УДК 519.7 DOI 10.17223/20710410/63/3

# BLIND SIGNATURE AS A SHIELD AGAINST BACKDOORS IN SMART CARDS

L. R. Akhmetzyanova, A. A. Babueva, A. A. Bozhko

*CryptoPro, Moscow, Russia*

**E-mail:** {lah, babueva, bozhko}@cryptopro.ru

The problem of signature forgery (including signature key recovery) in the presence of backdoors in the hardware or software of functional key carriers (smart cards) is considered. A new approach to solving the problem based on using blind signature schemes is proposed. It is shown that honest-signer blindness and honest-but-curious unforgeability of the blind signature schemes imply security against backdoors in smart cards. As a concrete example, we consider a blind version of the GOST signature scheme (the blind signature scheme proposed by Camenisch) and show that this scheme is resistant to backdoors under the single assumption that GOST is secure in the standard sense.

**Keywords:** *blind signature scheme, GOST R 34.10-2012, untrusted smart cards, backdoors.*

# СХЕМЫ ПОДПИСИ ВСЛЕПУЮ КАК ЗАЩИТА ОТ ЗАКЛАДОК В СМАРТ-КАРТАХ

Л. Р. Ахметзянова, А. А. Бабуева, А. А. Божко

*КриптоПро, г. Москва, Россия*

Рассматривается задача обеспечения защиты от подделки подписи (в том числе за счёт восстановления ключа подписи) в условиях наличия закладок в аппаратном или программном обеспечении функциональных ключевых носителей (смарт-карт). Предлагается новый подход к решению задачи, основанный на использовании схем подписи вслепую. Показывается, что обеспечение схемой подписи вслепую свойств неотслеживаемости при условии честной генерации ключей и неподделываемости относительно «честного, но любопытного» нарушителя обеспечивает защиту от закладок в смарт-картах. В качестве конкретного примера рассматривается схема подписи вслепую на основе уравнения ГОСТ, предложенная Каменишем. Доказывается, что эта схема обеспечивает защиту от закладок при единственном предположении, что схема подписи ГОСТ обеспечивает свойство неподделываемости в стандартном смысле.

**Ключевые слова:** *схема подписи вслепую, ГОСТ Р 34.10-2012, недоверенные смарт-карты, закладки.*

## 1. Introduction

Consider an information system consisting of two components: a smart card (or token) used as a functional key storage and an application installed on a user device (desktop or handheld). The applied function of a system is to compute a signature of any document transmitted via the application with a key uploaded and stored on a smart card. The components usually interact in the following way:

1) the user opens the application, chooses the document to be signed and pushes the button "Sign";
2) the application connects to the smart card (usually by setting up a password-protected secure channel [1]) and sends it the selected document or document hash value;
3) the smart card computes the signature value of the document on its own under a stored private key and returns the computed value to the application;
4) the application verifies the received signature value and returns the signed document to the user.

The use of smart cards with unrecoverable on-board private key cryptography is considered one of the most secure approaches to key management that allows to protect against adversaries which can get physical access to key storage devices. However, it has its own disadvantages. Unlike software applications, which can be open source and therefore fully verified, self-compiled and securely installed by anyone, smart card development is a much more technically complex process that is usually carried out by companies that specialize in the field. Indeed, the signing code is often hardwired directly into smart card microchips to improve performance and, consequently, cannot be openly verified by outsiders: the users are given a ready-to-use "black-box" device. This makes it possible for unscrupulous developers to implement a malicious code.

In the paper, we address the security issues that arise when the smart card used is seen as an untrusted component and is believed to contain backdoors. In the context of systems based on ElGamal or Schnorr type signature schemes, these issues are highly crucial, since this type of signature uses one-time random values that are generated using a smart card and whose compromise immediately results in the recovery of the user's private key. For instance, malicious smart card can use low-entropy one-time values allowing an adversary (e.g., company implementing this backdoor) to perform the brute force attack and recover the user key from a correct signature.

**Related work.** The paper [2] is devoted to these issues. Firstly, the paper introduces two types of adversary to be considered:

External adversary: it models an honest-but-curious adversary acting on the application side; the adversary's goal is to make a new correct pair (message, signature) without interacting with a smart card or, in other words, to make a forgery. Note that this threat includes the stronger one — key recovery. Consideration of such adversaries covers the scenario where only honest user interacts with smart card through verified and trusted application, but this application is less protected from memory leaks compared to the smart card.

