Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.028895

ARTICLE

An Efficient and Provably Secure SM2 Key-Insulated Signature Scheme for Industrial Internet of Things

Senshan Ouyang[1,2], Xiang Liu[2], Lei Liu[2], Shangchao Wang[2], Baichuan Shao[3] and Yang Zhao[3,*]

[1]School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an, China

[2]Department of Process and Information Technology, Chengdu Aircraft Industrial (Group) Co., Ltd., Chengdu, China

[3]School of Information and Software Engineering, The Network and Data Security Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China, Chengdu, China

*Corresponding Author: Yang Zhao. Email: zhaoyang@uestc.edu.cn

Received: 14 January 2023 Accepted: 31 March 2023 Published: 22 September 2023

ABSTRACT

With the continuous expansion of the Industrial Internet of Things (IIoT), more and more organisations are placing large amounts of data in the cloud to reduce overheads. However, the channel between cloud servers and smart equipment is not trustworthy, so the issue of data authenticity needs to be addressed. The SM2 digital signature algorithm can provide an authentication mechanism for data to solve such problems. Unfortunately, it still suffers from the problem of key exposure. In order to address this concern, this study first introduces a key-insulated scheme, SM2-KI-SIGN, based on the SM2 algorithm. This scheme boasts strong key insulation and secure key-updates. Our scheme uses the elliptic curve algorithm, which is not only more efficient but also more suitable for IIoT-cloud environments. Finally, the security proof of SM2-KI-SIGN is given under the Elliptic Curve Discrete Logarithm (ECDL) assumption in the random oracle.

KEYWORDS

Key-insulated; SM2 algorithm; digital signature; Industrial Internet of Things (IIoT); provable security

1 Introduction

In recent years, the Industrial Internet of Things (IIoT), a core subset of the Internet of Things (IoT) [1,2], has seen rapid development and has brought substantial and sustainable advancement to industries [3]. The IIoT is a technology that connects sensors, smart devices and actuators to the existing "Internet" through Wireless Sensor Networks (WSNs) [4]. In IIoT environment, all smart devices can monitor, transmit, collect, and analyze information automatically. Apparently, compared to traditional industries, IIoT achieves more efficient and sustainable production, significantly reducing operating costs and resource consumption [5]. Consequently, the implementation of IIoT-centered smart industry plays a significant role in promoting the development of traditional manufacturing industry to smart manufacturing industry [6]. However, despite IIoT brings plenty of benefits, it also faces thorny data processing issues. Of particular concern is the huge amount of data that is monitored and collected by IIoT smart devices. How to store and process the big data raise serious

challenges [7]. Fortunately, cloud computing can provide us with a solution to appropriately deal with the aforementioned problems [8]. Cloud computing has broad network access and resource pooling, as well as formidable processing power and low cost advantages [9]. In IIoT-cloud computing environment, the challenges of big data collection, storage and processing can be properly solved [10].

Although the IIoT-cloud computing environment brings new ideas to solve the aforementioned problems, the authenticity and integrity of the data still need to be addressed urgently [5]. Generally, the channel between the cloud server and the smart device is considered undependable [11]. Therefore, ensuring that the authenticity of data is not maliciously intercepted and modified during transmission is a very difficult challenge. The digital signatures are a promising cryptographic primitive to address these challenges [12] (We give an example of a digital signature in Fig. 1). The data can be signed by the signer's private key before it is sent from the smart devices to the cloud server. The recipient, in turn, can verify the integrity of the message by verifying the signature [4]. Consequently, a series of public key infrastructure (PKI) signature protocols were progressively presented [13].

