Support Outsourcing Unsigncryption and Member Revocation Identity-based Proxy Signcryption Scheme with Drone Environment

Xiaodong Yang*, Wanting Xi, Ningning Ren, Jiaqi Wang, and Meijuan Li

Department of Computer Science and Engineering, Northwest Normal University, room 9c102, Teaching Building No.9, 967 Anning East Road, Anning District, Lanzhou City, China

*Email: y200888@163.com; swanteen@163.com

Abstract: The identity-based signcryption scheme combines the advantages of the signcryption scheme and the identity-based cryptosystem scheme, which has high calculation and communication efficiency and easy key management. In order to effectively solve the problems of high latency and data information privacy during the secure transmission of drone data to cloud servers, we propose an identity-based proxy signcryption scheme that supports outsourcing decryption and member revocation. The complex bilinear pairing calculation is outsourced to the cloud server to reduce the user's computing overhead. When an illegal user wants to access the data, the illegal user's identity value ID will be put into the revocation list and the update private key algorithm is executed to generate an update password for the unrevoked user Key, revoked users lose the ability to decrypt because they cannot obtain the updated key that matches them. The identity-based signcryption scheme protected by the proxy can satisfy unforgeability and forward security. Finally, the comparison results show that this solution is superior to previous work in terms of function and calculation time.

1. Introduction

The advancement of science and technology and the continuous reduction of manufacturing cost, UAV have entered the industrial production and People's Daily life from the military field. Nowadays, UAV have been widely used in weather monitoring, express transportation, traffic control, emergency rescue and other fields. Although UAV has a broad application prospect, it has not been widely used. This is mainly because there are still many unresolved problems in technology, safety and supervision of drones.

In terms of technology, drones have relatively high requirements for network delays, while cloud-based data processing mode has a large delay, so it is difficult to meet their requirements. UAVs also face severe security issues. UAVs are easily hijacked by hackers, and their data collection capabilities make them a very attractive target for malicious actors. Therefore, UAVs collect a large amount of data and users privacy data are also face the risk of exposure.

In response to the challenges faced by UAV application, edge-cloud computing [1] gives a feasible solution. Edge-cloud computing can meet the low-latency requirements of drones. Offloading part of the data processing tasks collected by UAV to edge computing devices can greatly reduce the delay of data processing. When the UAV transmits data to the cloud server, the drone is easily attacked by hackers and cause privacy data leakage. The identity-based signcryption scheme [2] can prevent...
hackers and other illegal users from hijacking data. The identity-based signcryption scheme introduces a signature to the identity-based cryptographic scheme proposed by Shamir [3] in 1984, which makes the scheme more efficient. UAV users can regard the edge node device as a proxy signer and authorize the edge node device to perform proxy signcryption on the transmitted data to ensure the security of data privacy. Eventually, the proxy signcryption uploads the signed data ciphertext to the Cloud Server (CS). After the data visitor sends a download ciphertext request to CS, the cloud server provides outsourcing decryption, and the data user obtains the plaintext through only a few calculations. When an illegal user wants to access the data, the user identity value ID will be put into the list of revocation and revoke that illegal user.

The concept of proxy signature was proposed by Mambo [4] et al., in 1996. The proxy signature is that the original signer authorizes his signature power to the proxy signer and then the proxy signer generates a valid signature on behalf of the original signer. Proxy signature [5,6] is an encryption algorithm that combines the idea of signcryption algorithm with proxy signature. Zheng [7] proposed the first signcryption scheme. Currently, Li and Chen [8] proposed an ID-based proxy signcryption scheme. However, their scheme are not protected by proxy and do not satisfied with unforgeability and forward security. Chen et al. [9] proposed a possible secure Identity-based proxy signcryption model under the assumption of CDH and BDH problem. In this paper, by combining the ideas of ID-based signcryption and proxy signature schemes, we propose an identity-based proxy signcryption scheme that supports member revocation and outsourced decryption in a drone environment, which is safer and more effective than the above schemes.

We organize the paper as follows. We introduced preliminary work in Section 2. The system model and the details of each entity are described in Section 3. Section 4 is the definition of the security model. Section 5 provides the structure of the specific scheme, while Section 6 provides the correctness analysis. Finally, we summarize the whole paper in Section 7.

