# Security of the SM2 Signature Scheme Against Generalized Key Substitution Attacks

Zhenfeng Zhang[1], Kang Yang[1(✉)], Jiang Zhang[2], and Cheng Chen[1]

[1] Laboratory of Trusted Computing and Information Assurance, Institute of Software, Chinese Academy of Sciences, Beijing, China
{zfzhang,yangkang,chencheng}@tca.iscas.ac.cn
[2] State Key Laboratory of Cryptology, Beijing, China
jiangzhang09@gmail.com

**Abstract.** Though existential unforgeability under adaptively chosen-message attacks is well-accepted for the security of digital signature schemes, the security against key substitution attacks is also of interest, and has been considered for several practical digital signature schemes such as DSA and ECDSA. In this paper, we consider *generalized* key substitution attacks where the base element is considered as a part of the public key and can be substituted. We first show that the general framework of certificate-based signature schemes defined in ISO/IEC 14888-3 is vulnerable to a *generalized* key substitution attack. We then prove that the Chinese standard SM2 signature scheme is existentially unforgeable against adaptively chosen-message attacks in the generic group model if the underlying hash function $h$ is uniform and collision-resistant and the underlying conversion function $f$ is almost-invertible, and the SM2 digital signature scheme is secure against the *generalized* key substitution attacks if the underlying hash functions $H$ and $h$ are modeled as *non-programmable* random oracles (NPROs).

**Keywords:** Digital signatures · Key substitution attacks · Provable security

## 1 Introduction

The well-known security notion for digital signature schemes is existential unforgeability under adaptively chosen-message attacks (EUF-CMA) introduced by Goldwasser et al. [8,9], which states that an adversary given any signatures for messages of its choice is unable to create a valid signature for a new message. However, as a de facto standard security notion for signature schemes, the security notion of EUF-CMA fails to capture the duplicate-signature key selection attacks introduced by Blake-Wilson and Menezes [3]. Later, this type of attacks is called key substitution (KS) attacks by Menezes and Smart [12],

The work was supported by National Basic Research Program of China (No. 2013CB338003), and National Natural Science Foundation of China (No. 61170278).


L. Chen and S. Matsuo (Eds.): SSR 2015, LNCS 9497, pp. 140–153, 2015.
DOI: 10.1007/978-3-319-27152-1_7

when they investigated the security of signature schemes in a multi-user setting. The KS attacks for some EUF-CMA secure signature schemes can be found in [3,7,12,15]. Informally, a KS-adversary is given a public key $pk$ and a valid message-signature pair $(m, \sigma)$ under the public key $pk$, and attempts to produce another public key $pk'$ such that the message-signature pair $(m, \sigma)$ is still valid under the different public key $pk' \neq pk$. In [3], Blake-Wilson and Menezes showed that if the underlying signature scheme suffers from the KS attacks, the station-to-station (STS) key agreement protocol [6] using a message authentication code (MAC) algorithm to provide key confirmation is vulnerable to unknown key-share (UKS) attacks. This gives a direct evidence that KS attacks might be harmful in practice. Two types of key substitution attacks are considered by Menezes and Smart [12]: if the KS-adversary is further required to output the private key $sk'$ corresponding to $pk'$, then this kind of KS attacks are called the weak-key substitution (WKS) attacks, else this type of attacks are referred to as the strong-key substitution (SKS) attacks. Obviously, a signature scheme which is secure against SKS attacks is also secure against WKS attacks. Afterwards, Bohli, Rohrich and Steinwandt [4] explored the security of some practical signature schemes against key substitution attacks in the presence of a *malicious* signer, where an adversary is given a set of domain parameters $params$, and aims at outputting two different public keys $pk$ and $pk'$ and a message-signature pair $(m, \sigma)$ such that (1) both public keys $pk$ and $pk'$ are valid under the same set of domain parameters $params$, and (2) the pair $(m, \sigma)$ is valid under both $pk$ and $pk'$ (also with respect to the same set of domain parameters $params$).

Besides, a related notion–domain parameter substitution attacks[1] are considered in [16,17]. In this kind of attacks, an adversary is given a set of domain parameters $params$, a public key $pk$ and a signing oracle. The goal of the adversary is to output a new set of domain parameters $params'$ and a signature $\sigma$ on an un-queried message $m$ such that (1) $params'$ passes the test for the domain parameters verification algorithm, and (2) the pair $(m, \sigma)$ is valid under the same public key $pk$ but with respect to the different set of domain parameters $params'$.

**SM2 Digital Signature Scheme.** The SM2 digital signature scheme [2] was issued by the State Cryptography Administration Office of Security Commercial Code Administration in 2010, and had become the Chinese cryptographic public key algorithm standard GM/T 0003.2-2012. Later, it was adopted by Trusted Computing Group (TCG) in the TPM 2.0 specification [10] which will be published as the international standard ISO/IEC 11889:2015 [11].

### 1.1 Our Contributions

In this paper, we consider *generalized* key substitution attacks where the base element is regarded as a part of the public key and can be substituted. In detail,

[1] In [17], Vaudenay referred to this type of attacks presented in [16,17] as domain parameter shifting attacks. Later, this kind of attacks are called domain parameter substitution attacks in [4].

given a public key $(g, pk)$ that $g$ is a basis and $pk$ is the other part of the public key and a valid message-signature pair $(m, \sigma)$ under $(g, pk)$, the goal of an adversary is to output another public key $(g', pk')$ such that $(g, pk) \neq (g', pk')$ and $(m, \sigma)$ is also valid under $(g', pk')$. This is possible when the domain parameters are generated by a signer, or the domain parameters are not properly validated, and has been considered by Blake-Wilson and Menezes [3] when examining the security against key substitution attacks on DSA and ECDSA signature schemes.