**Remark 1.** Note that this type does not cover the capabilities of active adversaries that can directly interact (e.g., using its own malicious application) with the smart card. In practice, it means that the adversary that steals the smart card cannot get access to its API. Considering only passive adversaries is justified by the fact that smart cards are usually also protected with a memorable password that should be entered by the human to get access to its API [3].

Adversary with agent: this adversary is supposed to consist of two parts. The first part is *a fully active adversary* on the smart card side but it can interact only with the trusted application, i.e., there is no other channel for data transmition from smart card. The second part collects the pairs (message, signature) computed by application and malicious smart card — this is the agent. Similar to the first type of adversary, the goal is to make a forgery.

In order to deal with these adversaries, the paper [2] proposed a solution for the GOST signature scheme [4] based on the usage of the interactive Schnorr zero-knowledge proof. This protocol is executed with the main signing algorithm and its purpose is to prove to the application that smart card is using the "correct" one-time value (for details see the original paper). This solution has the following two significant drawbacks:

1) it allows to protect against *the semi-trusted* smart card only: the crucial assumption for security is that low-level (short) arithmetic operations are implemented correctly in the smart cards. Although it is realistic assumption, there are no convenient ways to validate this on practice;
2) it is not secure if the smart card can terminate the signing process with the error on the application side. The paper [2] describes the concrete attack where the malicious smart card successfully completes the signing protocol only if certain bits of resulting signature are equal to certain bits of the signing key. One approach to protect against this attack is to delete the private signing key immediately after such errors occur. However, in practice, errors can occur not only due to the adversary's actions, but also due to technical failures, so deleting the key after each error is not a practical solution.

**Our contribution.** To negate the disadvantages mentioned above, we propose a new approach, the main idea of which is to use the "blind versions" of the signature schemes. The blind signature schemes firstly introduced by Chaum [5] allow one party called User to obtain a signature for an arbitrary message after interacting with another party called Signer holding a signing key in such a way that the Signer does not receive any information about either the message or the signature value (blindness property) and the User can compute only one single signature per interaction with the Signer (unforgeability property).

In the context of considered signing system, the smart card executes the Signer side and the application executes the User side. Due to the blindness property, the malicious smart card learns no information about the signature during the protocol execution and, therefore, cannot "control" the signature values, e.g., covertly transmiting bits of private key through the signature values.

In this paper, we introduce two new security notions for blind signature schemes: honest-but-curious unforgeability and backdoor resilience, which characterize the security of the proposed solution against external adversary and adversary with agent. We show that honest-signer blindness (where an adversary cannot affect the key generation algorithm) and standard unforgeability imply backdoor resilience. Moreover, for the GOST signature scheme we propose the concrete blind signature scheme for use: the Camenisch scheme [6] that provides perfect blindness (and thus honest-signer blindness) and honest-but-curious unforgeability (and thus standard unforgeability), which is implied only by the unforgeability of GOST. It means that the Camenisch blind signature scheme provides the security against both external adversary and adversary with agent under a single assumption that the GOST signature scheme provides standard security, i.e., is unforgeable under the chosen message attack.

The rest of the paper is organised as follows. In Section 2 we remind the definitions of conventional and blind signature schemes, the accompanying security notions are given. In Section 3 the formal definitions of honest-but-curious unforgeability and backdoor resilience are introduced. Section 4 is devoted to the formal analysis and Section 5 considers the Camenish blind signature scheme in details.

## 2. Basic definitions

**(Conventional) signature schemes.** The conventional signature scheme $\mathsf{SS}$ is determined by three algorithms:

— $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{SS.KGen}()$: a key generation algorithm that outputs a secret key $\mathsf{sk}$ and a public key $\mathsf{pk}$;

— $\sigma \leftarrow \mathsf{SS.Sig}(\mathsf{sk}, m)$: a signature generation algorithm that takes a secret key $\mathsf{sk}$ and a message $m$ and returns a signature $\sigma$;

— $b \leftarrow \mathsf{SS.Vf}(\mathsf{pk}, m, \sigma)$: a (deterministic) verification algorithm that takes a public key $\mathsf{pk}$, a message $m$, and a signature $\sigma$, and returns 1 if $\sigma$ is valid on $m$ under $\mathsf{pk}$ and 0 otherwise.

