


A Secure Group Data Encryption Scheme in Intelligent Manufacturing Systems for IIoT

Ming Te Chen, National Chin-Yi University of Technology, Taiwan*

 <https://orcid.org/0000-0001-9583-4419>

ABSTRACT

In recent years, there are many industries that imported intelligent systems to help them make the intelligent factory and get product record analysis to evaluate product rate. These intelligent systems could generate product records, store them to the on-line database, and provide product rate analysis from these records. Due to the rapid development of internet of things (IoT), the stockholder can construct its own smart factory with the smart intelligent system to develop its own industrial internet of things (IIoT) architecture. With the help of IIoT, the smart intelligence system can collect data information with IoT sensors embedded into each machine in the production line. However, there are some security issues arising between smart intelligent systems and IoT devices. In addition, the authors also discovered that there are fewer methodologies to talk about the data security during the machine transmitting its censored data to the other machines under the same network environment.

KEYWORDS

Encryption, Industrial Internet of Things, Intelligent Manufacturing System, Internet of Things

INTRODUCTION

In recent years, intelligent manufacturing is getting popular in the world. Each industry imports intelligent systems to establish their intelligent product line group and also collects product information from these intelligent systems in order to increase product rate by analyzing these records. At the same time, many traditional industries are located in mid of Taiwan and some of them are also beginning to import intelligent systems into their product line. Most stockholders begin to apply Internet of things (IoT) devices to equip their working machines and also fetch the final production data from them. This has also produced the trend called industrial IoT (IIoT), by which the manufacturing machine can equip some proper IoT devices or sensors to fetch the internal production process information and transfer these censored data to the console dashboard of the stockholder's terminal machine or cloud service Mell et al. (2009). Under this architecture, the stockholder can become familiar with the status of each machine whether it is working or not. At the same time, each machine also needs the wire or wireless network to transmit its own censored data by using network protocols such as transport layer security (TLS) to withstand the security issue trends Chhetri et al. (2017). However, during transferring these data to the other machine, only applying the TLS protocol or other communication

DOI: 10.4018/IJSI.312577

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

protocol is not enough. Sanchez et al. (2020) thought that the data of the transmitted message also need to be protected well. Hence, the author's goal is confirming data integrity in the transferring process between machines. The author proposes a secure encryption scheme to withstand a chosen cipher-text attack, so that only a designated machine can decrypt the final cipher-text for the IIoT architecture. On one hand, the author's proposed scheme could offer a secure encryption framework for intelligent systems to withstand a chosen cipher-text attack to deliver their desired messages. In addition, the author's scheme offers a constant system group size when intelligent systems increase and provides efficiency comparison in this paper.

RELATED WORKS AND SECURITY DEFINITION

Related Works

In recent years, the intelligent manufacturing system has become an important research topic and many factories' stockholders have moved to industry 4.0 through the IoT. At the same time, they must construct their facilities or equipment with IoT devices (Atzori et al., 2010) to help construct the IIoT (Sisinni et al., 2018) environment and collect production data of each machine. Through the IIoT, stockholders could also build up the cyber physical system to perform further data analysis on the collected data from each machine. However, Tuptuk et al. (2018) and Zhou et al. (2019) pointed out that there are some examples of security attacks in intelligent manufacturing systems. They also suggested some solutions on the current known attacks on this aspect. On the other hand, due to the constrained nature of IoT devices, the intelligent manufacturing system must collect data from the sensors embedded in the working machine of the factory. In addition, each machine also has to transmit the censored data by using WiFi, Bluetooth, Zigbee or WirelessHART communication protocols. As a result, intelligent manufacturing systems have become decentralized systems (Moghaddam et al., 2018) and can form their network topology in the production line. Under this architecture, each machine must apply some well-known secure protocol such as TLS to protect the message. However, recently, Sanchez et al. (2020) argued that only applying the TLS protocol is not enough. Sanchez et al. figured out end-to-end security, which could guarantee the data are not exposed to the adversary or other third parties between each intelligent machine system. However, their scheme adopts the attributed-based encryption basic ideal and does not provide any mechanism. Hence, the author proposed a secure encryption scheme that is lightweight and suitable for intelligent systems to encrypt product record between other intelligent systems; the author also offered a formal security proof with this scheme. In addition, the author adopted the certificateless techniques (Al-Riyami & Paterson, 2003; Girault, 1991) to design his scheme and reduce authentication computation cost between each machine. At the same time, the author made this choice because the elliptic-curve key is short and is fit for preserving IoT devices and because an elliptic-curve group has the same security level that the Rivest-Shamir-Adleman (RSA) system provides, with large prime module numbers and corresponding large keys. Hence, the author took an elliptic curve as building block and offered the computation comparison with the RSA system. Besides offering security proof, the author's proposed approach could maintain an intelligent system size limit in the constant value.