2. Preliminary Knowledge

2.1. Bilinear Pairings

Bilinear pairing is the basis of modern cryptography. Boneh [10] constructed the first practical and verifiable identity-based encryption scheme using bilinear pairings in 2001. Since then, bilinear mapping [11] has been widely used in cryptography and has become an effective tool for constructing many encryption schemes. Assume that $G_0$ and $G_1$ are the addition and multiplication cyclic groups having similar large prime order $P$ respectively. The bilinear pair $\hat{e}: G_0 \times G_0 \rightarrow G_1$ is a mapping that satisfies the following three properties:

- **Bilinear**: $\forall P, Q \in G_0, a, b \in \mathbb{Z}_P$, then $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
- **Non-degenerate**: $\exists P, Q \in G_0$, so that $\hat{e}(P, Q) \neq 1_{G_1}$, where $1$ is the $G_1$ unit element.
- **Computability**: for $\forall P, Q \in G_0$, an effective polynomial time algorithm can be found to calculate $\hat{e}(P, Q)$.

2.2. Related Difficult Problems

Decisional Bilinear Diffie-Hellman [12] Problem (DBDH): Assuming that $G_0$ and $G_1$ are the additive cyclic group and multiplicative cyclic group of large prime numbers of order $P$ respectively. A bilinear mapping $\hat{e}: G_0 \times G_0 \rightarrow G_1$, given a quintuple $(P, aP, bP, cP, X \in G_1)$ where $a, b, c \in \mathbb{Z}_P$, determines whether the equation $X = \hat{e}(P, Q)^{abc}$ is true.

Computational Diffie-Hellman Problem (CDH): $G_0$ is an additive cyclic group of order $P$, given the triples $(P, aP, bP)$, calculate $abP \in G_0$.

3. Model of System
The system model of this paper consists of five entities as shown in Figure 1:

- **KGC**: This is an authority used for the user’s private key computing from their identity information, which can be trusted.
- **Delegator (UAV A)**: This entity wishes to delegate its signcryption authority to its proxy signcryptor (edge node device B).
- **Proxy Signcryptor (Edge Node device B)**: This is an entity that uses special information called “proxy key” to generate signed encrypted messages on behalf of the delegator (UAV A) and upload them to A trusted cloud service provider (CSP).
- **Cloud Server (CS)**: This entity provides storage services and sends signcrypted ciphertext to authorized users.
- **Data visitor (User C)**: An entity who wants to download at anytime and from anywhere via the Internet, restore the message content and verify its validity.

**Figure 1. System model.**

### 4. Security Model

Our scheme consists of seven algorithms, in this section we briefly define these algorithms.

- **Setup**: After taking security parameters $\lambda$ as input, KGC will output public parameters $\text{params}$ and retain the master key $\alpha$.
- **Extract**: The public parameters $\text{params}$ identity $ID$ and master key $\alpha$ are taken as input, and the private key $SK_{ID}$ is output. KGC must transmit it to the corresponding entity in a secure manner.
- **Key update generation**: The identity $ID$ master key $\alpha$ time $T$ and the private key $SK_{ID}$ are taken as input, returns the updated private key $SK_{ID,T} = SK_{ID} + UK_{ID,T}$.
- **Proxy certificate**: This algorithm run by the delegator . It takes $\text{params}$, the updated private key $SK_{ID,T} = SK_{ID} + UK_{ID,T}$, and the warrant $M_\Omega$ (including the delegation time, the identity of the delegator and the proxy signcryptor) as input, outputs the proxy certificate $PC$. The recipient can use the proxy certificate as verification information.
- **ProxyKeyGen**: The algorithm run by the proxy signcryptor. Takes as input the public parameters $\text{params}$, a warrant $M_\Omega$, proxy certificate and the updated private key $SK_{ID,T} = SK_{ID} + UK_{ID,T}$, returns proxy key $SK_{p_{B,T}} = SK_{B,T} + PC$. 
• Proxy signcryption: The algorithm run by the proxy signcryptor. The algorithm takes params, the identity of the data visitor $W_{ID}$, and $V_{ID,T}$, the proxy key $SK_{pt}$, the warrant $M_\Omega$ and the plaintext message $M$ as input, and outputs a signcryption text $CT$.

• Unsigncryption: take the public parameters params, ciphertext CT and $SK_{C,T}$ as input, and then confirm whether the ciphertext is correct, if "yes" continue and output the plaintext $M$, otherwise output $\perp$ as an error symbol.