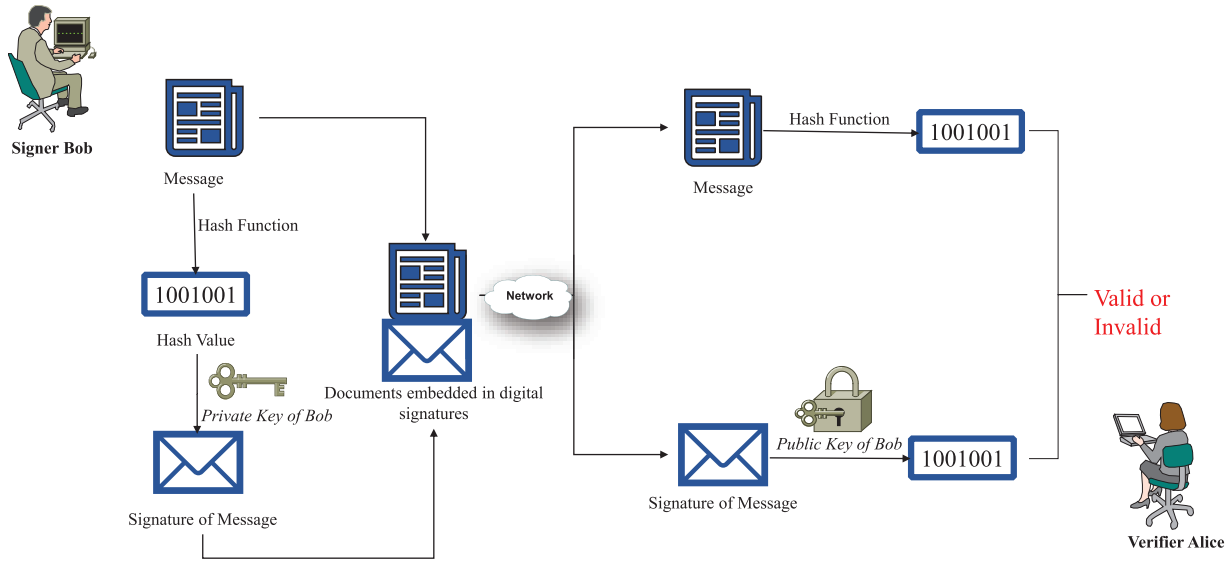

Figure 1: Digital signature

In a PKI-based digital signature system, a trusted certification authority (CA) binds a user's identity to a corresponding public key using an issued certificate. In 1976, the first digital signature scheme was proposed by Diffie et al. [14]. It is through the paper [14] that the foundation of Public Key Cryptography (PKC) has been established for the first time. In the next decades, Public Key Infrastructure (PKI) is a popularly applied authentication architecture in traditional PKC-based schemes. Based on the aforementioned knowledge, the U.S. government has released a federal information processing standard: Digital Signature Standard (DSS). And the Chinese government adopts RSA digital signature scheme.

With the development of cryptography and computer technology, the commonly used 1024-bit RSA algorithms are facing serious security threats. In 1987, the Elliptic Curve Cryptography (ECC), which performs better than traditional cryptosystem (such as RSA and DSA) in security and efficiency, was proposed for the first time [15]. On December 17 2010, the public key cryptographic algorithm SM2, published by the Chinese State Cryptography Administration Office in 2010 [16], is also an ECC. Noticeable, it has been standardized by ISO/IEC in ISO/IEC 14888-3:2016/DAMD 1

[17]. Since the algorithm is based on ECC, its signature speed and secret key generation speed are faster than RSA. Compared with RSA algorithm, 256-bit SM2 password strength is already higher than 2048-bit RSA password strength. In order to demonstrate the advantages of SM2 over RSA more intuitively, we have made a comparison between the two dimensions of security and speed. The comparison results are listed in Tables 1 and 2. Thus, SM2 has better performance and security: high password complexity, fast processing speed, and less machine performance consumption. Now, SM2 algorithm is already widely executed in lots of fields, such as electronic authentication systems, E-Commerce systems and E-Government systems.

Table 1: The comparison of security between SM2 and RSA

RSA key strength (Length)	SM2 key strength (Length)	Password cracking time
521 bits	106 bits	104 years (cracked)
768 bits	132 bits	108 years (cracked)
1024 bits	160 bits	1011 years
2048 bits	210 bits	1020 years

Table 2: The comparison of speed between SM2 and RSA

Signature algorithm	Signature speed	Verification speed
1024-bit RSA	2792 times per second	51224 times per second
2048-bit RSA	455 times per second	15122 times per second
256-bit SM2	4095 times per second	871 times per second

Another inevitable thorny problem is the key exposure problem since the signature operations are often executed frequently on insecure smart devices. It is obvious that key exposure will lead to disastrous consequences. The primitive of key-insulated was given by Dodis et al. in 2002 [18] for the first time. This cryptographic primitive effectively deals with the problem of catastrophic key exposure. The signer's temporary signing key completes the key evolution with the assistance of the helper. Without the helper providing an update message, the signer's key cannot be updated from the last time period to the current time period. With the helper's key secure, an adversary can only forge the signature scheme for the current time period rather than the next one. After that, A strong key-insulated signature scheme was proposed by Dodis et al. [19]. Then, a number of well-designed key-insulated schemes were gradually constructed based on the work of Hanaoka et al. [20–22]. It is worth noting that the scheme proposed by Zhou et al. [22] does not have the nature of strong key-insulated. This means that an adversary can forge a signature as a legitimate user if the helper's key is cracked. Therefore, Weng et al. [23] proposed a promising idea, namely secure key-updates. At present, this idea has been widely applied.

Given the above analysis, it faces the key exposure issue when the SM2 digital signature algorithm is integrated into the IIoT-cloud computing environment. This problem has attracted widespread attention from domestic and international authors [24,25]. In order to address the thorny issue of key exposure mentioned above, an efficient and provable secure key-insulated signature scheme based on SM2 (SM2-KI-SIGN) is proposed by us in the IIoT-cloud environment now. Our scheme is inspired by the idea of secure key-updates [23]. Our scheme also has the properties of strong key-insulated and

secure key-updates. However, it is more efficient than the Weng et al. [23] due to the use of Elliptic Curve Cryptography (ECC).

Our core contributions in this paper are as follows:

1) Introduction of an efficient and secure key-insulated signature scheme based on the SM2 cryptosystem, termed SM2-KI-SIGN;
2) Demonstration that SM2-KI-SIGN achieves EUF-CMA (existential unforgeability under chosen message attacks) and has the key-insulated property, thereby efficiently mitigating the key exposure issue;
3) Empirical validation of the efficiency and applicability of SM2-KI-SIGN through specific experimental simulations and performance assessments.

The organization is illustrated in this paragraph. In Section 2, we demonstrate some corresponding preliminaries such as elliptic curve, security assumption, and system framework. In Section 3, the concrete construction of SM2-KI-SIGN is provided. In Section 4, the associated security proof, the theoretical as well as experiment evaluation is demonstrated. Finally, Section 5 gives a summary of this paper.

2 Preliminaries

2.1 Elliptic Curve Discrete Logarithm (ECDL) Problem

Set E ($\mathbb{F}_q$) as an elliptic over $\mathbb{F}_q$ where $G \in E(\mathbb{F}_q)$.There are two points $P, Q \in E(\mathbb{F}_q)$ of order q. Besides Q is a multiplicity of points of P. If there exists a positive integer $l \in [0, q-1]$ that makes $Q = l \cdot P$, then obtaining the value of l from P and Q is the ECDL problem.