*Correctness.* We say that $\mathsf{SS}$ is correct if for each message $m$, with probability one over the sample of parameters and the key pair $(\mathsf{sk}, \mathsf{pk})$, the equality $\mathsf{SS.Vf}(\mathsf{pk}, m, \mathsf{SS.Sig}(\mathsf{sk}, m)) = 1$ holds.

**Blind signature schemes.** The blind signature scheme $\mathsf{BS}$ is defined in the same way as the conventional signature scheme except for the signature generation algorithm which is replaced by the following protocol:

— $(b, \sigma) \leftarrow \langle \mathsf{BS.Signer}(\mathsf{sk}), \mathsf{BS.User}(\mathsf{pk}, m)\rangle$: an interactive signing protocol that is run between a Signer with a secret key $\mathsf{sk}$ and a User with a public key $\mathsf{pk}$ and a message $m$; the Signer outputs $b = 1$ if the interaction completes successfully and $b = 0$ otherwise, while the User outputs $\sigma$ that is either the resulting signature or an error message.

*Correctness.* We say that $\mathsf{BS}$ is correct if for each message $m$, with probability one over the sample of parameters and the key pair $(\mathsf{sk}, \mathsf{pk})$, the signing protocol $\langle \mathsf{BS.Signer}(\mathsf{sk}), \mathsf{BS.User}(\mathsf{pk}, m)\rangle$ completes with $(1, \sigma)$, $\sigma \neq \perp$, such that $\mathsf{BS.Vf}(\mathsf{pk}, m, \sigma) = 1$.

In the paper, we are interested in the blind signature schemes that are based on some conventional signature schemes. We will say that the $\mathsf{BS}$ scheme is *a blind version* of the $\mathsf{SS}$ scheme, if the $\mathsf{KGen}$ and $\mathsf{Vf}$ algorithms of these schemes coincide and for any $(\mathsf{sk}, \mathsf{pk})$, any message $m$, and any signature $\sigma$

$$\Pr[(1, \sigma) \leftarrow \langle \mathsf{BS.Signer}(\mathsf{sk}), \mathsf{BS.User}(\mathsf{pk}, m)\rangle] = \Pr[\sigma \leftarrow \mathsf{SS.Sig}(\mathsf{sk}, m)],$$

where the corresponding probability spaces are determined by the randomness used in the signing protocol and signing algorithm.

*Three-move blind signature schemes.* For simplicity, this paper focuses on three-move blind signature schemes. For such schemes, the signing protocol can be described as follows:

$$\begin{aligned}
&(msg_{S,1}, state_S) \leftarrow \mathsf{BS.Signer}_1(\mathsf{sk}),\\
&(msg_U, state_U) \leftarrow \mathsf{BS.User}_1((\mathsf{pk}, m), msg_{S,1}),\\
&(msg_{S,2}, b) \leftarrow \mathsf{BS.Signer}_2(state_S, msg_{U,1})\\
&\sigma \leftarrow \mathsf{BS.User}_2(state_U, msg_{S,2}),
\end{aligned}$$

where $msg_{role,i}$, $role \in \{U, S\}$, is the $i$-th message sent by the side with role $role$ during the protocol execution. The variable $state_{role}$ is aimed to keep the internal state for using on the next protocol stage. Here the User performs the $\mathsf{BS.User}_1$ and $\mathsf{BS.User}_2$ functions, and the Signer performs the $\mathsf{BS.Signer}_1$ and $\mathsf{BS.Signer}_2$ functions during the protocol execution.

**Security notions.** Next, we describe security concepts using a game-based approach [7]. This approach uses the notion of "experiment" played between a challenger and an adversary. The adversary and challenger are modelled using consistent interactive

probabilistic algorithms. The challenger simulates the functioning of the analysed cryptographic scheme for the adversary and may provide him access to one or more oracles. The parameters of an adversary $\mathcal{A}$ are its computational resources (for a fixed model of computation and a method of encoding) and oracles query complexity. The query complexity usually includes the number of queries. Denote by $\mathsf{Adv}_S^{\mathrm{M}}(\mathcal{A})$ the measure of the success of the adversary $\mathcal{A}$ in realizing a certain threat, defined by the security notion M for the cryptographic scheme S.

The standard security notion for (probabilistic) signature schemes is strong unforgeability under chosen message attack (sUF-CMA). The formal definition is given below.