Security Definitions

This section provides some security definitions.

Definition One: Bilinear Group

In the following, the author briefly reviews the bilinear map (Zhang et al., 2004) and bilinear group definitions:

- G and G_1 are two (multiplicative) cyclic groups of prime order p .
- g is a generator of G .
- e is a bilinear map $e : G \times G \rightarrow G_1$.

Given G and G_1 as two bilinear groups as above, a bilinear map has the following properties:

- Bilinearity, which means that, for all $u, v \in G$, and $a, b \in \mathbb{Z}_p^*$, $e(u^a, v^b) = e(u, v)^{ab}$.
- No-degeneracy, which means $e(g, g) \neq 1$.

Definition Two: l -Weak Bilinear Diffie-Hellman Inversion Assumption

Given G as a bilinear group of order p and α as a random number in \mathbb{Z}_p^* , it is possible to define the problems related to the l -weak bilinear Diffie-Hellman inversion l -wBDHI* (Boneh et al., 2005) as follows.

Given $(g, h, g^\alpha, g^{\alpha^2}, \dots, g^{\alpha^l})$, an algorithm A has advantage ϵ in solving decisional l -wBDHI* in G if:

$$\Pr \left[A \left(g, h, g^\alpha, g^{\alpha^2}, \dots, g^{\alpha^l}, e(g, g)^{\alpha^{l+1}} \right) = 1 \right] - \Pr \left[A \left(g, h, g^\alpha, g^{\alpha^2}, \dots, g^{\alpha^l}, e(g, g)^z \right) = 1 \right] \geq \epsilon \quad (1)$$

where the probability is over the random choices of generators g, h in G , the random choice of z , the random choice of $\alpha \in \mathbb{Z}_p^*$, and the random bits used by A .

Definition Three: Indistinguishable Selective Identity Chosen Cipher-Text Attack

In this section, the author defines an experiment that includes an adversary A and a challenger C in the following phases:

Setup: The challenger C runs the setup algorithm first, then gives A the system parameters $params$, and finally keeps the master-key to itself.

Phase One: A could issue queries (q_1, \dots, q_m) , where each issued query q_i belongs to one of the following:

- **Private Key Queries (ID_i):** When the challenger C received this kind query, it performs the keygen algorithm to generate d_i corresponding to ID_i and forwards d_i to the C .
- **Decryption Queries ($ID_i, CT_{j,i}$):** When the challenger C received this query from A , it runs the decryption algorithm Decrypt according to ID_i 's private key to decrypt the cipher-text $CT_{j,i}$ and sends the result back to the A .

Challenge: Once A decides Phase One is over, it outputs an identity ID^* and two equal length plaintexts $(M_0, M_1) \in M$ that it wishes to be challenged. The restriction is that A could not make a private key query for the target ID^* or a prefix of ID^* . Then, C picks a random bit $b \in \{0, 1\}$, makes the challenge cipher-text CT^* to be the $Encrypt(params, ID^*, M_b)$, and forwards it to A .

Phase Two: A could issues additional queries (q_{m+1}, \dots, q_n) , where each q_i is one of:

- Private key query (ID_i) where $ID_i \neq ID^*$ and ID_i is not prefix of ID^* .
- Decryption query ($C_i \neq CT^*$) for ID^* or any prefix of ID^* . In both cases above, C usually responds as in Phase One, and these queries may be adaptive by A .

Guess: Finally, A outputs a final guess result $b' \in \{0, 1\}$ and wins if $b = b'$. Then, it is possible to define such an adversary A as an indistinguishable selective identity chosen cipher-text (IND-sID-CCA) adversary. The author defines that advantage of the adversary A in attacking the above scheme that the author assumes as ϵ and:

$$Adv_{\epsilon, A}(t, \theta) = \left| \Pr[b = b'] - 1/2 \right| \quad (2)$$

where the probability is over the random bits the challenger C and the adversary A use in the polynomial time bound t and the security parameter θ .