2.2 ECDLP Assumption

There is a P.P.T algorithm $\mathscr{A}$ has advantage at least ε to solve ECDL problem in $E(\mathbb{F}_q)$.

$Pr[A(P, Q) = l | Q = l \cdot P, l \in \mathbb{Z}_q^*] \geqslant \varepsilon$

2.3 Bilinear Pairings

Let $\mathbb{G}$ be an addictive group and $\mathbb{G}_T$ be a multiplicative group. $\mathbb{G}$ and $\mathbb{G}_T$ has the equivalent prime order q. P is one of the generators of $\mathbb{G}$. The bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ satisfies the below properties:

1) Bilinearity: $\forall m, n \in \mathbb{Z}_q^*$, $e : (mP, nP) = e : (P, P)^{mn}$.
2) Non-degeneracy: $e : (P, P) \neq 1$.
3) Computability: There exists an algorithm to calculate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

2.4 Elliptic Curve Cryptography

In recent decades, Elliptic Curve Cryptography (ECC) has been widely studied and applied. In 1985, a mathematician named Victor Miller studied elliptic curves in cryptography and hypothesised that it was highly unlikely that exponential calculus methods would work for elliptic curves. ECC is a public key cryptography method based on the algebraic structure of elliptic curves over a finite field, allowing the use of smaller keys to provide equivalent security. Elliptic curves have now been applied to tasks such as key negotiation, digital signatures, pseudo-random generators. ECC utilises smaller keys, which reduces storage and transmission consumption. Thus, ECC can be better adapted to the IIoT-cloud environment.

2.5 Notations

The notations presented in the SM2-KI-SIGN scheme are defined in Table 3.

Table 3: Notations

Acronym	Description
d	The private key
P	Public key
t_i	Time period index
X_i	Time period function
T_i	Temporary key in time period t_i
hk	Private key for helper
HK	Public key for helper
$\sigma = (r, s, \phi)$	Signature
G	Generator of $E(\mathbb{F}_p)$
H_1, H_2, H_3	Three cryptographic hash functions
$PSK_{i,j}$	Partial temporary key
$ENTL_{ID}$	Length of a signer's ID

2.6 Outline of SM2-KI-SIGN

The SM2-KI-SIGN scheme consists of six different algorithms described below:

1) Setup: Input the security parameter k, the KGC produces params.
2) KeyGen: Given params, time period t, the user generates the public and private key (d, P) for him/her own as well as generates the public and private key for the helper (hk, HK).
3) Upd*: Input params, time period t_i and t_j, the helper output the partial temporary key $PSKi, j$.
4) Upd: Input params, t_i, T_j, and $PSK_{i,j}$, the helper output T_i.
5) Sign: Input the params, t_i, T_i, and the message m, a signer generate a signature ϕ on m.
6) Verify: Input the params, P, HK, and a message-signature pair (m, ϕ), a verifier output 1 when the signature is valid.

3 Our Proposed SM2-KI-SIGN Scheme

In this section, we further elaborate the detailed construction of SM2-KI-SIGN digital signature scheme we proposed. This scheme consists of six different algorithms as listed below. In these algorithms, Upd* and Upd are mainly designed for address the problem of key exposure. The flow of interaction between entities in the SM2-KI-SIGN is illustrated in Fig. 2.

1. Setup: Input the security parameter k, the administrator operates as follows:
   - Generate an elliptic curve $y^2 = x^3 + ax + b$ over a finite field $\mathbb{F}_p$ as well as the discriminant $\Delta = 4a^3 + 27b^2 \neq 0$. (p, a, b, q) are the parameters of the curve, where p and q are two large prime numbers. p is the size of $\mathbb{F}_p$.
   - Select $G \in_R E(\mathbb{F}_p)$ as one of the generators. Besides let q be the order of G.
   - Set the public parameters params $= (p, a, b, q, G)$ and then output it.

- Select three cryptographic hash functions H_1, H_2, H_3 and describe them with details here: $H_1 : \{0,1\}^* \to E(\mathbb{F}_p)$, $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$, and $H_3 : \{0,1\}^* \to \{0,1\}^{256}$.