We first examine the security of a general framework of certificate-based signature schemes specified by ISO/IEC CD 14888-3 [1], and show that it is vulnerable to *generalized* key substitution attacks in the *weak* sense that the adversary knows the private key corresponding to the substituted public key.

Then, we analyze the security of the SM2 signature scheme [2] against chosen-message attack and generalized KS attacks respectively. Concretely, we not only show that SM2 signature scheme satisfies the EUF-CMA notion in the generic group model [14] provided that the underlying hash function $h$ is uniform and collision-resistant and the underlying conversion function $f$ is almost-invertible, but also give a formal proof that SM2 is secure against the *generalized* key substitution attacks in the *strong* sense (i.e., the adversary is not required to output the private key corresponding to the substituted public key) if the underlying hash functions $H$ and $h$ are both modeled as *non-programmable* random oracles [13].

## 2 Preliminaries

**Notation.** Throughout this paper, $\kappa$ denotes the security parameter. We denote by $s \xleftarrow{\$} S$ the fact that $s$ is picked uniformly at random from a finite set $S$. We write $(y_1, y_2, \dots) \leftarrow A(x_1, x_2, \dots)$ as the process that runs a randomized algorithm $A$ on input $(x_1, x_2, \dots)$ and obtains its output $(y_1, y_2, \dots)$. The notation $[n]$ denotes the set $\{1, \dots, n\}$ for some positive integer $n$. We use $\mathbb{F}_q$ and $\mathbb{A}_n$ to denote the set $\{0, 1, \dots, q-1\}$ and a group with the order $n$ respectively. We say that a function $f : \mathbb{N} \to [0, 1]$ is negligible if for every positive $c$ and sufficiently large $\kappa$ we have $f(\kappa) < 1/\kappa^c$, and is overwhelming if $1 - f$ is a negligible function.

### 2.1 Collision-Resistant Hash Functions

A hash function $h : \{0,1\}^* \mapsto \mathcal{R}$ is said to be collision-resistant if for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that

$$\Pr[(x, y) \leftarrow \mathcal{A}(1^\kappa, h) : x \neq y \wedge h(x) = h(y)] \leq \nu(\kappa),$$

where $\mathcal{R}$ denotes the range of $h$.

### 2.2 Uniform (Smooth) Hash Functions

Following the definition [5], the uniformity (or smoothness) of a hash function $h$ is described as below. Let $h : \{0,1\}^* \mapsto \mathcal{R}$ be a hash function. Let $\mathcal{D} \subseteq \{0,1\}^*$ such that

1. For $x \xleftarrow{\$} \mathcal{D}$, $y = h(x)$ can be efficiently generated.
2. For each $y \in \mathcal{R}$, the set $S_y = h^{-1}(y) \cap \mathcal{D}$ is sufficiently large so that the probability $1/|S_y|$ is sufficiently small (negligible) to make guessing a randomly picked secret element of $S_y$ infeasible.

We say that $h$ is uniform for $\mathcal{D}$ if for any PPT adversary $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that

$$\left| \Pr[x \xleftarrow{\$} \mathcal{D} : \mathcal{A}(1^\kappa, h, h(x)) = 1] - \Pr[y \xleftarrow{\$} \mathcal{R} : \mathcal{A}(1^\kappa, h, y) = 1] \right| \leq \nu(\kappa)$$

### 2.3 Almost-Invertibility of Conversion Functions

SM2 uses a conversion function $f : \mathbb{A}_n \mapsto \mathbb{F}_n$ which could be efficiently computed. Almost-invertibility of the conversion function is associated with the EUF-CMA security of SM2, and is defined in [5]. Concretely, a conversion function $f$ is almost-invertible if an almost-inverse of $f$ is efficiently computed. An almost-inverse of $f$ is a probabilistic polynomial time (PPT) algorithm $f^{-1} : \mathbb{F}_n \mapsto \mathbb{A}_n$ which on input $x \in \mathbb{F}_n$ produces a $Q \in \mathbb{A}_n \cup \{\texttt{Invalid}\}$ such that:

- The probability $Q \neq \texttt{Invalid}$ is at least $1/10$ over random choices of $x$ and the almost-inverse $f^{-1}$.
- If $Q \neq \texttt{Invalid}$, $f(Q) = x$.
- If independently random inputs $x \xleftarrow{\$} \mathbb{F}_n$ are repeatedly input to the algorithm $f^{-1}$ until the output $Q \neq \texttt{Invalid}$, the probability distribution of the resulting $Q$ is computationally indistinguishable from the distribution of a random element $Q \in \mathbb{A}_n$.

## 3 Definitions

Following the definitions in [12], we present the syntax and security notions of a signature scheme in the multi-user setting. Concretely, the syntax is described in Definition 1, the security model for existential unforgeability under adaptively chosen-message attacks (EUF-CMA) [9] is formalized in Definition 2, and the security notion for (generalized) strong key substitution (SKS) attacks is defined in Definition 3.

**Definition 1 (Syntax).** *A signature scheme in the multi-user setting consists of the following algorithms.*

- $\mathsf{Setup}(1^\kappa)$. On input a security parameter $\kappa$, the setup algorithm returns the domain parameters *params*.
- $\mathsf{Keygen}(params)$. On input the domain parameters *params*, the key generation algorithm returns a public-private key pair $(\mathrm{PK}, \mathrm{SK})$. Recall that PK contains the base element.
- $\mathsf{Sign}(params, \mathrm{SK}, m)$. On input the domain parameters *params*, the private key SK and a message $m$, the signing algorithm returns a signature $\sigma$ on the message $m$.

