwanted ciphertext. In the next section we discuss an enhanced file transfer protocol based on [3] for the named data network.

4 Proposed Authentication for FTP-NDN

This section demonstrates the proposed authentication technique as a plugin for the Fan et al.'s FTP-NDN. The upgraded version of [3] with incorporated authentication constrains is discussed below.

1. $\mathsf{param} \leftarrow \mathsf{Setup(k)}$: For the security parameter k, KGC performs the same actions as discussed in Sect. 3.1 excepts the followings:

(a) chooses $(s,t) \in_R (Z_p^*)^2$ and sets $msk = (s,t)$.
(b) computes $P_{pub1} = sP$ and $P_{pub2} = tP_{pub1}$.
Finally, it publishes the public parameter as

$$param = \{G, G_T, p, P, e, SymEnc, SymDec, P_{pub1}, P_{pub2}, H, H_1, H_2\}$$

2. $sk_i \leftarrow$Extract(param, ID_i, s): The algorithm acts as the same like in Sect. 3.1. In addition, it computes $K_i = tH(ID_i)$ in the NDN. Therefore the secret key of user i is $sk_i = (sH(ID_i), tH(ID_i))$.
3. $RK_{i,j} \leftarrow$ReKeyGen(param, ID_i, ID_j, s): This algorithm executes same tasks defined in Sect. 3.1. In addition, it computes $K_i = tH(ID_i)$.
4. $CT_{i,0} \leftarrow$Enc(param, m, ID_i): On receiving a plaintext message $m \in \{0,1\}^*$ along with sufficient inputs, the encryption performs as follows:
   (a) selects an element $K \in_R G_T$ and sets $r = H_2(m, K) \in Z_p^*$.
   (b) computes $C_1 = SymEnc_{H_1(K)}(m)$, $C_2 = rP$, $U_{i,0} = K \cdot e(rH(ID_i), P_{pub1})$, and $V_{i,0} = K \cdot e\left(K_{ID_{i-1}}, hP_{pub1}\right)$, where $h = H_1(K \cdot e(rH(ID_{i-1}), P_{pub1}))$. Finally, it sets ciphertext $CT_{i,0} = (C_1, C_2, U_{i,0}, V_{i,0})$.
5. $CT_{j,k+1} \leftarrow$ReEnc(param, $CT_{i,k}$, $RK_{i,j}$): On receiving $CT_{i,k} = (C_1, C_2, U_{i,k}, C_{3,k})$ with sufficient inputs, it performs the followings:
   (a) computes $X = U_{i,k} \cdot e(RK_{i,i-1}, C_2)$ and sets $h = H_1(X)$
   (b) if $(V_{i,k} \neq e(H(ID_{i-1}), hP_{pub2}))$ then aborts the connection.
   else,
   – sets $h' = H_1(U_{i,k})$ and computes $V_{j,k+1} = e(K_i, h'P_{pub1})$
   – calculates $U_{j,k+1} = U_{i,k} \cdot e(RK_{i,j}, C_2)$
   – outputs the re-encrypted text as $CT_{j,k+1} = (C_1, C_2, U_{j,k+1}, V_{j,k+1})$.
   If it doesn't abort, then it is noticed that $V_{i,k} = e(H(ID_{i-1}), hP_{pub2})$ always hold as $e(H(ID_{i-1}), hP_{pub2}) = e(tH(ID_{i-1}), hP_{pub1}) = e(K_{i-1}, hP_{pub1})$ where $h = H_1(U_{i,k} \cdot e(RK_{i,i-1}, C_2)) = H_1(U_{i,k-1})$. Besides, we have $U_{j,k+1} = U_{i,k} \cdot e(RK_{i,j}, C_2) = K \cdot e(rH(ID_j), P_{pub})$.
6. m/$\perp \leftarrow$Dec(param, $CT_{x,n}$, sk_x): On receiving sufficient inputs, the algorithm performs following:
   (a) computes $Y = U_{x,n} \cdot e(RK_{x,x-1}, C_2)$ and sets $h = H_1(Y)$.
   (b) if $(V_{x,n} \neq e(H(ID_{x-1}), hP_{pub2}))$, then returns $\perp$
   else,
   – computes $K = U_{x,n} \cdot e(sk_x, C_2)^{-1}$.
   – decrypts $m = SymDec_{H_1(K)}(C_1)$.
   – if $C_2 \stackrel{?}{=} H_2(m, K)P$, returns m; otherwise returns $\perp$.

This completes the description of proposed authentication plugin mounted in Fan et al.'s FTP-NDN [3]. For better clarity, we further discuss it in Fig. 2. This picture considers five nodes (A, B, C, D, E), and two users $(Alice, Bob)$. In order to receive any remote file in the NDN, Alice requests to its nearby node C. On receiving the request, C checks whether it is present in its local memory or not. As we can see that the file does not exist, so, it asks its nearby nodes D, and then from D to A. The response comes in the reverse order, i.e., $Bob \rightarrow A \rightarrow D \rightarrow C \rightarrow Alice$ to achieve the file. Here, we mount our authentication technique suitably and the same is highlighted as blue.