2. KeyGen: Input params, the user operates as follows:
   - Select $d \in_R \mathbb{Z}_q^*$ as the private key.
   - Calculate $P = d \cdot G$ and set P as the public key.
   - Output the pair of the private and public key (d, P).
   - Given the time period t_0, the helper for the user executes as follows.
   - Select $hk \in_R \mathbb{Z}_q^*$ as the private key for the helper.
   - Calculate the public key for the helper $HK = hk \cdot G$.
   - Output (hk, HK).
   - Calculate initial time period key $T_0 = hk \cdot X_0$ and time period function $X_0 = H_1(t_0)$.
3. Upd*: Input two time period indices t_i and t_j, the helper for the user executes as below:
   - Calculate $X_{i,j} = H_1(t_i) - H_1(t_j)$.
   - Calculate the partial temporary key $PSK_{i,j} = hk \cdot L_{i,j}$.
   - Return $PSK_{i,j}$.
4. Upd: Input a time period index t_i, the partial temporary key $PSK_{i,j}$ and the temporary key T_j, the signer obtains the temporary key for the time period t_i as below:
   - Set $T_i = T_j + PSK_{i,j}$.
   - Return the temporary key T_i.
5. Sign: Input params, the message m to be signed, time period index t_i, as well as the private key d, the signer operates as follows:
   - Calculate $Z = H_3(ENTL_{ID} \| ID \| a \| b \| G \| x \| y)$. $ENTL_{ID}$ denotes the length of a signer's ID.
   - Calculate $e = H_2(\bar{m})$, where $\bar{m} = Z \| m$.
   - Select $k \in_R \mathbb{Z}_q^*$, then calculate $K = k \cdot G$.
   - Calculate $K' = K + k \cdot T_i = (x_1, y_1)$, $r = x_1 + e \bmod q$.
   - Calculate $s = (1 + d)^{-1} \cdot (k - r \cdot d) \bmod q$.
   - Calculate $\phi = (1 + hk)^{-1} \cdot (k - r \cdot hk) \bmod q$.
   - Output the signature $\sigma = (r, s, \phi)$.
6. Verify: Input params, the public key $P = d \cdot G$, the public key of helper $HK = hk \cdot G$, the message m as well as the related signature σ, and then the verifier operates as below:
   - Calculate $Z = H_2(ENTL_{ID} \| ID \| a \| b \| G \| x \| y)$. The definition of $ENTL_{ID}$ is the same as the aforementioned one.
   - If $r \notin \mathbb{Z}_q^*$, the verification fails and then terminate the algorithm.
   - If $s \notin \mathbb{Z}_q^*$, the verification fails and then terminate the algorithm.
   - Set $\bar{m} = Z \| m$, and calculate $e = H_2(\bar{m})$.
   - Calculate $t = (r + s) \bmod q$. If $t = 0$, the verification fails and terminate the algorithm.
   - Calculate $\psi = (r + \phi) \bmod q$. If $\psi = 0$, the verification fails and terminate the algorithm.
   - Calculate $(x_1, y_1) = s \cdot G + t \cdot P + \phi \cdot X_i + \psi \cdot T_i$.
   - Calculate $R = (e + x_1) \bmod n$, if $R = r$, the signature is valid and the verification passes, otherwise the verification fails.

7. Correctness

$$
\begin{aligned}
&(x_1, y_1) = s \cdot G + t \cdot P + \phi \cdot X_i + \psi \cdot T_i \\
&= s \cdot G + (r + s) \cdot P + \phi \cdot X_i + (r + \phi) \cdot T_i \\
&= s \cdot G + (r + s) \cdot d \cdot G + \phi \cdot X_i + (r + \phi) \cdot hk \cdot L_i \\
&= (1 + d) \cdot s \cdot G + r \cdot d \cdot G \\
&+ (1 + hk) \cdot \phi \cdot X_i + r \cdot hk \cdot X_i \\
&= (1 + d) \cdot (1 + d)^{-1} \cdot (k - r \cdot d) \cdot G + r \cdot d \cdot G \\
&+ (1 + hk) \cdot (1 + hk)^{-1} \cdot (k - r \cdot hk) \cdot X_i + r \cdot hk \cdot X_i \\
&= (k - r \cdot d) \cdot G + r \cdot d \cdot G \\
&+ (k - r \cdot hk) \cdot X_i + r \cdot hk \cdot X_i \\
&= k \cdot G + k \cdot T_i
\end{aligned}
$$

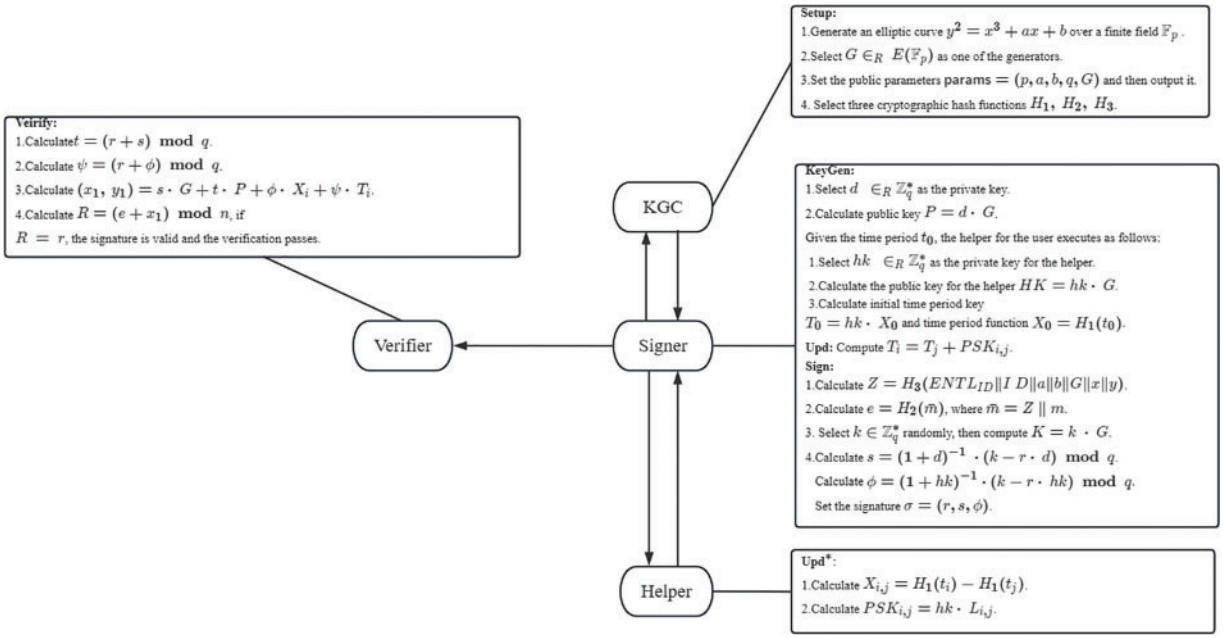

Figure 2: Process of SM2-KI-SIGN scheme

4 Analysis

4.1 Security Proof

1) **Theorem 1.** The SM2-KI-SIGN scheme we proposed is perfectly key-insulated against a P.P.T adversary $\mathscr{A}$ in *Game*.