**Definition 1.** For an adversary $\mathcal{A}$ and a signature scheme $\mathsf{SS}$:

$$\mathsf{Adv}_{\mathsf{SS}}^{\text{sUF-CMA}}(\mathcal{A}) = \Pr\big[\mathbf{Exp}_{\mathsf{SS}}^{\text{sUF-CMA}}(\mathcal{A}) \to 1\big],$$

where the $\mathbf{Exp}_{\mathsf{SS}}^{\text{sUF-CMA}}(\mathcal{A})$ experiment is defined in the following way:

| $\mathbf{Exp}_{\mathsf{SS}}^{\text{sUF-CMA}}(\mathcal{A})$ | Oracle $Sign(m)$ |
|---|---|
| 1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{SS.KGen}()$ | 1 : $\sigma \leftarrow \mathsf{SS.Sig}(\mathsf{sk}, m)$ |
| 2 : $\mathcal{L} \leftarrow \varnothing$ | 2 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$ |
| 3 : $(m, \sigma) \leftarrow \mathcal{A}^{Sign}(\mathsf{pk})$ | 3 : **return** $\sigma$ |
| 4 : **if** $(m, \sigma) \in \mathcal{L}$ : **return** 0 | |
| 5 : **return** $\mathsf{SS.Vf}(\mathsf{pk}, m, \sigma)$ | |

**Remark 2.** The same security notion can be applied to the blind version $\mathsf{BS}$ of the signature scheme $\mathsf{SS}$. In this case, line 1 in the *Sign* oracle is replaced with the line $(1, \sigma) \leftarrow \langle \mathsf{BS.Signer}(\mathsf{sk}), \mathsf{BS.User}(\mathsf{pk}, m)\rangle$. It is easy to see that for such schemes sUF-CMA-security of the $\mathsf{SS}$ scheme implies sUF-CMA-security of the $\mathsf{BS}$ scheme and vice versa.

The standard notions for blind signature schemes are one-more unforgeability (OMUF notion that considers a malicious user in the parallel setting) and blindness (Blind notion that considers a malicious signer), their formal definitions can be found in [8]. Note that the original definition of blindness proposed in [9] considers an honest signer that can not affect key generation process. In the paper, we consider only this weak notion and refer to it as "honest-signer blindness" (HS-Blind notion).

*Honest-signer blindness.* Informally, the blind signature scheme provides blindness if there is no way to link a (message, signature) pair to the certain execution of the signing protocol. In the context of strong notion, the adversary can fully control the Signer side. In the context of weaker HS-Blind notion, we assume that the key pair is generated honestly at the beginning of the experiment. The formal definition is given below.

**Definition 2.** For an adversary $\mathcal{A}$ and three-move blind scheme $\mathsf{BS}$:

$$\mathsf{Adv}_{\mathsf{BS}}^{\text{HS-Blind}}(\mathcal{A}) = \Pr\Big[\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},1}(\mathcal{A}) \to 1\Big] - \Pr\Big[\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},0}(\mathcal{A}) \to 1\Big],$$

where the $\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},b}(\mathcal{A})$, $b \in \{0, 1\}$, experiments are defined in the following way:

$\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},b}(\mathcal{A})$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $b_0 \leftarrow b$
3 : $b_1 \leftarrow 1 - b$
4 : $b' \leftarrow \mathcal{A}^{Init, User_1, User_2}(\mathsf{sk}, \mathsf{pk})$
5 : **return** $b'$

Oracle $Init(m_0, m_1)$

1 : $sess_0 \leftarrow \mathsf{init}$
2 : $sess_1 \leftarrow \mathsf{init}$

Oracle $User_1(i, msg)$

1 : **if** $i \notin \{0, 1\} \vee sess_i \neq \mathsf{init}$ : **return** $\perp$
2 : $sess_i \leftarrow \mathsf{open}$
3 : $(msg_i, state_i) \leftarrow \mathsf{BS.User}_1((\mathsf{pk}, m_{b_i}), msg)$
4 : **return** $msg_i$

Oracle $User_2(i, msg)$

1 : **if** $sess_i \neq \mathsf{open}$ : **return** $\perp$
2 : $sess_i \leftarrow \mathsf{closed}$
3 : $\sigma_{b_i} \leftarrow \mathsf{BS.User}_2(state_i, msg)$
4 : **if** $sess_0 = sess_1 = \mathsf{closed}$ :
5 : **if** $\sigma_{b_0} = \perp \vee \sigma_{b_1} = \perp$ : **return** $(\perp, \perp)$
6 : **return** $(\sigma_0, \sigma_1)$
7 : **return** $\varepsilon$

## 3. New security notions for blind signatures

Here we give the formal game-based definitions of two security notions: backdoor resilience and honest-but-curious unforgeability.