5. Description of the Specific Scheme

In this section, we specifically describe the seven algorithms of the scheme in this paper.

5.1. System Establishment

Setup$(1^k)$: Take security parameters $1^k$ as input, the KGC select two cyclic groups $G_0$ and $G_1$ with prime order $P$, a generator of the cyclic group $G_0$ is $g$, the bilinear mapping satisfies $e: G_0 \times G_0 \rightarrow G_1$, and hash functions $H_1 : [0,1]^* \rightarrow G_0$, $H_2 : [0,1]^* \rightarrow Z_p^*$, $H_3 : G_1 \rightarrow [0,1]^*$, $H_4 : [0,1]^n \times G_1 \rightarrow Z_p^*$. The KGC chooses a random number $\alpha \in Z_{p^*}$ as the master key and calculates $g_1 = \alpha g$, also chooses a secure symmetric cipher $DK$. Then KGC discloses public parameters of the system params = $\{G_0, G_1, n, e, g, g_1, H_1, H_2, H_3, H_4, DK\}$ and keeps the master key $\alpha$ secret. The bit length of the message is $n$ bits. Each user has its unique identity value $ID$, add it to the user list LU, LU is used to store $(ID, username)$, user revocation list LR is used to store the identity value $ID$ of the user to be revoked.

5.2. Key Extraction

Extract $(ID, \alpha)$: Take identity value $ID$ and master key $\alpha$ as input, The KGC calculates $W_{ID} = H_1(ID)$ and $SK_{ID} = \alpha \cdot H_1(ID)$. Use $SK_{ID}$ as the user’s private key and send the private key back to its owner.

5.3. Update Key Generation

Key update generation $(\alpha, ID, T, SK_{ID})$: On input master key $\alpha$, identity value $ID$, time $T$. For any identity within $T$ time, if $ID \notin LR$ (the list of revocation), the user has not been revoked at $T$, then $V_{ID,T} = H_1(ID, T)$ and the updated key is $UK_{ID,T} = \alpha \cdot H_1(ID, T)$. Transfer the key to the user in a secure way, users who have not been revoked can update their private key as $SK_{ID,T} = SK_{ID} + UK_{ID,T}$

5.4. Proxy Certificate

5.4.1. Proxy Certificate $(params, SK_{A,T}, M_\Omega)$. Take the public parameter params, delegator private key $SK_{A,T}$ and a warrant $M_\Omega$ as input. Then the principal generates a proxy certificate $PC$. Select a random number $\beta \in Z_{p^*}$, set $h = \beta g$, $z = H_2(M_\Omega, h)$, then:

$$PC = z \cdot SK_{A,T} + ah = z(SK_{ID} + UK_{ID,T}) + \beta g_1$$  (1)

Send the $(M_\Omega, h, PC)$ triples to the proxy signcryptor, and the information in warrant $M_\Omega$ can help the receiver to verify.

5.5. ProxyKeyGen
ProxyKeyGen \((M_\Omega, h, PC, SK_{B,T}')\): After receives the \((M_\Omega, h, PC)\) triples, the proxy signcryptor verifies the validity of the proxy certificate received through calculation.

The verification formula is: \(\hat{e}(g, PC) = \hat{e}(g_1, z(W_{ID} + V_{ID,T}) + h)\), refer to equation (1) for the specific verification process is as in equation (2):

\[
\begin{align*}
\hat{e}(g, PC) &= \hat{e}(g, z\alpha H_1(ID) + z\alpha H_1(ID, T) + \beta g) \\
&= \hat{e}(g, z\alpha H_1(ID) + z\alpha H_1(ID, T) + \alpha \beta g) \\
&= \hat{e}(g, z(W_{ID} + V_{ID,T}) + h)
\end{align*}
\]

If the verification is correct, the proxy signcryptor calculates the proxy key as \(SK_{ps} = zSK_{B,T} + PC\).