Proof: Given an ECDL problem instance (P, P_0), $\mathscr{B}$ computes $a \in_R \mathbb{Z}_q^*$, such that $P_0 = a \cdot P$, where P is G and $\mathscr{B}$ controls the stochastic prediction machine.

Setup: First, $\mathscr{B}$ initializes $\mathscr{A}$ with $P_{KGC} = P_0$, then it sends the public parameters params $= (p, a, b, q, G)$ and (P, P_{KGC}) to $\mathscr{A}$.

Query : The interaction process between adversary $\mathscr{A}$ and $\mathscr{B}$ is as follows. $\mathscr{A}$ can execute queries adaptively.

1) H_1 query: $\mathscr{B}$ manages the list L_1 with the tuple (t_i, X_i). After $\mathscr{A}$ delivered the (t_i, X_i) query to the $H_1()$ oracle, $\mathscr{B}$ retrieves the list L_1 at the beginning. If L_1 includes (t_i, X_i), $\mathscr{B}$ answers to $\mathscr{A}$ with X_i. Otherwise, $\mathscr{B}$ selects $X_i \in_R \mathbb{Z}_q^*$, returns X_i to $\mathscr{A}$ and inserts the tuple (t_i, X_i) into L_1.
2) H_2 query: $\mathscr{B}$ manages the list L_2 with the tuple $(e, \bar{m})$. After $\mathscr{A}$ delivered the $(e, \bar{m})$ query to the $H_2()$ oracle, $\mathscr{B}$ retrieves the list L_2 at the beginning. If L_2 includes $(e, \bar{m})$, $\mathscr{B}$ answers to $\mathscr{A}$ with e. Otherwise, $\mathscr{B}$ selects $e \in_R \mathbb{Z}_q^*$, returns e to $\mathscr{A}$ and inserts the tuple $(e, \bar{m})$ into L_2.
3) H_3 query: $\mathscr{B}$ manages the list L_3 with the tuple (ID, Z). After $\mathscr{A}$ delivered the (ID, Z) query to the $H_3()$ oracle, $\mathscr{B}$ retrieves the list L_3 at the beginning. If L_3 includes (ID, Z), $\mathscr{B}$ answers to $\mathscr{A}$ with Z. Otherwise, $\mathscr{B}$ selects $Z \in_R \mathbb{Z}_q^*$, returns Z to $\mathscr{A}$ and inserts a tuple (ID, Z) into L_3.
4) Extract-Private-Key: $\mathscr{B}$ manages the list L_{pri} with the tuple $(ID, d, hk, PSK_{i,j})$. After the identity ID is delivered to the oracle, then $\mathscr{B}$ retrieves the list L_{pri}. If $ID_i = ID_I$, then $\mathscr{B}$ terminate the simulation (Event E_1). Otherwise L_{pri} includes $(ID, d, hk, PSK_{i,j})$, $\mathscr{B}$ gives $\mathscr{A}$ answers with $(d, hk, PSK_{i,j})$; If L_{pri} does not include $(ID, d, hk, PSK_{i,j})$, $\mathscr{B}$ chooses d_i, $hk_i \in \mathbb{Z}_q^*$ randomly, and computes $PSK'_{i,j} = hk_i \cdot H_{i,j}$. Then $\mathscr{B}$ inserts the tuple $(ID, d_i, hk_i, PSK'_{i,j})$ into L_{pri}. Lastly, $\mathscr{B}$ answers to $\mathscr{A}$ with $(d_i, hk_i, PSK'_{i,j})$.
5) Extract-Public-Key: $\mathscr{B}$ manages the list L_{pub} with the tuple (ID, P, HK). After the identity ID is provided to this oracle, $\mathscr{B}$ retrieves the list L_{pub}. If L_{pub} includes (ID, P, HK), $\mathscr{B}$ answers to $\mathscr{A}$ with (ID, P, HK). Otherwise L_{pub} does not include (ID, P, HK), $\mathscr{B}$ makes queries to L_{par}, L_{pri} and compute $P = d \cdot G$ and $HK = hk \cdot G$ as well as inserts the tuple (P, HK) into the L_{pub}. Lastly, $\mathscr{B}$ answers to $\mathscr{A}$ with (ID, P, HK).
6) Public-Key-Replace: After $\mathscr{A}$ makes a query of (ID, P', HK'), $\mathscr{B}$ retrieves the list L_{pub}. If L_{pub} does not include (ID, P, HK), $\mathscr{B}$ first does a Extract-Public-Key query with identity ID, and then, sets $P = P'$, $HK = HK'$. To respond the query, $\mathscr{B}$ will update the list L_{pub} with (P, HK).
7) Signature query: After $\mathscr{A}$ makes a query of (ID, M'), $\mathscr{B}$ $\mathscr{B}$ picks a number $a \in \mathbb{Z}_q^*$ at random, and sets $hk = a$, $\phi = (1 + hk)^{-1} \cdot (k - r \cdot hk) \bmod q$. After that, $\mathscr{B}$ returns a valid signature θ to $\mathscr{A}$.