**Backdoor resilience/Security against adversary with agent**

Consider an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ consisting of two algorithms. An algorithm $\mathcal{A}_2$ denotes the part of the adversary $\mathcal{A}$ collecting signature values for adaptively chosen messages. An algorithm $\mathcal{A}_1$ denotes the agent acting on the backdoored smart card side.

The formal definition of BDres (BackDoor resilience) for blind signature schemes is given below (see Definition 3). We parametrize this security model by the value $k$ which determines the number of attempts by the challenger to produce a correct signature for a message (details are described below).

**Definition 3.** For any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and blind signature scheme BS:

$$\mathsf{Adv}_{\mathsf{BS}}^{\text{BDres}_k}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{\text{BDres}_k}(\mathcal{A}) \to 1\right],$$

where the $\mathbf{Exp}_{\mathsf{BS}}^{\text{BDres}_k}(\mathcal{A})$, $k \in \mathbb{N}$, experiment is defined in the following way:

$\mathbf{Exp}_{\mathsf{BS}}^{\text{BDres}_k}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $\mathsf{lost} \leftarrow \mathsf{false}$
4 : $st \leftarrow \mathcal{A}_1(\mathsf{sk}, \mathsf{pk})$
5 : $(m, \sigma) \xleftarrow{\$} \mathcal{A}_2^{Sign}(\mathsf{pk})$
6 : **if** $((m, \sigma) \in \mathcal{L}) \vee (\mathsf{lost} = \mathsf{true})$:
7 : **return** 0
8 : **return** $\mathsf{BS.Vf}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

1 : $i \leftarrow 0$
2 : **do**
3 : $(st, \sigma) \leftarrow \langle \mathcal{A}_1(st), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
4 : $i \leftarrow i + 1$
5 : **until** $(i \geqslant k) \vee (\sigma \neq \perp)$
6 : **if** $\sigma = \perp$:
7 : $\mathsf{lost} \leftarrow \mathsf{true}$
8 : **return** $\perp$
9 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
10 : **return** $\sigma$

At the experiment initialization stage (line 1), the challenger modeling an honest application generates a key pair $(\mathsf{sk}, \mathsf{pk})$ according to the key generation algorithm and sends to $\mathcal{A}_1$ a pair $(\mathsf{sk}, \mathsf{pk})$ (line 4), while to $\mathcal{A}_2$ it sends a verification key $\mathsf{pk}$ only (line 5). This stage models the trusted process of generating keys, issuing corresponding certificate and uploading key material onto the smart card.

The $\mathcal{A}_2$ algorithm can make queries to the challenger signing oracle $Sign$ that returns signature values $\sigma$ for messages $m$ arbitrarily chosen by the adversary. Each signature value is computed during the execution of the blind signing protocol between oracle that models the honest User side and the $\mathcal{A}_1$ algorithm modeling the malicious Signer side (line 3 in the oracle). Here the variable $st$ denotes the internal state of $\mathcal{A}_1$ that is kept from call to call.

The $\mathcal{A}_1$ algorithm is allowed to terminate the protocol execution with an error $\perp$ on the User side (line 5 in the oracle). For this reason, for any requested message $m$ the oracle makes $k$ attempts to compute a correct signature, and in the case when all $k$ attempts fail, challenger returns 0 as a game result (meaning that the adversary loses, see line 7 in the oracle). This simulates the scenario where the smart card has failed and is no longer used.

**Remark 3.** Note that if the algorithm $\mathcal{A}_2$ can obtain errors from the signing oracle, then there is always a trivial attack. Consider the agent $\mathcal{A}_1$ that successfully completes the signing protocol execution iff $i$-th bit of $\mathsf{sk}$ is equal to 1, where $i$ is a sequence number of query to oracle. Having such an agent on the smart card side, the $\mathcal{A}_2$ algorithm can recover all bits of signing key and trivially make a forgery.

To break a backdoor resilience, the algorithm $\mathcal{A}_2$ is needed to make a forgery $(m, \sigma)$ containing a signature $\sigma$ that has not previously been returned by the oracle $Sign$ in response to a query $m$.