**Experiment** $\mathbf{Exp}^{\text{EUF-CMA}}_{SIG,\mathcal{F}}(\kappa)$

$\mathcal{Q} := \varnothing$; $params \leftarrow \mathsf{Setup}(1^\kappa)$; $(\text{PK}, \text{SK}) \leftarrow \mathsf{Keygen}(params)$;

$(m^*, \sigma^*) \leftarrow \mathcal{F}^{\mathsf{Sign}(params,\text{SK},\cdot)}(params, \text{PK})$;

If $m^* \notin \mathcal{Q} \ \wedge \ \mathsf{Verify}(params, \text{PK}, \sigma^*, m^*) = 1$, return 1.

Otherwise, return 0.

**Signing oracle** $\mathsf{Sign}(params, \text{SK}, m)$

$\sigma \leftarrow \mathsf{Sign}(params, \text{SK}, m)$;

$\mathcal{Q} := \mathcal{Q} \cup \{m\}$;

Return $\sigma$.

**Fig. 1.** Experiment for EUF-CMA security

– $\mathsf{Verify}(params, \text{PK}, \sigma, m)$. On input the domain parameters $params$, the public key PK and a candidate signature $\sigma$ on a message $m$, the deterministic verification algorithm returns 1 if $\sigma$ is valid on $m$ under PK and 0 otherwise.

Besides, we define an additional checking algorithm $\mathsf{Check}$ to check the validity of a public key PK. Specifically, given the domain parameters $params$ and a candidate public key PK, the checking algorithm $\mathsf{Check}(params, \text{PK})$ returns 1 if and only if the public key PK is valid under the domain parameters $params$.

**Definition 2 (EUF-CMA).** *We say that a signature scheme is existentially unforgeable under adaptively chosen-message attacks (EUF-CMA) if for any probabilistic polynomial time (PPT) adversary $\mathcal{F}$, there exists a negligible function $\nu(\cdot)$ such that*

$$\mathbf{Adv}^{\text{EUF-CMA}}_{SIG,\mathcal{F}}(\kappa) := \Pr[\mathbf{Exp}^{\text{EUF-CMA}}_{SIG,\mathcal{F}}(\kappa) = 1] \leq \nu(\kappa),$$

*where* $\mathbf{Exp}^{\text{EUF-CMA}}_{SIG,\mathcal{F}}(\kappa)$ *is defined in* **Fig. 1**.

**Definition 3 (SKS).** *We say that a signature scheme is secure against (generalized) strong key substitution attacks if for any PPT adversary $\mathcal{F}$, there exists a negligible function $\nu(\cdot)$ such that*

$$\mathbf{Adv}^{\text{SKS}}_{SIG,\mathcal{F}}(\kappa) := \Pr[\mathbf{Exp}^{\text{SKS}}_{SIG,\mathcal{F}}(\kappa) = 1] \leq \nu(\kappa),$$

*where* $\mathbf{Exp}^{\text{SKS}}_{SIG,\mathcal{F}}(\kappa)$ *is defined in* **Fig. 2**.

**Experiment** $\mathbf{Exp}^{\mathrm{SKS}}_{SIG,\mathcal{F}}(\kappa)$

$\mathcal{Q} := \varnothing$; $params \leftarrow \mathsf{Setup}(1^{\kappa})$; $(\mathrm{PK}, \mathrm{SK}) \leftarrow \mathsf{Keygen}(params)$;

$(m^*, \sigma^*, \mathrm{PK}^*) \leftarrow \mathcal{F}^{\mathsf{Sign}(params,\mathrm{SK},\cdot)}(params, \mathrm{PK})$;

If $\mathrm{PK}^* \neq \mathrm{PK} \ \wedge \ \mathsf{Check}(params, \mathrm{PK}^*) = 1 \ \wedge \ (m^*, \sigma^*) \in \mathcal{Q}$

$\wedge \ \mathsf{Verify}(params, \mathrm{PK}^*, \sigma^*, m^*) = 1$, return 1.

Otherwise, return 0.

**Signing oracle** $\mathsf{Sign}(params, \mathrm{SK}, m)$

$\sigma \leftarrow \mathsf{Sign}(params, \mathrm{SK}, m)$;

$\mathcal{Q} := \mathcal{Q} \cup \{(m, \sigma)\}$;

Return $\sigma$.

**Fig. 2.** Experiment for (generalized) strong key substitution attacks

We could easily modify the definition for (generalized) strong key substitution (SKS) attacks to capture the notion for (generalized) weak key substitution (WKS) attacks. The experiment for (generalized) WKS attacks is the same as $\mathbf{Exp}^{\mathrm{SKS}}_{SIG,\mathcal{F}}(\kappa)$ defined in **Fig.** 2, except that an adversary against (generalized) WKS attacks is further required to output the private key $\mathrm{SK}^*$ of the substituted public key $\mathrm{PK}^*$.

## 4 Generalized WKS Attacks Against a General Framework of ISO/IEC CD 14888-3

In this section, we first review the general framework of certificate-based mechanisms of ISO/IEC CD 14888-3 [1] in the setting that a signer chooses the base element as a part of its public key, and then show that the general framework is vulnerable to the *generalized* weak key substitution (WKS) attacks.

**General Framework.** The general framework of certificate-based mechanisms specified in ISO/IEC CD 14888-3 [1] is presented as follows.