Definition Four: Security

The decisional $l - wBDHI^*$ assumption holds in G if no polynomial bounded adversary has advantage better than negligible in solving the decisional $l - wBDHI^*$ problem in G .

Theorem One: Given G as a bilinear group of prime order p and that the decision $(t, \epsilon, l) - wBDHI^*$ assumption holds in G , then the previously defined data encryption scheme for intelligent system is (t, ϵ, l) IND-sID-CCA secure for arbitrary l, q_s , and $t' < t - (\tau \cdot l \cdot q_s)$, where τ is the maximum time for an exponentiation in G and at most q_s times decryption query.

THE PROPOSED SCHEME

This section provides some definitions related to the author's proposed scheme.

Preliminary

- p : A large prime order number which forms a finite primes group.
- l : A security parameter that defines the length of a hashed message.
- g : A group generator that is the base point.
- M : An original message in which an intelligent system communicates with others with the same length of the hashed message in the same production line.
- \oplus : Bit exclusive or operation that is used for encryption and decryption.
- q : A product line group order that contains at most q intelligent manufacturing systems.
- i : An intelligent system that performs encryption/decryption operations in a product line group, where $i \in \{0, q - 1\}$.
- $CT_{i,j}$: A cipher-text that was generated by an original plain-text M from one system i to another system j , where i and j belong to the same intelligent system group, that is, $i, j \in \{0, q - 1\}$.
- gpk : A group manager's public key that could be used to perform the encryption operation for system members.
- gmk : A group manager's secret key that could be used to perform the decryption operation for system members.

- ID_i : A group member i 's identity value string, where $i \in \{0, q-1\}$.
- pk_i : A public key of an intelligent system i , where i belongs to an intelligent system of some product line group, that is, $i \in \{0, q-1\}$.
- d_i : A private key of an intelligent system i , where i belongs to an intelligent system of some product line group, that is, $i \in \{0, q-1\}$.
- $h(\cdot)$: A secure hash function that maps $\{0,1\} \rightarrow \{0,1\}^l$ to l -bits integer value.
- $H_1(\cdot)$: A secure hash function that maps the points to a l -bits string value, where $G \cdot G \rightarrow \{0,1\}^l$.

Setup Phase

In this phase, the author assumed an intelligent system group whose order is q and for which each system could publish its public key with other parameters by the following steps:

1. An intelligent system manager C selects its master secret key $gmk = s \in_R Z_p^*$ and $t \in Z_p^*$. Then, it also generates the corresponding group public key $gpk = (g, g^s, g^t, h_1, \dots, h_q)$ with q random numbers uniformly chosen in Z_p^* .
2. The intelligent system manager C publishes system parameters $(g, g^s, g^t, h_1, \dots, h_q)$ to the group system members and finishes this phase.

Keygen Phase

In this phase, the author assumed a total of q intelligent manufacturing systems to communicate with each other in the same production line. One of them (i.e., i , where $i = \{1, \dots, q\}$) runs a key-generation algorithm to generate its private key in the following steps:

1. i inputs the master public key g^s and selects a random number $r_i \in Z_p^*$. Then, it inputs its own identity ID_i and gets its own private key d_i by the key-generation function, as follows:

$$d_i = \frac{s}{r_i + h(ID_i)} \quad (3)$$

2. The system i also produces its public key $pk_i = g^{d_i}$, publishes its public key into the group members, and finishes this phase.

Encryption Phase

In this phase, there is an intelligent system (called j as a message sender) attempting to forward its own cipher-text $CT_{j,i}$ to the receiver i in the same product line group. This intelligent system j has to perform the following steps:

1. It randomly chooses two parameters g^{γ_j} and $(h_1)^{\gamma_j}$, where $\gamma_j \in Z_p^*$, g^{γ_j} is R_j , and $(h_1)^{\gamma_j}$ is W_j .

2. It prepares a message M that contains product detail information and production parameters. Then, it computes the cipher-text $CT_{j,i}$ by Equation 4:

$$CT_{j,i} = H_1 \left[\left[e \left(g, pk_i^{\gamma_j} \cdot W_j \right) \cdot e \left(R_j, \prod_{i=2}^q h_i \right) \right] \right] \oplus M \quad (4)$$

3. After computing the cipher-text $CT_{j,i}$ successfully, j forwards $CT_{j,i}$ with W_j and R_j to the system i . When i has received this cipher-text tuple $(CT_{j,i}, W_j, R_j)$, it terminates this phase and enters the next phase.