   Forgery: After polynomially bounded queries, $\mathscr{A}$ forges a signature $\sigma = (r_1, s_1, \phi_1)$ on message (ID^*, M) with non-negligible probability ε. If $ID \neq ID_i \neq ID_I$, the challenge of $\mathscr{B}$ fails and stops (event E_2); otherwise, the forgery succeeds. Then, depending on the forking lemma, $\mathscr{A}$ repeats the aforementioned query using different hash values, two more signature pairs (r_2, s_2, ϕ_2) and (r_3, s_3, ϕ_3) can be generated.

   $(x_1, y_1) = s_j \cdot G + t_j \cdot P + \phi_j \cdot X_I + \psi_j \cdot T_I, j = 1, 2, 3$

   Set $(x_1, y_1) = c \cdot G + c \cdot T_i$. Because $P_0 = a \cdot P \cdot G$, $T_i' = v \cdot P \cdot X_i$, we can obtain $c = s_j + a \cdot P \cdot G + \phi_j + v \cdot P \cdot X_i$.

   There are three unknown numbers c, a, v that are linearly independent of each other. Combining the three equations can find the value of a. $\mathscr{B}$ successfully solves an *ECDLP* instance using the capabilities of $\mathscr{A}$. To forge a pair of signatures successfully, the following three events need to be satisfied:

1). π_1 represents that no partial private key query has been performed on it, i.e., the event E_1 does not occur, $Pr[\pi_1] \geqslant \left(1 - \frac{1}{q_{H_1}}\right)^{q_{ppk}}$.

2). π_2 The signature forgery under the message M^* is valid.

3). π_3 The forged signature is subject to ID-consistency, i.e., the event E_2 does not occur, $Pr[\pi_3|\pi_1 \wedge \pi_2] \geqslant \frac{1}{q_{H_1}}$.

Thus, $\mathscr{B}$ uses the ability of $\mathscr{A}$ in polynomial time with non-negligible probability $\varepsilon' = Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] = Pr[\pi_1] \cdot Pr[\pi_2|\pi_2] \cdot Pr[\pi_3 \wedge \pi_2 \wedge \pi_1] \geqslant \left(1 - \frac{1}{q_{H_1}}\right)^{q_{ppk}} \cdot \varepsilon \cdot \frac{1}{q_{H_1}}$ successfully solves an *ECDLP* instance, which contradicts the ECDLP's difficulty contradiction, so the scheme is able to resist the attacker's $\mathscr{A}$ adaptive selection existential forgery under the choice message attack.

2) **Theorem 2.** The proposed SM2-KI-SIGN is strong key-insulated secure against adversary $\mathscr{B}$.

Proof: The adversary $\mathscr{B}$ has the non-negligible probability $\varepsilon' \geqslant \left(1 - \frac{1}{q_{H_1}}\right)^{q_{ppk}} \cdot \varepsilon \cdot \frac{1}{q_{H_1}}$.

The proof is same as those of **Theorem 1**, so we omit the proof here.

3) **Theorem 3.** The SM2-KI-SIGN scheme we proposes in this paper has secure key updates.

Proof: As to any period indices t_i and t_j, the update key $PSK_{i,j}$ can be evolved from T_i and T_j.

4) **Theorem 4.** The proposed SM2-KI-SIGN is secure against EUF-CMA.

Proof: At first, assume that a P.P.T adversary $\mathscr{A}$ can exchange information with the signer. Thus, L, r and s, ϕ can be viewed by $\mathscr{A}$ in the key-insulated signature generating step because of $s = (1 + d)^{-1} \cdot (k - r \cdot d) \bmod q$ and $\phi = (1 + hk)^{-1} \cdot (k - r \cdot hk) \bmod q$. $\mathscr{A}$ obtains the value of r'. If $\mathscr{A}$ wants to obtain d and hk from s and ϕ, he/she must get the value of k. Although $\mathscr{A}$ knows $L = k \cdot G + k \cdot T_i$, it is a *ECDLP* to calculate k from K. If *ECDLP* is difficult to solve, then the private key cannot be received by $\mathscr{A}$ when he/she exchanges information with the signer. In our proposed SM2-KI-SIGN signature scheme, the signing and verification equations we designed are consistent with the SM2 digital signature scheme. The SM2-KI-SIGN key-insulated signature scheme we proposed is unforgeable under the EUF-CMA attack, since the SM2 signature scheme satisfies EUF-CMA.

4.2 Performance Comparison

To certify the efficiency and feasibility of the proposed SM2-KI-SIGN scheme, we compare it with the existing works in this subsection. The comparison results are demonstrated in figures and tables.

In Table 4, we summarise and compare the properties between SM2-KI-SIGN scheme and other relevant schemes. We compare the existing schemes from three dimensions: strong key-insulated, secure key-updates and security assumption in Table 4. Here, it should be noted that the symbol "✓" indicates that the scheme satisfies this corresponding property, as well as the symbol "×" means that this capability cannot be achieved by this scheme. Apparently, our proposed SM2-KI-SIGN scheme can satisfy all properties. And this can be proven secure under standard *ECDLP* assumptions which is weaker than other security assumptions.