**Honest-but-curious unforgeability/Security against external adversary**

This notion considers only an honest-but-curious adversary acting on the User side. This adversary can adaptively choose messages to be signed by making a query $m$ to the oracle and obtain in return a signature $\sigma$ and a specific value $view$. The latter consits of all incoming messages and the values of all random parameters processed and sampled by the User side during the execution of the signing protocol. This simulates the scenario, where the adversary gets an access to the memory of trusted application.

The formal definition of HBC-UF is given below.

**Definition 4.** For an adversary $\mathcal{A}$ and a blind signature scheme $\mathsf{BS}$:

$$\mathsf{Adv}^{\text{HBC-UF}}_{\mathsf{BS}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{HBC-UF}}_{\mathsf{BS}}(\mathcal{A}) \to 1\right],$$

where the $\mathbf{Exp}^{\text{HBC-UF}}_{\mathsf{BS}}(\mathcal{A})$ experiment is defined in the following way:

$\mathbf{Exp}^{\text{HBC-UF}}_{\mathsf{BS}}(\mathcal{A})$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $(m, \sigma) \leftarrow \mathcal{A}^{Sign}(\mathsf{pk})$
4 : **if** $(m, \sigma) \in \mathcal{L}$ : **return** 0
5 : **return** $\mathsf{BS.Vf}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

1 : $(1, (\sigma; view)) \leftarrow \langle \mathsf{BS.Signer}(\mathsf{sk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
2 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
3 : **return** $\sigma, view$

It is easy to see that for any blind signature scheme HBC-UF-security implies sUF-CMA-security.

## 4. Security analysis

### 4.1. Backdoor resilience / Security against adversary with agent

In this section, we prove that honest-signer blindness and standard unforgeability (sUF-CMA) imply backdoor resilience.

**Theorem 1.** Fix $k \in \mathbb{N}$. For any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the $\mathrm{BDres}_k$ model with summary time complexity at most $t$ making at most $q$ queries to the signing oracle, there exist an adversary $\mathcal{B}$ in the sUF-CMA model making at most $q$ queries to the signing oracle and an adversary $\mathcal{C}$ in the HS-Blind model such that

$$\mathsf{Adv}_{\mathsf{BS}}^{\mathrm{BDres}_k}(\mathcal{A}) \leqslant \mathsf{Adv}_{\mathsf{BS}}^{\text{sUF-CMA}}(\mathcal{B}) + q \cdot k \cdot \mathsf{Adv}_{\mathsf{BS}}^{\text{HS-Blind}}(\mathcal{C}).$$

Time complexities of $\mathcal{B}$ and $\mathcal{C}$ are at most $t$ and $tkq$ correspondingly.

**Remark 4.** If the blind signature scheme provides perfect blindness (i.e., $\mathsf{Adv}_{\mathsf{BS}}^{\text{HS-Blind}}(\mathcal{C}) = 0$ for any $\mathcal{C}$ with any time complexity), then the bound is transformed as follows:

$$\mathsf{Adv}_{\mathsf{BS}}^{\mathrm{BDres}_k}(\mathcal{A}) \leqslant \mathsf{Adv}_{\mathsf{BS}}^{\text{sUF-CMA}}(\mathcal{B}).$$

From the perspective of using conventional signature scheme $\mathsf{SS}$, this inequality means that in order to provide backdoor resilience, it is enough for this signature scheme to have its blind version $\mathsf{BS}$ (with $\mathsf{Adv}_{\mathsf{BS}}^{\text{sUF-CMA}}(\mathcal{B}) = \mathsf{Adv}_{\mathsf{SS}}^{\text{sUF-CMA}}(\mathcal{B})$) and to be unforgable in the standard model. Note, that the bound does not depend on $k$, so this value can be chosen arbitrarily by the application developers.

**Remark 5.** For clarity, the proof is carried out for three-move blind signatures, but the proof does not base on any specific features of such scheme type and can be easily adapted for any-move blind signatures.

***Proof.*** The proof consits of two parts.

Part 1. Consider the consequence of several experiments, where each next experiment slightly differs from the previous one.

Game 0. Let $\mathbf{Exp}_{\mathsf{BS}}^{0}(\mathcal{A}) = \mathbf{Exp}_{\mathsf{BS}}^{\mathrm{BDres}_k}(\mathcal{A})$.