- $\mathsf{Setup}(1^{\kappa})$. Given a security parameter $\kappa$, pick a finite commutative group $\mathbf{E}$ where multiplicative notation is used, and a prime divisor $q$ of the cardinality of $\mathbf{E}$, and choose an element $G \in \mathbf{E}$ of order $q$. Return $params := (\mathbf{E}, q, G)$ as a set of domain parameters.
- $\mathsf{Keygen}(params)$. Given the set of domain parameters $params = (\mathbf{E}, q, G)$, choose $X \xleftarrow{\$} \mathbb{F}_q \backslash \{0\}$. Then, compute $Y := G^X$. Actually, in the certificate-based mechanisms of ISO/IEC CD 14888-3, $Y$ is equal to either $G^X$ or $G^{X^{-1}}$

relying on the specific mechanism. Without loss of generality, we only consider the case that $Y = G^X$. Finally, output $\text{PK} = (G, Y)$ and $\text{SK} = X$ as the public key and the private key respectively.

– $\mathsf{Sign}(params, \text{SK}, M)$. Given the set of domain parameters $params = (\mathbf{E}, q, G)$, the private key $\text{SK} = X$ and a message $M$, the signing process is executed as follows:
  1. (*Producing the randomizer*) Choose $K \xleftarrow{\$} \mathbb{F}_q \backslash \{0\}$.
  2. (*Producing the pre-signature*) Compute $\Pi := G^K$.
  3. (*Preparing the message for signing*) Depending on the particular mechanism, one of $M_1$ and $M_2$ is set as $M$, and the other is set as empty.
  4. (*Computing the witness (the first part of the signature)*) The values of $\Pi$ and $M_1$ are taken as inputs to the witness function which is specified in the concrete mechanism. The output of the witness function is the witness $R$.
  5. (*Computing the assignment*) The witness $R$, $M_2$ and (optionally) $Y$ are taken as input to the assignment function which is defined in the particular mechanism. Then, the assignment function outputs assignment $T = (T_1, T_2)$ where $T_1$ and $T_2$ are integers such that $0 < |T_1| < q$ and $0 < |T_2| < q$.
  6. (*Computing the second part of the signature*) Let $S$ be the second part of the signature and $(A, B, C)$ is a permutation of three elements $(S, T_1, T_2)$ depending on the particular mechanism. Solve the following signature equation for $S$ where $S \in \mathbb{F}_q \backslash \{0\}$:

$$AK + BX + C \equiv 0 \pmod{q}.$$

  7. Output $\sigma := (R, S)$ as the signature.

– $\mathsf{Verify}(params, \text{PK}, \sigma, M)$. Given the set of domain parameters $params$, the public key $\text{PK} = (G, Y)$ and a candidate signature $\sigma = (R, S)$ on a message $M$, the verification process is executed as below:
  1. (*Preparing message for verification*) Divide the message $M$ into two parts $M_1$ and $M_2$.
  2. (*Retrieving the assignment*) Recompute the assignment $T = (T_1, T_2)$ using the assignment function with the inputs $R$, $M_2$ and (optionally) $Y$.
  3. (*Recomputing the pre-signature*) Set $(A, B, C)$ as $(S, T_1, T_2)$ according to the order specified in the signature algorithm. Recompute the pre-signature $\Pi' := Y^m G^n$ where $m = -A^{-1}B \mod q$ and $n = -A^{-1}C \mod q$.
  4. (*Recomputing the witness*) Recompute the witness $R'$ via executing the witness function with the inputs $\Pi'$ and $M_1$.
  5 (*Verifying the witness*) If $R = R'$, then return 1, else return 0.

**Generalized WKS Attacks.** Recall that given a valid message-signature pair $(M, (R, S))$ under the public key $\text{PK} = (G, Y)$ of some legitimate user, the goal of a *generalized* WKS adversary $\mathcal{A}$ is to produce a public-private key pair $(\text{PK}' = (G', Y' = (G')^{X'}), X')$ such that $\text{PK}' \neq \text{PK}$, but the message-signature pair

$(M,(R,S))$ is still valid under the public key $\text{PK}'$. A *generalized* WKS adversary $\mathcal{A}$ for the general framework of certificate-based mechanisms of ISO/IEC CD 14888-3 [1] is described as follows.

The adversary $\mathcal{A}$ first computes $m$ and $n$ with $(M,(R,S))$ and (optionally) $Y$ following the verification process. Then, $\mathcal{A}$ computes $\Pi = Y^m G^n$. In the following, the attack manner of $\mathcal{A}$ is divided into the following two cases depending on whether or not $Y$ is used to generate $(T_1, T_2)$:

- If $Y$ is not used to generate $(T_1, T_2)$, then the values of $m$ and $n$, which are created with $(M,(R,S))$, remain unchanged according to the verification process. Choose $X' \xleftarrow{\$} \mathbb{F}_q \backslash \{0\}$, and then compute $G' := \Pi^{1/(mX'+n)}$ and $Y' := (G')^{X'}$. Finally, output the new public-private key pair $(\text{PK}' = (G', Y'), X')$. It is easy to see that $\Pi' = (Y')^m (G')^n = (G')^{mX'+n} = \Pi$.
- If $Y$ is used to generate $(T_1, T_2)$, choose $t \xleftarrow{\$} \mathbb{F}_q \backslash \{0\}$ and compute $Y' := \Pi^t$. Then, compute $m'$ and $n'$ with $(M,(R,S))$ and $Y'$ following the verification process. Finally, compute $X' = (1 - tm')/tn' \mod q$ and $G' = \Pi^{tX'}$, and then output the new public-private key pair $(\text{PK}' = (G', Y'), 1/X')$. Again, one can easily verify that $\Pi' = (Y')^{m'} (G')^{n'} = \Pi^{tm'+tn'X'} = \Pi$.