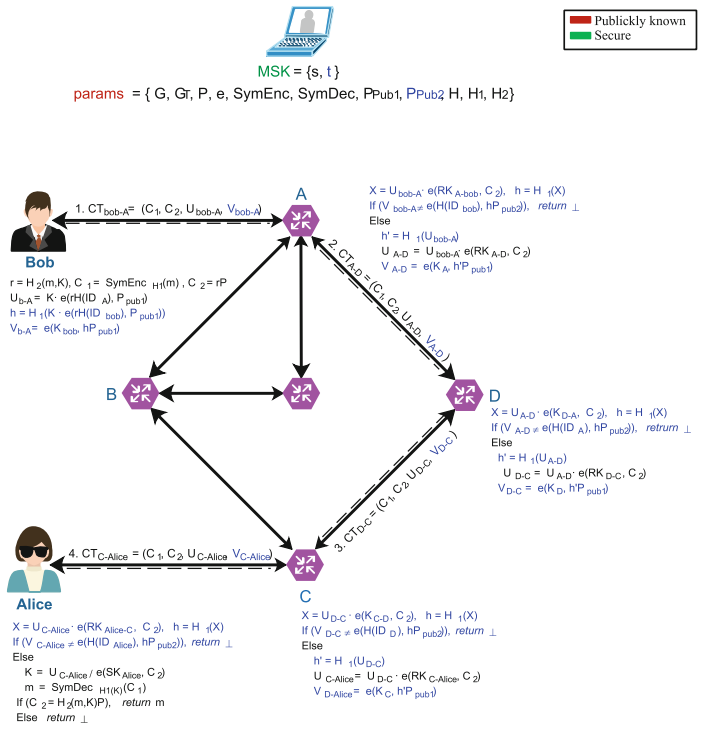

Fig. 2. The proposed authentication technique plugged in the Fan et al.'s FTP-NDN

5 Performance Measurement

This section discusses three essential aspects, such as, the computational, communication and storage costs. Here, we mention only the additional time required to active our technique in Fan et al.'s FTP-NDN [3]. Now, it illustrates each of the above mentioned aspects.

1. *Computational cost: Setup* algorithm executes one exponentiation along with the associated computation discussed in Fan et al. [3]. Similarly, each of *Extract* and *ReKeyGen* algorithms requires additionally one scalar multiplication operation. The *Enc* and *Dec* algorithms on the user sides require additionally two scalar point multiplications and two pairing computations to produce and verify a ciphertext respectively. The *Re-Enc* requires two scalar multiplication and three pairing computations.
2. *Communication cost:* The proposed technique requires one element along with the ciphertext size in Fan et al.'s [3] Protocol. Thus, additional computation cost due to authentication plugin is considered as $|G_T|$ which indicates the number of bits required to represent one element in G_T.

3. *Storage cost:* The proposed technique stores two additional keys, i.e., one during *Extract* and another during *ReKeyGen* algorithms. Therefore, the additional required memory is considered as $2|G_1|$.

Table 2 demonstrates the additional cost measured in different algorithms in Fan et al's FTP-NDN due to the integration of our authentication technique.

Table 2. Required additional cost measurement

	Encryption (User: Producer)	Decryption (User: Consumer)	Re-encryption (Node)	|Ciphertext|	Private Key (KGC)	Re-encryption Key (KGC)
Ours	$2T_s + 2T_p$	$2T_s + 2T_p$	$2T_s + 3T_p$	$\|G_T\|$	T_s	T_s

T_s: the cost of a scalar multiplication in an additive group;$|M|$: the length of M;T_p: the cost of a bilinear pairing;

6 Conclusion

Recently, Fan et al. designed an efficient File Transfer Protocol Based on Re-Encryption for Named Data Network (FTP-NDN). The protocol is useful where simultaneous secure transmission of same file/service are the constrain. The FTP-NDN is shown to be secured under Decisional Bilinear Diffie-Hellman (DBDH) assumption and computationally efficient than other existing techniques. However, The protocol doesn't support one main cryptographic aspect called authenticity property. In this work, we enhanced Fan et al.'s protocol by introducing an authentication techniques. Our technique is efficient and it can be mounted easily into the existing FTP-NDN. Besides, we measured performance of the proposed authentication technique and showed its efficiency.

References

1. 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). https://doi.org/10.1007/3-540-44647-8_13
2. Etefia, B., Gerla, M., Zhang, L.: Supporting military communications with named data networking: an emulation analysis. In: MILCOM, pp. 1–6 (2012)
3. Fan, C.I., Chen, I.T., Cheng, C.K., Huang, J.J., Chen, W.T.: FTP-NDN: file transfer protocol based on re-encryption for named data network supporting nondesignated receivers. IEEE Syst. J. **12**(1), 473–484 (2018)
4. Grassi, G., et al.: ACM hotmobile 2013 poster: vehicular inter-networking via named data. ACM SIGMOBILE Mobile Comput. Commun. Rev. **17**(3), 23–24 (2013)
5. Hamdane, B., Serhrouchni, A., Fadlallah, A., El Fatmi, S.G.: Named-data security scheme for named data networking. In: 2012 Third International Conference on the Network of the Future (NOF), pp. 1–6. IEEE (2012)
6. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L.: Networking named content. In: Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies, pp. 1–12. ACM (2009)

7. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
8. Zhang, L., et al.: Named data networking (NDN) project. Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC 157, 158 (2010)