5.6. Proxy Signcryption

5.6.1. Proxy signcryption \((\text{params, } M, W_{ID}, V_{ID,T}, SK_{ps}, SK_{B,T}', M_{\Omega}, T)\). When the proxy signcryptor wants to signcrypt the plaintext message \(M\), the proxy signcryptor first chooses the random number \(\varphi \in \mathbb{Z}_p\), and then calculates \(W_{ID} = H_1(ID)\), \(V_{ID,T} = H_1(ID, T)\),

\[
C_1 = \hat{e}(g, g_1)^\varphi \\
C_2 = H_3(\hat{e}(g_1, W_{ID} + V_{ID,T})^\varphi)
\]

\(C_3 = DK_{C_2}(M)\), \(C_4 = H_4(C_3, C_1)\), \(C = \varphi g_1 - (C_4 SK_{B,T} + SK_{ps})\), \(CT = (C_3, C_4, C, M_{\Omega})\). The proxy signcryptor then uploads the ciphertext \(CT = (C_3, C_4, C, M_{\Omega})\) to the cloud server (CS).

5.7. Unsigncryption

Unsigncryption \((\text{params, } SK_{C,T}', CT)\): The cloud server (CS) can outsource the decryption of the sign ciphertext. When user \(C\) wants to access data, he can download the encrypted ciphertext \(CT = (C_3, C_4, C, M_{\Omega})\) from the cloud server (CS) and execute the decryption algorithm as follows:

\(W_A = H_1(A)\), \(W_B = H_1(B)\), \(V_{A,T} = H_1(A, T)\), \(V_{B,T} = H_1(B, T)\),

\(C_1' = \hat{e}(g, C)\hat{e}(g_1, W_{B} + V_{B,T})^{z_{C_1'}} \hat{e}(g_1, z(W_A + V_{A,T}) + h)\),

\(C_2' = H_3(\hat{e}(C, W_C + V_{C,T}) \hat{e}(W_B + V_{B,T}, SK_{C,T}, SK_{C,T}'))^{z_{C_1'}} \hat{e}(z(W_C + V_{C,T}) + h, SK_{C,T}')\) Where the subscript \(A, B, C\) of \(W_{ID}\) and \(V_{ID,T}\) is the identity value ID. If \(C_4 = H_4(C_3, C_1')\), receive plaintext \(M = DK_{C_2'}(C_3)\); otherwise, the operation does not legally and return an error symbol \(\perp\).

6. Correctness Analysis and Performance Comparison Analysis

6.1. Correctness Analysis

With reference to equation (3) and (4), the establishment of the following equations as in equation (5) and equation (6) proves that the scheme we proposed is correct.
\[ C_1 = \hat{e}(g, C)\hat{e}(g_1, W_B + V_{B,T})^{z + C_1} \hat{e}(g_1, z(W_A + V_{A,T}) + h) \]
\[ = \hat{e}(g, C)\hat{e}(g_1, W_B + V_{B,T})^{z + C_1} \hat{e}(g, z(SK_A + UK_{A,T}) + \beta g_1) \]
\[ = \hat{e}(g, C)\hat{e}(g_1, W_B + V_{B,T})^{z + C_1} \hat{e}(g, PC) \]
\[ = \hat{e}(g, C)\hat{e}(g, C_4 \cdot SK_{B,T}) \hat{e}(g, zSK_{B,T} \cdot \hat{e}(g, PC)) \]
\[ = \hat{e}(g, C_4 \cdot SK_{B,T} \cdot \hat{e}(g, PC)) \hat{e}(g, SK_m) \]
\[ = \hat{e}(g_1, C_4) \hat{e}(g, SK_m) \]
\[ = C_1 \]

\[ C_2 = H_1(\hat{e}(C, W_C + V_{C,T})\hat{e}(W_B + V_{B,T}, SK_{C,T})^{z + C_2} \hat{e}(z(W_C + V_{C,T}) + h, SK_D)) \]
\[ = H_1(\hat{e}(C, W_C + V_{C,T}) \hat{e}(C_4, W_B + V_{B,T}) \hat{e}(zSK_{B,T} \cdot W_C + V_{C,T}) \hat{e}(zSK_{A,T} + \beta g_1, W_C + V_{C,T})) \]
\[ = H_1(\hat{e}(\phi g_1, C_4, SK_{B,T} \cdot W_C + V_{C,T}) \hat{e}(SK_m, W_C + V_{C,T})) \]
\[ = H_1(\hat{e}(\phi g_1, W_C + V_{C,T})) \]
\[ = H_1(\hat{e}(g_1, W_C + V_{C,T})) = C_2 \]