Table 4: The comparison of properties

Scheme	Strong key-insulated	Secure key-updates	Security assumption
[26]	×	×	EBSDH&BSDH
[27]	×	×	q-mBDHI&CDH&q-CAA
[28]	×	×	ECDLP
[29]	✓	✓	GDH
Ours	✓	✓	ECDLP

Then, a simulation experiment that runs on a Windows 10 computer equipped with an Intel Core i7-6700@2.60-GHz processor, as well as 8 GB, is given in this section. Then, it is implemented in IDEA with Java pairing-based cryptography (JPBC) library. To achieve the same security level as 1024-bit RSA, the super-singular curve $y^2 = x^3 + x \pmod{p}$ with an embedding degree of 2 is utilized, where $q = 2^{159} + 2^{17} + 1$ is a 160-bit Solinas prime and $p = 12q \cdot r - 1$ is a 512-bit prime. As to the ECC-based scheme, in order to offer the security with the equivalent level, we used the Koblitz elliptic curve $y^2 = x^3 + a \cdot x + b$ defined on $\mathbb{F}_{2163}$ providing the ECC group. In Table 5, a theoretical evaluation of the signature length, signing cost, as well as verification cost is given. Besides the notations of required signature length and cost of signing and verification are also enumerated in the footnote of Table 5.

Table 5: The performance comparison of different schemes

Scheme	Signature length	Signing cost	Verification cost
[26]	$1\|G_1\| + 1\|G_2\|$	$2T_{G_1} + 1T_{G_2}$	$2T_{G_1} + 1T_p$
[27]	$3\|G_1\| + 1\|G_2\|$	$3T_{G_1} + 1T_{G_2} + 1T_p$	$2T_{G_1} + 2T_p + 1T_{G_2}$
[28]	$2\|G_1\|$	$3T_{G_1}$	$2T_{G_1} + 2T_p$
[29]	$2\|G_1\| + 1\|G_2\|$	$2T_{G_1} + 1T_{G_2}$	$1T_{G_1} + 2T_p$
Ours	$3\|Z_q^*\|$	$4T_m$	$3T_m$

Note: $|G_1|$: size of a point in G_1, $|G_2|$: size of a point in G_2, $|Z_q^*|$: bit length in Z_q^*, T_{G_1}: exponentiation in G_1, T_{G_2}: exponentiation in G_2, T_p: pairing operation, T_m: scalar multiplication.

Compared with the existing schemes especially the schemes listed here, our scheme has more advantages in cost. This advantage makes SM2-KI-SIGN scheme more suitable for untrusted channels in IIoT-cloud computing environment. At the same time, we show a cost comparison of SM2-KI-SIGN with other schemes [26–29] in Fig. 3.

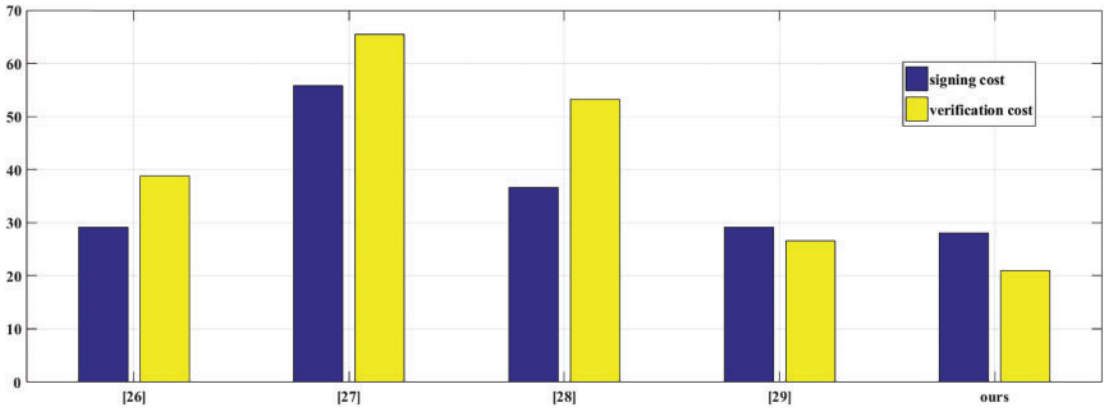

Figure 3: Comparison of cost

5 Conclusions

This paper presented the first key-insulated digital signature scheme SM2-KI-SIGN based on the SM2 algorithm. The proposed SM2-KI-SIGN scheme can effectively reduce the risk of key exposure due to untrusted channels in IIoT-cloud computing environment. We first gave a formal outline of the scheme. Following this, a concrete scheme and the formal security proof under the *ECDLP* assumption in the random oracle model were given. Finally, according to the theoretical analysis and simulation experiments, the SM2-KI-SIGN scheme is more efficient and practical than other related key-insulated works. In the current research field, SM2-KI-SIGN introduces a method to make up for the key exposure defects of existing SM2 signature algorithms. On the other hand, our work can provide a new idea for future commercial digital signature schemes.

Acknowledgement: We have already revised the Acknowledgement section in the manuscript.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China (Nos. 62072074, 62076054, 62027827, 62002047), the Sichuan Science and Technology Innovation Platform and Talent Plan (Nos. 2020JDJQ0020, 2022JDJQ0039), the Sichuan Science and Technology Support Plan (Nos. 2020YFSY0010, 2022YFQ0045, 2022YFS0220, 2023YFG0148, 2021YFG0131), the YIBIN Science and Technology Support Plan (No. 2021CG003), the Medico-Engineering Cooperation Funds from University of Electronic Science and Technology of China (Nos. ZYGX2021YGLH212, ZYGX2022YGRH012).

Author Contributions: study conception and design: Senshan Ouyang, Baichuan Shao and Yang Zhao; analysis and interpretation of results: Xiang Liu, Lei Liu, Shangchao Wang; draft manuscript preparation: Senshan Ouyang and Baichuan Shao; figures and tables production: Baichuan Shao.