Decryption Phase

In this phase, when i has received this cipher-text $(CT_{j,i}, W_j, R_j)$ from j , it could decrypt it by its own private key d_i in the following steps:

1. i uses its own private key d_i to compute k_i by Equation 5:

$$k_i = \left[e \left(R_j, g^{d_i} \right) \cdot e \left(R_j, \prod_{i=1}^q h_i \right) \right] \quad (5)$$

2. The original construction of the ciphertext $CT_{j,i}$ becomes as follows:

$$\begin{aligned} CT_{j,i} &= H_1 \left[\left[e \left(g, pk_i^{\gamma_j} \cdot W_j \right) \cdot e \left(R_j, \prod_{i=2}^q h_i \right) \right] \right] \\ \oplus M &= H_1 \left[\left[e \left(g, (pk_i \cdot h_1)^{\gamma_j} \right) \cdot e \left(g^{\gamma_j}, \prod_{i=2}^q h_i \right) \right] \right] \\ \oplus M &= H_1 \left[\left[e \left(g^{\gamma_j}, \frac{s}{g^{r_i + h(ID_i)}} \right) \cdot e \left(g^{\gamma_j}, h_1 \right) \cdot e \left(g^{\gamma_j}, \prod_{i=2}^q h_i \right) \right] \right] \\ \oplus M &= H_1 \left[\left[e \left(R_j, g^{d_i} \right) \cdot e \left(R_j, \prod_{i=1}^q h_i \right) \right] \right] \oplus M = H_1(k_i) \oplus M \end{aligned} \quad (6)$$

3. i computes the decryption key k_i from Equation 5 and takes k_i to decrypt the cipher-text $CT_{j,i}$ from the following Equation 7:

$$\begin{aligned} CT_{j,i} \oplus H_1(k_i) &= \left(H_1 \left[\left[e \left(R_j, g^{d_i} \right) \cdot e \left(R_j, \prod_{i=1}^q h_i \right) \right] \right] \oplus M \right) \\ \oplus \left(H_1(k_i) \right) &= \left(\left[e \left(R_j, g^{d_i} \right) \cdot e \left(R_j, \prod_{i=1}^q h_i \right) \right] \oplus M \right) \\ \oplus H_1 \left(\left[e \left(R_j, g^{d_i} \right) \cdot e \left(R_j, \prod_{i=1}^q h_i \right) \right] \right) &= M \end{aligned} \quad (7)$$

SECURITY ANALYSIS

This section provides functional analysis about the author's proposed scheme and security analysis in appendix section.

Correctness

In the proposed scheme, the author could check that the receiver i could obtain the message M from the system j in the above Equations 5, 6, and 7. After i received the cipher-text from j , it could decrypt it by using its own private key d_i . Then, i could obtain the original message M .

Certificateless

The proposed scheme includes a KGC role (Boneh et al., 2001; Girault, 1991; Hu, 2007) whose goal is to generate each machine's private key in the same production line group without each machine performing authentication with others. The author also applies a random number r_i and $h(\cdot)$ to prevent the KGC impersonating each machine i during the keygen algorithm, where $i = \{1, \dots, q\}$. At the same time, the author also assumes that KGC is a trusted party, in this paper.

Efficiency Comparison

The author assumed that the proposed scheme is based on the bilinear mapping in elliptic curves. A comparison with the RSA algorithm evidenced that a small key length will achieve the same security of RSA in elliptic curves. The author could take elliptic curves as implementation environment and discover that the key length of 192-bit has the same security level with the key length of 1024-bit in RSA.

In the following, the author assumes that EC_p is a bilinear pairing operation on elliptic curve, EC_m is the point scalar multiplication operation on elliptic curve, EC_A is two points addition operation on elliptic curve, T_H is the computation time of one-way hash function, $T(S)_{E/D}$ is the time of an asymmetric encrypting/decrypting operation, $T(D)$ is the time of a Diffie-Hellman exponential operation, I is the computation cost of the inverse operation, \oplus is the computation cost time of the exclusive-or operation, and M is the multiplication operation in a modulo (Li et al., 2001; Zhang et al., 2004).

Then, the author assumes that a random number in Z_p^* is 160 bits, a point over elliptic curve is 160 bits, the output size of SHA-1 is 160 bits, and the block size of AES is 128 bits. Based on Li et al.'s (2001) study, the author assumes that E is the computation cost of a modulo exponentiation in a 1024-bit modulo and he also establishes the relations such as:

$$E \approx 8.24EC_m, E \approx 600T_H, E \approx 3.2EC_p, EC_A \approx 5M, I \approx 0.9M$$

and $E \approx 240$. Table 1 shows the performance comparisons.