6.2. Performance Comparison
Table 1 compares the performance and functions of the proposed scheme and two related schemes, that is, the calculation cost of proxy signcryption, the calculation cost of unsigncryption, and the issue of whether user revocation can be effectively realized. In the bilinear mapping, E represents the exponentiation operation on the cyclic group, and P represents the bilinear pair operation on the cyclic group.

| Schemes | Proxy signcryption | Unsigncryption | Total | Revocation |
|---------|--------------------|----------------|-------|------------|
| [13]    | 2E+2P+3M           | 1E+5P+1M       | 3E+7P+4M | ×          |
| [14]    | 1E+1P+3M           | 1E+4P+1M       | 2E+5P+4M | ×          |
| Ours    | 2M                 | 4P+2M          | 4P+4M  | ✓          |

7. Conclusion and Future Work
This paper proposes an identity-based proxy signcryption scheme that supports outsourcing decryption and member revocation. On the basis of the identity-based signcryption scheme, an updated private key algorithm is added to generate updated keys for users who have not revoked. The revoked user loses the ability to unsign cryptography because he cannot obtain the updated key which matches him, thereby realizing user revocation. On this basis, the introduction of outsourcing decryption technology which will be a lot of decryption operation safety outsource to the cloud server. Data visitors can decrypt the ciphertext with only a very small amount of operations, which greatly improves the efficiency of decryption, and the cloud server cannot obtain any information about the plaintext, that solved the problem of the semi-trusted cloud server. Compared with the current schemes, this scheme not only adds the function of member revocation, but also greatly reduces the decryption cost of users. Our next step is to consider how to reduce the complexity of the signcryption algorithm. In addition, we can regard the UAV itself as a mobile edge computing center, unloading a part of the simple data computing tasks of the terminal equipment to the UAV for processing, which can greatly reduce the delay caused by the transfer of large amounts of data to the cloud server.
Acknowledgments
This work was supported by the Foundation of Northwest Normal University (NWNU-LKQN-14-7), Lanzhou Science and Technology Project (2013-4-22), the National Natural Science Foundation of China (61662069, 61562077) and China Postdoctoral Science Foundation (2017M610817).

References
[1] Pan Y, Chen M, Yang Z, Huang N and Shikh-Bahaei M 2018 Energy-efficient, NOMA-based mobile edge computing offloading IEEE Communications Letters 23(2) 310-313
[2] Malone-Lee J 2002 Identity-based signcryption IACR Cryptology ePrint Archive 2002 98
[3] Shamir A 1984 Identity-based cryptosystems and signature schemes Proc. of Workshop on the theory and Application of Cryptographic Techniques Springer 47-53
[4] Mambo M, Usuda K and Okamoto E 1996 Proxy signatures: Delegation of the power to sign messages IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences 79 1338–54
[5] Zhou C, Zhao Z, Zhou W and Mei Y 2017 Certificateless key-insulated generalized signcryption scheme without bilinear pairings Security and Communication Networks 2017 17
[6] Gamage C, Leiwó J and Zheng Y 1999 An efficient scheme for secure message transmission using proxy-signcryption Proc. of the 22nd Australasian Computer Science Conference (Springer) 420–431
[7] Zheng Y and Imai H 1998 How to construct efficient signcryption schemes on elliptic curves Information Processing Letters 68 227-33
[8] Li X and Chen K 2004 Identity based proxy-signcryption scheme from pairings in Proc. of Services Computing 2004(SCC 2004), Proceedings 2004 IEEE International Conference on, IEEE p 494–497
[9] Chen S X, Zhou S X, Yao X F and Li F W 2011 Efficient identity-based proxy signcryption scheme Application Research of Computers 7 084
[10] Boneh D, Franklin M 2001 Identity-based encryption from the Weil pairing Annual International. Cryptology Conference (Springer) 2001 213-29
[11] SHAO J, CAO Z and WANG L 2007 Proxy re-signature schemes without random oracles The 8th International Conference on Cryptology (Springer) 197-209
[12] Escala A, Herold G and Kiltz E 2017 An Algebraic Framework for Diffie---Hellman Assumptions Journal of Cryptology 31 1-47
[13] Yu H, Wang Z, Li J and Gao X 2018 Identity-based proxy signcryption protocol with universal composability Security and Communication Networks 2018 11
[14] Ming Y, Feng J and Hu Q J 2014 Secure identity-based proxy signcryption scheme in standardmodel Journal of Computer Applications 34 2834–39