Availability of Data and Materials: Our current research is limited to algorithm design and analysis, and has not yet applied it to practical scenarios, so we have not yet addressed the source and use of data and materials.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Chen, J., Liu, G., Liu, Y. (2020). Lightweight privacy-preserving raw data publishing scheme. *IEEE Transactions on Emerging Topics in Computing, 9(4),* 2170–2174.
2. Chen, J., Liu, Y., Xiang, Y., Sood, K. (2021). RPPTD: Robust privacy-preserving truth discovery scheme. *IEEE Systems Journal, 16(3),* 4525–4531.
3. Boyes, H., Hallaq, B., Cunningham, J., Watson, T. (2018). The Industrial Internet of Things (IIoT): An Analysis Framework. *Computers in Industry, 101(8),* 1–12.
4. Xiong, H., Mei, Q., Zhao, Y. (2019). Efficient and provably secure certificateless parallel key-insulated signature without pairing for IIoT environments. *IEEE Systems Journal, 14(1),* 310–320.
5. 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.
6. Yu, K., Tan, L., Aloqaily, M., Yang, H., Jararweh, Y. (2021). Blockchain-enhanced data sharing with traceable and direct revocation in IIoT. *IEEE Transactions on Industrial Informatics, 17(11),* 7669–7678.
7. Sadeghi, A. R., Wachsmann, C., Waidner, M. (2015). Security and privacy challenges in industrial Internet of Things. *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, IEEE.
8. Xiong, H., Zhao, Y., Hou, Y., Huang, X., Jin, C. et al. (2020). Heterogeneous signcryption with equality test for IIoT environment. *IEEE Internet of Things Journal, 8(21),* 16142–16152.
9. Kumar, M., Sharma, S. C., Goel, A., Singh, S. P. (2019). A comprehensive survey for scheduling techniques in cloud computing. *Journal of Network and Computer Applications, 143(2),* 1–33.
10. Li, Q., Yue, Y., Wang, Z. (2020). Deep robust cramer shoup delay optimized fully homomorphic for IIoT secured transmission in cloud computing. *Computer Communications, 161(10),* 10–18.
11. Hou, Y., Xiong, H., Huang, X., Kumari, S. (2021). Certificate-based parallel key-insulated aggregate signature against fully chosen key attacks for Industrial Internet of Things. *IEEE Internet of Things Journal, 8(11),* 8935–8948.
12. Wang, W., Xu, H., Alazab, M., Gadekallu, T. R., Han, Z. et al. (2021). Blockchain-based reliable and efficient certificateless signature for IIoT devices. *IEEE Transactions on Industrial Informatics, 18(10),* 7059–7067.
13. ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory, 31(4),* 469–472.
14. Diffie, W., Hellman, M. E. (2022). New directions in cryptography. In: *Democratizing cryptography: The work of whitfield diffie and martin hellman*, pp. 365–390.
15. Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation, 48(177),* 203–209.
16. Administration, S. C. (2010). Public key cryptographic algorithm SM2 based on elliptic curves-Part 2: Digital signature algorithm. http://www.sca.gov.cn/sca/xwdt/2010-12/17/1002386/files/b791a9f908bb4803875ab6aeeb7b4e03.pdf
17. Administration, S. C. (2016). Our SM2 and SM9 digital signature algorithms officially become ISO/IEC international standard. http://www.sca.gov.cn/sca/qjd/2017-11/17/content_1019960.shtml
18. Dodis, Y., Katz, J., Xu, S., Yung, M. (2002). Key-insulated public key cryptosystems. In: *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 65–82. Springer.
19. Dodis, Y., Katz, J., Xu, S., Yung, M. (2003). Strong key-insulated signature schemes. *International Workshop on Public Key Cryptography*, Springer.
20. Hanaoka, G., Hanaoka, Y., Imai, H. (2006). Parallel key-insulated public key encryption. *International workshop on public key cryptography*, pp. 105–122. Springer.
21. Hanaoka, Y., Hanaoka, G., Shikata, J., Imai, H. (2005). Identity-based hierarchical strongly key-insulated encryption and its application. *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 495–514. Springer.

22. Zhou, Y., Cao, Z., Chai, Z. (2006). Identity based key insulated signature. *International Conference on Information Security Practice and Experience*, pp. 226–234. Springer.
23. Weng, J., Liu, S., Chen, K., Li, X. (2006). Identity-based key-insulated signature with secure key-updates. *International Conference on Information Security and Cryptology*, pp. 13–26. Springer.
24. Hou, H. X., Yang, B., Zhang, L. N., Zhang, M. R. (2020). Secure two-party SM2 signature algorithm. *Acta Electonica Sinica, 48(1),* 1–8.
25. Zhang, Y., He, D., Zhang, F., Huang, X., Li, D. (2020). An efficient blind signature scheme based on SM2 signature algorithm. *International Conference on Information Security and Cryptology*, pp. 368–384. Springer.
26. Karati, A., Islam, S. H., Karuppiah, M. (2018). Provably secure and lightweight certificateless signature scheme for IIoT environments. *IEEE Transactions on Industrial Informatics, 14(8),* 3701–3711.
27. Chen, J., Wang, L., Wen, M., Zhang, K., Chen, K. (2021). Efficient certificateless online/offline signcryption scheme for edge IoT devices. *IEEE Internet of Things Journal, 9(11),* 8967–8979.
28. Cha, S. C., Chen, J. F., Su, C., Yeh, K. H. (2018). A blockchain connected gateway for BLE-based devices in the internet of things. *IEEE Access, 6,* 24639–24649.
29. Reddy, P. V., Gopal, P. (2017). Identity-based key-insulated aggregate signature scheme. *Journal of King Saud University-Computer and Information Sciences, 29(3),* 303–310.