Game 1. Consider the following modified experiment $\mathbf{Exp}_{\mathsf{BS}}^{1}(\mathcal{A})$:

$\mathbf{Exp}_{\mathsf{BS}}^{1}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$

$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
$\mathcal{L} \leftarrow \varnothing$
$\mathsf{lost} \leftarrow \mathsf{false}$
$st \leftarrow \mathcal{A}_1(\mathsf{sk}, \mathsf{pk})$
$(m, \sigma) \xleftarrow{\$} \mathcal{A}_2^{Sign}(\mathsf{pk})$
**if** $((m, \sigma) \in \mathcal{L}) \vee (\mathsf{lost} = \mathsf{true})$:
  **return** 0
**return** $\mathsf{BS.Vf}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

$i \leftarrow 0$
**do**
  $(st, \sigma) \leftarrow \langle \mathcal{A}_1(st), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
  **if** $\sigma \neq \perp$:
    $(1, \sigma) \leftarrow \langle \mathsf{BS.Signer}(\mathsf{sk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
  $i \leftarrow i + 1$
**until** $(i \geqslant k) \vee (\sigma \neq \perp)$
**if** $\sigma = \perp$:
  $\mathsf{lost} \leftarrow \mathsf{true}$
  **return** $\perp$
$\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
**return** $\sigma$

$\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A})$ differs from $\mathbf{Exp}^0_{\mathsf{BS}}(\mathcal{A})$ in additional lines 4 and 5 of the $Sign$ oracle code. If the oracle, interacting with the agent $\mathcal{A}_1$ as a user, completes the signing protocol with a correct signature, then the oracle recomputes a new signature honestly executing the signing protocol on its own (without interaction with the agent). The second part of the proof is devoted to estimation of winning probability difference for $\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A})$ and $\mathbf{Exp}^0_{\mathsf{BS}}(\mathcal{A})$ .

Game 2. Consider the next modification: the experiment $\mathbf{Exp}^2_{\mathsf{BS}}(\mathcal{A})$. Here the oracle always responses to requests of $\mathcal{A}_2$ with a correct honestly generated signature even in the case when $\mathcal{A}_1$ provokes errors $k$ times in a row that sets the flag $\mathsf{lost}$ in $\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A})$.

$\mathbf{Exp}^2_{\mathsf{BS}}(\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2))$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $st \leftarrow \mathcal{A}_1(\mathsf{sk}, \mathsf{pk})$
4 : $(m, \sigma) \xleftarrow{\$} \mathcal{A}_2^{Sign}(\mathsf{pk})$
5 : **if** $((m, \sigma) \in \mathcal{L})$:
6 : **return** 0
7 : **return** $\mathsf{BS.Vf}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

1 : $i \leftarrow 0$
2 : **do**
3 : $(st, \sigma) \leftarrow \langle \mathcal{A}_1(st), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
4 : $i \leftarrow i + 1$
5 : **until** $(i \geqslant k) \vee (\sigma \neq \perp)$
6 : $(1, \sigma) \leftarrow \langle \mathsf{BS.Signer}(\mathsf{sk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
7 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
8 : **return** $\sigma$

For this experiment:

$$\Pr\left[\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A}) \to 1\right] = \Pr\left[\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A}) \to 1 \wedge (\mathsf{lost} = \mathsf{false})\right] + \\ + \underbrace{\Pr\left[\mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A}) \to 1 \wedge (\mathsf{lost} = \mathsf{true})\right]}_{= 0 \text{ due to line 6 of } \mathbf{Exp}^1_{\mathsf{BS}}(\mathcal{A})} \leqslant \Pr\left[\mathbf{Exp}^2_{\mathsf{BS}}(\mathcal{A}) \to 1\right].$$

Game 3. Note that in the $\mathbf{Exp}^2_{\mathsf{BS}}(\mathcal{A})$ experiment the agent $\mathcal{A}_1$ can be thrown away, since it can no longer influence the value of the signature (see the $\mathbf{Exp}^3_{\mathsf{BS}}(\mathcal{A}_2)$ experiment below). Note that $\Pr\left[\mathbf{Exp}^2_{\mathsf{BS}}(\mathcal{A}_1, \mathcal{A}_2) \to 1\right] = \Pr\left[\mathbf{Exp}^3_{\mathsf{BS}}(\mathcal{A}_2) \to 1\right]$.