Since in both cases we always have that $\Pi'$ is equal to $\Pi$, the message-signature pair $(M,(R,S))$ is valid under the new public key $\text{PK}'$ according to the verification process. This shows that the above constructed adversary $\mathcal{A}$ will break the security against *generalized* WKS attacks on the general framework with probability 1.

## 5 Security of the SM2 Signature Scheme

In this section, we first recall the description of SM2 digital signature scheme, and then we present the formal security proofs showing that SM2 satisfies both EUF-CMA security and the security against *generalized* strong key substitution attacks.

### 5.1 SM2 Digital Signature Scheme

The Chinese digital signature standard SM2 [2] is based on elliptic curve which has a formal of $y^2 + xy = x^3 + ax^2 + b$ over $\mathbb{F}_q$ for some integer $q = 2^m$, and $y^2 = x^3 + ax + b$ over $\mathbb{F}_q$ for some large prime $q$. In other words, the curve is parameterized by $q$ and $(a, b)$. Denote $E(\mathbb{F}_q)$ as the additive finite group which consists of all the integer points (including the infinity point 0) on the elliptic curve. In the following, we give the formal description of the four algorithms of the SM2 signature scheme.

- $\mathsf{Setup}(1^\kappa)$: Given a security parameter $\kappa$, generate the elliptic curve parameters $(q, a, b, n)$ such that $n$ is a prime divisor of the cardinality of $E(\mathbb{F}_q)$ and $|n| \geq 2\kappa$, where $q, a, b$ is the curve parameter. Choose a (random) generator $G \in E(\mathbb{F}_q)$ of order $n$. Output a set of domain parameters $params := (q, a, b, n, G)$.

Let $h : \{0,1\}^* \mapsto \mathbb{F}_n$ and $H : \{0,1\}^* \mapsto \{0,1\}^{256}$ be two cryptographic hash functions. Let $\mathbb{A}_n \subseteq E(\mathbb{F}_q)$ be the cyclic group generated by $G$. The conversion function $f : \mathbb{A}_n \mapsto \mathbb{F}_n$ is defined as $f(Q) = x_Q \mod n$, where $x_Q$ is an integer representation of the $x$-coordinate of the elliptic curve point $Q \in \mathbb{A}_n$.

- $\mathsf{Keygen}(params)$: Given the domain parameters $params = (q, a, b, n, G)$, pick $d \xleftarrow{\$} \mathbb{F}_n \backslash \{0, n-1\}$ and compute $Y = dG$. Output the public-private key pair $\big(\text{PK} = (G, Y), \text{SK} = d\big)$.
- $\mathsf{Sign}(params, \text{SK}, \text{PK}, m)$: Given the set of domain parameters $params = (q, a, b, n, G)$, the private key $\text{SK} = d$, the public key $\text{PK} = (G, Y)$, and a message $m$, let $Z = H(ENTL \| ID \| a \| b \| G \| Y)$ where $ENTL$ denotes the length of $ID$ and $ID$ is the identity of the owner of PK and do the following:
  1. Let $\bar{m} = Z \| m$.
  2. Compute $e := h(\bar{m})$.
  3. Choose $k \xleftarrow{\$} \mathbb{F}_n \backslash \{0\}$.
  4. Compute $x_1 := f(kG)$.
  5. Compute $r := (e + x_1) \mod n$. If $r = 0$ or $r + k = n$, go back to step 3.
  6. Compute $s := (k - rd)/(1 + d) \mod n$.
  7. The signature on $m$ is $\sigma := (r, s)$.
- $\mathsf{Verify}(params, \text{PK}, \sigma', m')$: Given the set of domain parameters $params = (q, a, b, n, G)$, the public key $\text{PK} = (G, Y)$, and a signature $\sigma' = (r', s')$ on a message $m'$, let $Z = H(ENTL \| ID \| a \| b \| G \| Y)$ and do the following:
  1. If $r' \notin [1, n-1]$, output 0 and exit.
  2. If $s' \notin [1, n-1]$, output 0 and exit.
  3. Let $\bar{m}' = Z \| m'$.
  4. Compute $e' := h(\bar{m}')$.
  5. Compute $t := (r' + s') \mod n$. If $t = 0$, output 0 and exit.
  6. Compute $x'_1 := f(s'G + tY)$.
  7. Compute $R := (e' + x'_1) \mod n$.
  8. If $R = r'$, then output 1, else output 0.

The conversion function $f : \mathbb{A}_n \mapsto \mathbb{F}_n$ of SM2 is exactly the same as that of ECDSA, and has been shown to be almost-invertible in [5].

### 5.2 EUF-CMA Security of SM2

Now, we proceed to give a formal security proof showing the EUF-CMA security of SM2. Formally, we have the following theorem.

**Theorem 1.** *If $h$ is a uniform and collision-resistant hash function, and the conversion function $f$ is almost-invertible, SM2 is existentially unforgeable under adaptively chosen-message attacks in the generic group model.*

Note that in the generic group model, an adversary is not given direct access to the group, but rather only receives "handles" representing group elements. More concretely, the adversary must interact with an oracle to perform the group operations (including scalar-multiplication and addition) and obtain handles for

new elements. In particular, it is assumed that the adversary can only use handles previously received from its environment. Back to our case, in addition to directly to get group element handles from group operation queries, the adversary can also obtain handles from the public key and the signatures from signing oracle queries. Actually, the adversary can use the handles in the public key and the signatures as the "bases" to perform further groups operations. More formally, let $(G, Y)$ be the group element handles in the public key, and let $(V_1, \ldots, V_{q_s})$ be the group element handles created in the signing queries, where $q_s$ is the number of signing queries made by the adversary. Then, by the assumption that all the group elements that the adversary want to compute have a form of $z_1G + z_2Y + z_3V_1 + \ldots + z_{q_s+2}V_{q_s}$, where $z_1, \ldots, z_{q_s+2}$ are known integers chosen by the adversary. Thus, we can unify all the group operation queries by the coefficient vector $(z_1, \ldots, z_{q_s+2})$[2]. For example, multiplying the base element $G$ by an integer $z$ can be expressed as a group operation query $(z, 0, \ldots, 0)$.