Table 1. Computation cost evaluation in each phase

Phase	Cost Evaluation	Approximation
Key-generation phase	$1E + 1M$	$241M$
Encryption phase	$2E + (q - 1)M + 2EC_p + 1 \oplus$	$(q + 629)M + 1 \oplus$
Decryption phase	$qM + 2EC_p + 1 \oplus$	$(q + 150)M + 1 \oplus$
Total cost	$3E + (2q - 1)M + 4EC_p + 2 \oplus$	$(2q + 1020)M + 2 \oplus$

q : A product line group order that it contains at most q intelligent manufacturing systems

\oplus : exclusive-or bit-wise operation

CONCLUSION

In this paper, the author proposed a secure data encryption scheme for intelligent systems and offered a formal security proof in the Appendix. Not only could the author's framework provide data protection, but also limit system size in constant. The author's future goal is to design a hierarchical secure data authentication scheme for intelligent manufacturing systems in the IoT network or heterogenous network. At the same time, the author considers lightweight and practical authentication schemes as further future research goals.

REFERENCES

- Al-Riyami, S. S., & Paterson, K. G. (2003). Certificateless public key cryptography. In C. S. Lai (Ed.), *Advances in cryptology - ASIACRYPT 2003* (pp. 452–473). Lecture Notes in Computer Science. Springer. doi:10.1007/978-3-540-40061-5_29
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of things: A survey. *Computer Networks*, 54(15), 2787–2805. doi:10.1016/j.comnet.2010.05.010
- Boneh, D., Boyen, X., & Goh, E.-J. (2005). Hierarchical identity-based encryption with constant size ciphertext. In *Proceedings of Advances in cryptology - EUROCRYPT 2005*. Lecture Notes in Computer Science (vol. 3494, pp. 440–456). Academic Press.
- Boneh, D., & Franklin, M. K. (2001). Identity-based encryption from the Weil pairing. In *Proceedings of Advances in Cryptology 2001*. Lecture Notes in Computer Science (vol. 2139, pp. 213–229). doi:10.1007/3-540-44647-8_13
- Chhetri, S. R., Rashid, N., Faezi, S., & Al Faruque, M. A. (2017). Security trends and advances in manufacturing systems in the era of industry 4.0. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)* (pp. 1039–1046). doi:10.1109/ICCAD.2017.8203896
- Girault, M. (1991). Self-certified public keys. In *Proceedings of Advances in cryptology - EURO-CRYPTO'91*. Lecture Notes in Computer Science (vol. 547, pp. 490–497). doi:10.1007/3-540-46416-6_42
- Hu, B. C., Wong, D. S., Zhang, Z., & Deng, X. (2007). Certificateless signature: A new security model and an improved generic construction designs. *Codes and Cryptography*, 42(2), 109–126. doi:10.1007/s10623-006-9022-9
- Li, Z., Higgins, J., & Clement, M. (2001). Performance of finite field arithmetic in an elliptic curve cryptosystem. In *Proceedings of the 9th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications System (MASCOTS'01)* (pp. 249–256). Academic Press.
- Mell, P., & Grance, T. (2009). *Effectively and securely using the cloud computing paradigm*. NIST.
- Moghaddam, M., Cadavid, M. N., Kenley, C. R., & Deshmukh, A. V. (2018). Reference architectures for smart manufacturing: A critical review. *Journal of Manufacturing Systems*, 49, 215–225. doi:10.1016/j.jmsy.2018.10.006
- Sanchez, A. M., Barceló, M., Astorga, J., & Urbieto, A. (2020). Securing IIoT using defence-in-depth: Towards an end-to-end secure industry 4.0. *Journal of Manufacturing Systems*, 57, 376–378.
- Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial Internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, 14(11), 4724–4734. doi:10.1109/TII.2018.2852491
- Tuptuk, N., & Hailes, S. (2018). Security of smart manufacturing systems. *Journal of Manufacturing Systems*, 47, 93–106. doi:10.1016/j.jmsy.2018.04.007
- Zhang, F., Safavi-Naini, R., & Susilo, W. (2004). An efficient signature scheme from bilinear pairings and its applications. In *Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography*. Lecture Notes in Computer Science (vol. 2947, pp. 277–290). doi:10.1007/978-3-540-24632-9_20
- Zhou, W., Jia, Y., Peng, A., Zhang, Y., & Liu, P. (2019). The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved. *IEEE Internet of Things Journal*, 6(2), 1606–1616. doi:10.1109/JIOT.2018.2847733