*Proof.* In the following, for any PPT forger $\mathcal{F}$, we show there exists a challenger $\mathcal{C}$ to simulate the attack environment for $\mathcal{F}$ such that the advantage of $\mathcal{F}$ is negligible. In order to answer the group operation queries from $\mathcal{F}$, the challenger $\mathcal{C}$ will keep a table $L$ to record the information generated in the group operation queries. Formally, $\mathcal{C}$ first generates the handle $G$ of the base element by choosing $G \xleftarrow{\$} \mathbb{A}_n$, and adds $(1, G, -, -)$ into the table $L$, where $\mathbb{A}_n$ is a set supporting efficient sampling and representing the underlying group. Then, $\mathcal{C}$ chooses an integer $d \xleftarrow{\$} \mathbb{F}_n \backslash \{0, n-1\}$, $Y \xleftarrow{\$} \mathbb{A}_n$ as the handle of multiplying $G$ by $d$, and adds $(d, Y, -, -)$ into the table $L$. Let $q_s$ be the number of signing queries made by $\mathcal{F}$, $q_c$ be the current number of signing queries during the interaction between $\mathcal{C}$ and $\mathcal{F}$, and denote $V_1, \ldots, V_{q_s}$ as the group element handles that will be generated in the signing queries. $\mathcal{C}$ answers $\mathcal{F}$'s group operation queries and signing queries as follows.

- For a group operation query with input $(z_1, \ldots, z_{q_s+2})$ (i.e., $\mathcal{F}$ wants to compute $z_1G + z_2Y + z_3V_1 + \ldots + z_{q_s+2}V_{q_s}$), $\mathcal{C}$ does the following:
  1. Let $j$ be the maximum index such that $z_j \neq 0$.
  2. If $j > q_c + 2$, then return $\perp$ and exit.
  3. Otherwise, for each $i \in \{1, \ldots, j\}$, retrieve $k_i$ from the entry $(k_i, V_i, -, -)$ in table $L$ and compute $z' = z_1 + z_2 d + z_3 k_1 + \ldots + z_j k_j \mod n$.
  4. If there exists an entry $(z', V', -, -)$ in table $L$, $\mathcal{C}$ directly returns $V'$ to $\mathcal{F}$. Otherwise, it distinguishes the following two cases:
     - Case 1: If $z_2 = 0$, choose $V' \xleftarrow{\$} \mathbb{A}_n$, add $(z', V', (z_1, \ldots, z_{q_s+2}), -)$ into table $L$. Finally, return $V'$ to $\mathcal{F}$.
     - Case 2: If $z_2 \neq 0$, randomly choose $Z' \xleftarrow{\$} \{0,1\}^{256}, m' \xleftarrow{\$} \mathcal{M}$[3], and compute

$$V' = f^{-1}(z_2 - z_1 - z_3 k_1 - \ldots - z_{q_s+2} k_{q_s} - h(Z' \| m'))$$

[2] In this case, if some $z_i$ is equal to 0, it means that the corresponding group element is not involved in the computation.

[3] We use $\mathcal{M}$ to denote the efficiently sampling message space of SM2.

until $V' \in \mathbb{A}_n$. Then, add $(z', V', (z_1, \ldots, z_{q_s+2}), Z'\|m')$ into table $L$, and return $V'$ to $\mathcal{F}$.

– For a signing query on some message $m$, $\mathcal{C}$ chooses $k \xleftarrow{\$} \mathbb{F}_n \backslash \{0\}$, and makes a group operation query $(k, 0, \ldots, 0)$ by itself to obtain a handle $V$. Then, it computes $x = f(V), r = h(Z\|m) + x \mod n$, and $s = (k - rd)/(1 + d) \mod n$, where $Z$ is the other information as determined in the signing algorithm SM2. Finally, $\mathcal{C}$ returns $(r, s)$ as the signature on $m$ to $\mathcal{F}$.

After making polynomial time queries of the above two types, the adversary $\mathcal{F}$ will output a forged signature $(r^*, s^*)$ for $m^* \notin \{m_i\}_{i \in [q_s]}$. Below, we prove the probability that the forged signature is valid to be negligible under the assumption that $h$ is uniform and collision-resistant.

*Analysis.* Note that $\mathcal{C}$ honestly generates the public key and the signatures, if $\mathcal{C}$ also perfectly answers the group operation queries, then we have that $\mathcal{C}$ almost simulates a perfect attack environment for $\mathcal{F}$. Actually, it is easy to check that all the group element handles are uniformly chosen at random except in Case 2 of the group operation query. Now, we argue that the handle $V'$ generated in Case 2 is also uniformly distributed. In fact, since $Z'$ and $m'$ are uniformly chosen at random, and $h$ is a uniform function, we have that the input of $f^{-1}$ in Case 2 is uniformly distributed. By the fact that $f^{-1}$ is almost-invertible, we have $V'$ is uniformly distributed. In addition, by the Schwartz-Zippel Lemma, the probability that there exist two entries $(z', V', -, -)$ and $(z'', V'', -, -)$ in table $L$ such that $z' \neq z''$ but $V' = V''$ (i.e., $\mathcal{C}$ fails to simulate the generic group model due to the inconsistency) is bounded by $O\big(\frac{(q_G+q_s)^2}{n}\big)$ which is negligible, where $q_G$ denotes the total number of group operation queries made by $\mathcal{F}$. This finally shows that $\mathcal{C}$ almost perfectly simulates the attack environment for $\mathcal{F}$.