APPENDIX: SECURITY PROOF

Proof: The supposition is that A has advantage ϵ in attacking the above proposed encryption system. As this point, the author uses A to construct an algorithm C that could be used to solve the decision $l - wBDHI^*$ problem (Boneh et al., 2005) in G and also takes Boneh et al. (2005)'s security model as security proof reference.

First, for a generator $g \in G$ and for an unknown $\alpha \in Z_p^*$, the author lets $y_i = g^{(\alpha^i)} \in G$. An algorithm C is given as input a random sample $(g, h, y_1, \dots, y_l, T)$ that is either sampled from P_{wBDHI^*} (where T is $e(g, h)^{(\alpha^{l+1})}$) or from R_{wBDHI^*} (where T is uniform and independent in G_1). The goal of algorithm C is to output 1 when the input sample is from P_{wBDHI^*} and 0 otherwise. At this point, algorithm C has to communicate with algorithm A in the following experiment where C could choose its desired identity to attack selectively and to see if the attack of C is successful or not.

Initialization: In this phase, A starts to simulate each step of above security experiment and chooses selectively identity $ID^* = (I_1^*, \dots, I_m^*) \in (Z_p^*)^m$, where $m \neq l$ and ID^* means that A desired to attack identity. If the length of identity ID^* is less than l , then algorithm C has to fill zeros into identity to make a l -length one. Then, it is possible to assume that identity ID^* is a l -length vector. As a result, this phase is completed.

Setup: In this phase, algorithm C has to generate the system parameters. It picks up a random number $\gamma \in Z_p^*$ and sets $g_1 = g, g_2 = y_l \cdot g^\gamma$, and $g_3 = g^\gamma$. Then, C selects $\gamma_1, \dots, \gamma_l$ in Z_p and lets $h_i = g^{\gamma_i} / y_{l-i+1}$ for $i=1, \dots, l$.

Finally, C transfers above parameters $(g_1, g_2, h_1, \dots, h_l)$ with g_3 to the attacker A . The master private key is $g_2^\alpha = g^{\alpha(\alpha^l + \gamma)} = y_{l+1} y_1^\gamma$ which algorithm C is not able to compute as y_{l+1} .

Phase One: In this phase, A could make private key queries for user i , where $i=1, \dots, l$. First, A prepares an identity set ID , where $ID = (I_1, \dots, I_t) \in (Z_p^*)^t$ and $t \leq l$. Then, it chooses a target ID^* to be attack one and others ID are not identified with ID or a prefix of ID^* . If A makes a private key query from (I_1, \dots, I_t) , C starts to generate the private key for identity $(I_1, \dots, I_k, \dots, I_l)$.

In order to generate each private key for identity (I_1, \dots, I_t) , C first chooses a random r_i , where $i=1, \dots, l$ and $t \leq l$. Then, C also computes $R_i = g^{r_i}$, where $i=1, \dots, l$. Finally, C forwards each private key $d_i = (g_2^{\alpha \cdot r_i}) = (y_{l+1}, y_1^{\gamma \cdot r_i})$ and $R_i = g^{r_i}$ to A , where $i=1, \dots, l$ and $r_i \in Z_p^*$.

Challenge: In this phase, A will decide that Phase One is over, it outputs two messages $(M_0, M_1) \in \mathbb{G}_1$ which are the challenged messages. Algorithm C picks a random bit $b \in \{0, 1\}$ and returns back the challenge cipher-text $CT_{\{j, i\}}$ in the following Equation 8:

$$CT_{\{j,i\}} = (M_b) \oplus \left[T \cdot e \left(R_i, \prod_{i=1}^q h_i \right) \cdot e \left(R_i, (y_l, g^\gamma)^{r_i} \right) \cdot e \left(g, h^{(\alpha^{l+1})} \right)^{-1} \cdot e(y_1, h^\gamma) \cdot e(y_1, h^\gamma)^{-1} \right] \quad (8)$$

where h and T are from input tuple given to C . At this stage, the author lets some part of $CT_{\{j,i\}}$ to be CT_3 , where:

$$CT_3 = e \left(g, h^{(\alpha^{l+1})} \right)^{-1} \cdot e(y_1, h^\gamma)^{-1} \cdot e(y_1, h^\gamma)$$

Notably, if $h = g^c$ (for some unknown $c \in Z_p^*$), then:

$$\begin{aligned} CT_3 &= e \left(g, h^{(\alpha^{l+1})} \right)^{-1} \cdot e(y_1, h^\gamma)^{-1} \cdot e(y_1, h^\gamma) = e \left(g^\alpha, g^{c \cdot \alpha^l} \right)^{-1} \\ &\cdot e(y_1, g^{c^\gamma})^{-1} \cdot e(y_1, g^{c^\gamma}) = e(y_l, y_l)^{-C} (y_1, g^\gamma)^{-C} \cdot e(y_1, g^{c^\gamma}) = e(y_l, y_l \cdot g^\gamma)^{-C} \\ &\cdot e(y_1, g^{c^\gamma}) = e(g_1, g_2)^{-C} \cdot e(g_1, g^\gamma)^c = e(g_1, g_2)^{-C} \cdot e(g_1, g_3)^c \end{aligned} \quad (9)$$

Equations 8 and 9 produce Equation 10, below:

$$\begin{aligned} CT_{\{j,i\}} &= (M_b) \oplus \left[T \cdot e \left(R_i, \prod_{i=1}^q h_i \right) \cdot e \left(R_i, (g_2)^{r_i} \right) \cdot CT_3 \right] \\ &= (M_b) \oplus \left[T \cdot e \left(R_i, \prod_{i=1}^q h_i \right) \cdot e \left(R_i, (g_2)^{r_i} \right) \cdot e(g_1, g_2)^{-C} \cdot e(g_1, g_3)^c \right] \end{aligned} \quad (10)$$

Therefore, if $T = e(g, h)^{(\alpha^{l+1})}$ (where the input tuple is sampled from P_{wBDHI^*}), then the challenge cipher-text is a valid encryption of M_b under the original (unpadded) identity $ID^* = (I_1^*, \dots, I_m^*)$ chosen by the adversary, since:

$$CT_{\{j,i\}} = (M_b) \oplus \left[e \left(R_j, \prod_{i=1}^q h_i \right) \cdot e \left(R_j, (g_2)^{r_i} \right) \right] \quad (11)$$

On the other hand, if T is a random and it is uniform and independent in G_1^* (the input tuple is sampled from R_{wBDHI^*}). $CT_{\{j,i\}}$ is independent of b in the adversary's view.

Phase Two: A could continue issue queries not issued in Phase One. Algorithm C also responds as before.

Guess: In this phase, A finally outputs a guess $b' \in \{0,1\}$. Then, C could conclude its own simulating games to output its guess as follows. If $b = b'$, then C outputs 1, which means $T = e(g, h)^{(\alpha' + 1)}$. Otherwise, it outputs 0, which means T is a random number in G_1 .

Thus, if the input tuple is sampled from the space P_{wBDHI}^* (where $T = e(g, h)^{(\alpha' + 1)}$), it means that A is able to mount real attack in the above games with C . Thus, A 's advantage is $|Pr[b=b'] - 1/2| \geq \epsilon$. On the other hand, if input tuple is sampled from R_{wBDHI}^* (where T is a random number in G_1^*), then the probability of A is $Pr[b=b'] = 1/2$. Finally, with given parameters (g, h) chosen uniformly in G , α uniformly in Z_p , and T uniformly in G_1 , the result is Equation 12:

$$\left| Pr \left[C \left(g, h, y_{g, \alpha, l}^-, e(g, h)^{(\alpha' + 1)} \right) = 1 \right] - Pr \left[C \left(g, h, y_{g, \alpha, l}^-, T \right) = 1 \right] \right| = \left| (1/2 + \epsilon) - 1/2 \right| = \epsilon \quad (12)$$

Ming-Te Chen was born in Tainan on August 2, 1980. He received his M. S. degree in computer science and information engineering from National Sun Yat-sen University, Taiwan, in 2005, and his Ph.D. degree in computer science and information engineering at National Sun Yatsen University, in 2012. In 2018, he joined the faculty of the department of computer science and information engineering, National Chin-Yi University Technology, Taichung, Taiwan. His current research interests include information security, applied cryptographic protocols, digital signature, IoT security, and electronic commerce.