In order to finish the proof, we only have to show that the probability that $(r^*, s^*)$ is a valid signature on $m^*$ is negligible. Before continuing, we note that the secret key $d$ is perfectly hidden from the adversary $\mathcal{F}$. This is because in the generic group model, $d$ is chosen independently from the group element handle $Y$ in the public key, and $d$ is perfectly hidden from the signature $(r, s)$ in the signing query (due to the randomly choices of $k$ and $V$). Let $k^* = s^* + (s^* + r^*)d$, then $(r^*, s^*)$ is a valid signature on $m^*$ if and only if there exists an entry $(k^*, V^*, -, -)$ in table $L$ such that $r^* - h(Z\|m^*) = f(V^*)$. We first claim that $V^* \notin \{G, Y\}$ holds with overwhelming probability. Otherwise, the adversary can deterministically compute $d$ from $(r^*, s^*)$ by using the fact that $s^* \neq 0$ and $s^* + r^* \neq 0$, which contradicts to the fact that $d$ is perfectly hidden from the adversary $\mathcal{F}$. In other words, $V^*$ can only be created either in answering the group operation query or in answering the signature queries. We distinguish the following two cases:

– If $V^* \in \{V_1, \ldots, V_{q_s}\}$, then let $V^* = V_i$ for some $i$, and let $(r_i, s_i)$ be the signature on some message $m_i$ and auxiliary information $Z_i$ in the $i$-th signing query. In other words, we have $s^* + (s^* + r^*)d = s_i + (s_i + r_i)d$. By the fact that $d$ is perfectly hidden from the adversary $\mathcal{F}$, this can only happen with

non-negligible probability when both $s^* = s_i$ and $s^* + r^* = s_i + r_i$ hold. In this case, $(r^*, s^*)$ is a valid signature on $m^*$ if and only if the equation $h(Z_i \| m_i) = r_i - f(V_i) = r^* - f(V^*) = h(Z \| m^*)$ holds. Since $m^* \neq m_i$, this means that $\mathcal{F}$ has to find a collision $(Z_i \| m_i, Z \| m^*)$ of the hash function $h$. Under the assumption that $h$ is collision-resistant, this can only happen with negligible probability.

– Else, $V^*$ is created by a group operation query with input $(z_1^*, \ldots, z_{q_s+2}^*)$. In this case, we have $k^* = s^* + (s^* + r^*)d = z_1^* + z_2^* d + z_3^* k_1 + \ldots + z_{q_s+2}^* k_{q_s}$. Again, by the fact that $d$ is perfectly hidden from the adversary $\mathcal{F}$, this can only happen with non-negligible probability when both $s^* = z_1^* + z_3^* k_1 + \ldots + z_{q_s+2}^* k_{q_s}$ and $s^* + r^* = z_2^*$ hold. By a simple computation, we have $r^* = z_2^* - z_1^* - z_3^* k_1 - \ldots - z_{q_s+2}^* k_{q_s}$. Besides, according to the strategy of $\mathcal{C}$ (in Case 2), there exists a pair $(Z', m')$ chosen by $\mathcal{C}$ such that $f(V^*) = z_2^* - z_1^* - z_3^* k_1 - \ldots - z_{q_s+2}^* k_{q_s} - h(Z' \| m')$. In other words, $(r^*, s^*)$ is a valid signature if and only if $h(Z \| m^*) = r^* - f(V^*) = h(Z' \| m')$. However, under the assumption that $h$ is collision-resistant, the probability that $\mathcal{F}$ outputs a pair $(Z \| m^*)$ such that $h(Z \| m^*) = h(Z' \| m')$ is negligible.

In all, we have shown that under the assumption that $h$ is uniform and collision-resistant, the probability that $(r^*, s^*)$ is a valid signature on $m^*$ is negligible, which completes the proof. □

### 5.3 Security of SM2 Against Generalized SKS Attacks

In this subsection, we show that SM2 is secure against *generalized* SKS attacks. Formally, we have the following theorem.

**Theorem 2.** *If both $H$ and $h$ are modeled as non-programmable random oracles (NPROs), then SM2 is secure against generalized strong key substitution attacks.*

*Proof.* In the following, we will show that the advantage of any PPT adversary $\mathcal{F}$ against the *generalized* SKS security of SM2 is negligible. Formally, in order to simulate the attack environment for $\mathcal{F}$, the challenger $\mathcal{C}$ only has to generate the domain parameters *params* and $(\mathrm{PK}, \mathrm{SK})$, and answers the signing queries honestly. More concretely, $\mathcal{C}$ first runs the $\mathsf{Setup}$ and $\mathsf{Keygen}$ algorithms to obtain $params = (q, a, b, n, G)$ and $(\mathrm{PK}, \mathrm{SK}) = ((G, Y), d)$ where $Y = dG$. Then, let $ID$ be the identity of the owner of PK, and send $(params, \mathrm{PK})$ to $\mathcal{F}$. Recall that in our model $h$ and $H$ are modeled as NPROs, both the challenger $\mathcal{C}$ and the adversary $\mathcal{F}$ have to access the external random oracles $h$ and $H$ to realize the functionality of SM2.

After receiving the $i$-th signing query on a message $m_i$, $\mathcal{C}$ honestly computes $(r_i, s_i) \leftarrow \mathsf{Sign}(params, \mathrm{SK}, m_i)$ by making appropriate random oracle queries to $h$ and $H$, and returns $\sigma_i = (r_i, s_i)$ as the signature on $m_i$ to $\mathcal{F}$. Let $q_s$ be the number of the signing queries issued by $\mathcal{F}$.

Finally, $\mathcal{F}$ will return $(ID^*, m^*, (r^*, s^*), \text{PK}^* = (G^*, Y^*))$ as its output[4], such that (1) $(G^*, Y^*) \neq (G, Y)$, (2) $(G^*, Y^*)$ is valid[5], and (3) $(m^*, r^*, s^*) = (m_j, r_j, s_j)$ for some $1 \leq j \leq q_s$.

*Analysis.* Now, we will show that $\mathcal{F}$ can only win the SKS game with negligible probability. Specifically, the probability that the message-signature pair $(m_j, r_j, s_j)$ is valid under $\text{PK}^*$ is negligible in the security parameter $\kappa$. Note that $(m_j, r_j, s_j)$ is valid under $\text{PK}^* = (G^*, Y^*)$ if and only if

$$r_j = e_j + f\left(s_j G + (r_j + s_j)Y\right) = e^* + f\left(s_j G^* + (r_j + s_j)Y^*\right) \mod n, \quad (1)$$

where $e^* = h(H(ENTL^* \| ID^* \| a \| b \| G^* \| Y^*) \| m_j)$ and $e_j = h(H(ENTL \| ID \| a \| b \| G \| Y) \| m_j)$. Since $\text{PK} \neq \text{PK}^*$, we have that the distribution of $e_j$ is independent from that of $e^*$, and that $e_j \neq e^*$ holds with overwhelming probability. This means that the distribution of $\sigma_j = (r_j, s_j)$ is independent from $e^*$ according to the signing algorithm. In other words, the distribution of $e^*$ is still uniform conditioned on the equation (1) holds by the assumption that both $h$ and $H$ are NPROs. Note that $\mathcal{F}$ must first fix $\text{PK}^* = (G^*, Y^*)$ to make the appropriate $H$ query, and that the outputs of both $h$ and $H$ are uniformly distributed, the probability that $e^* = h(H(ENTL^* \| ID^* \| a \| b \| G^* \| Y^*) \| m_j)$ satisfying a prior fixed equation $e^* = r_j - f\left(s_j G^* + (r_j + s_j)Y^*\right) \mod n$ is negligible, which shows that the equation (1) can only hold with negligible probability. This completes the proof of Theorem 2. □

**Acknowledgements.** We would like to thank Hui Guo and the anonymous reviewers for their helpful comments.

## References

1. ISO/IEC 1st CD 14888–3 - Information technology - Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms
2. GM/T 0003.2-2012, Public Key Cryptographic Algorithm SM2 based on Elliptic Curves - Part 2: Digital Signature Algorithm (2010). http://www.oscca.gov.cn/UpFile/2010122214822692.pdf
3. Blake-Wilson, S., Menezes, A.: Unknown key-share attacks on the Station-to-Station (STS) Protocol. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 154–170. Springer, Heidelberg (1999)
4. Bohli, J.-M., Røhrich, S., Steinwandt, R.: Key substitution attacks revisited: taking into account malicious signers. Int. J. Inf. Secur. **5**(1), 30–36 (2006)
5. Brown, D.R.L.: Generic groups, collision resistance, and ECDSA. Des. Codes Crypt. **35**(1), 119–152 (2005)

[4] We also allow the adversary to output the identity $ID^*$ of the owner of $\text{PK}^*$. This is only because both the signing and verification algorithms of SM2 have an identity input. We do not have any additional restriction on $ID^*$.

[5] To verify the validity of $(G^*, Y^*)$, the following conditions need to be satified: (1) $G^* \in E(\mathbb{F}_q)$, (2) the order of $G^*$ is $n$, and 3) $Y^* \in < G^* > \setminus \{0\}$.

6. Diffie, W., Van Oorschot, P.C., Wiener, M.J.: Authentication and authenticated key exchanges. Des. Codes Crypt. **2**(2), 107–125 (1992)
7. Geiselmann, W., Steinwandt, R.: A Key Substitution Attack on SFLASH$^{v3}$. Cryptology ePrint Archive, Report 2003/245 (2003). http://eprint.iacr.org/
8. Goldwasser, S., Micali, S., Rivest, R.L.: A paradoxical solution to the signature problem. In: Proceedings of the IEEE 25th Annual Symposium on Foundations of Computer Science, pp. 441–448. IEEE (1984)
9. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (1988)
10. Trusted Computing Group. TCG TPM specification 2.0. (2013) http://www.trustedcomputinggroup.org/resources/tpm_library_specification
11. ISO/IEC 11889:2015. Information technology - Trusted Platform Module Library (2015)
12. Menezes, A., Smart, N.: Security of signature schemes in a multi-user setting. Des. Codes Crypt. **33**(3), 261–274 (2004)
13. Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: the non-committing encryption case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Heidelberg (2002)
14. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997)
15. Tan, C.H.: Key substitution attacks on some provably secure signature schemes. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E87–A**(1), 226–227 (2004)
16. Vaudenay, S.: The security of DSA and ECDSA. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 309–323. Springer, Heidelberg (2002)
17. Vaudenay, S.: Digital signature schemes with domain parameters. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 188–199. Springer, Heidelberg (2004)