$\mathbf{Exp}^3_{\mathsf{BS}}(\mathcal{A}_2)$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $\mathcal{L} \leftarrow \varnothing$
3 : $(m, \sigma) \leftarrow \mathcal{A}_2^{Sign}(\mathsf{pk})$
4 : **if** $(m, \sigma) \in \mathcal{L}$:
5 : **return** 0
6 : **return** $\mathsf{BS.Vf}(\mathsf{pk}, m, \sigma)$

Oracle $Sign(m)$

1 : $(1, \sigma) \leftarrow \langle \mathsf{BS.Signer}(\mathsf{sk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
2 : $\mathcal{L} \leftarrow \mathcal{L} \cup \{(m, \sigma)\}$
3 : **return** $\sigma$

Note that $\mathbf{Exp}^3_{\mathsf{BS}}$ is exactly the experiment $\mathbf{Exp}^{\text{sUF-CMA}}_{\mathsf{BS}}$, therefore $\Pr\left[\mathbf{Exp}^2_{\mathsf{BS}}(\mathcal{A}) \to 1\right] \leqslant$ $\leqslant \mathsf{Adv}^{\text{sUF-CMA}}_{\mathsf{BS}}(\mathcal{B})$ for $\mathcal{B} = \mathcal{A}_2$.

P a r t 2 . To finalize the proof, we construct an adversary $\mathcal{C}$ breaking the blindness property. Introduce the following auxiliary experiment:

$\mathbf{Exp}^{4,b}_{\mathsf{BS}}(\mathcal{C})$

1 : $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{BS.KGen}()$
2 : $b' \xleftarrow{\$} \mathcal{C}^{Init, User_1, User_2}(\mathsf{sk}, \mathsf{pk})$
3 : **return** $b'$

Oracle $Init(m)$

1 : $sess \leftarrow \mathsf{init}$

Oracle $User_1(msg)$

1 : **if** $sess \neq \mathsf{init}$ : **return** $\perp$
2 : $sess \leftarrow \mathsf{open}$
3 : $(msg, state) \leftarrow \mathsf{BS.User}_1((\mathsf{pk}, m), msg)$
4 : **return** $msg$

Oracle $User_2(msg)$

1 : **if** $sess \neq \mathsf{open}$ : **return** $\perp$
2 : $\sigma \leftarrow \mathsf{BS.User}_2(state, msg)$
3 : **if** $(\sigma \neq \perp) \wedge (b = 0)$:
4 : $\quad (1, \sigma) \leftarrow \langle \mathsf{BS.Signer}(\mathsf{sk}), \mathsf{BS.User}(\mathsf{pk}, m) \rangle$
5 : **return** $\sigma$

Here an adversary can make only one query to each oracle (execute only one session). The adversary obtains a signature value generated by the oracles intacting with adversary if $b = 1$, and a signature computed according to the protocol otherwise. Note that if the adversary provokes an error in the session, then it always gets $\perp$ from the $User_2$ oracle regardless of bit $b$.

Using a standard technique called "hybrid argument" [10], it can be trivially shown that there exists an adversary $\mathcal{C}'$ such that

$$\Pr\left[\mathbf{Exp}^{0}_{\mathsf{BS}}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^{1}_{\mathsf{BS}}(\mathcal{A}) \to 1\right] = \\ = q \cdot k \left(\Pr\left[\mathbf{Exp}^{4,1}_{\mathsf{BS}}(\mathcal{C}') \to 1\right] - \Pr\left[\mathbf{Exp}^{4,0}_{\mathsf{BS}}(\mathcal{C}') \to 1\right]\right).$$

Now let construct an adversary $\mathcal{C}$ using $\mathcal{C}'$ as a black box. The adversary $\mathcal{C}$ acts in the following way:

1) The adversary $\mathcal{C}$ obtains $(\mathsf{sk}, \mathsf{pk})$ and transmits this value to $\mathcal{C}'$.
2) When $\mathcal{C}'$ makes a query $m$ to the $Init$ oracle, $\mathcal{C}$ makes a query $(m, m)$ to its own $Init$ oracle.
3) After starting sessions, the adversary $\mathcal{C}$ firstly executes $sess_0$ according to the protocols:
   a) it computes $(msg^0_{S,1}, state_S) \leftarrow \mathsf{BS.Signer}_1(\mathsf{sk})$ and makes a query $(0, msg^0_{S,1})$ to its own $User_1$ oracle;
   б) upon receiving $msg^0_{U,1}$, the adversary $\mathcal{C}$ computes

$$(msg^0_{S,2}, 1) \leftarrow \mathsf{BS.Signer}_2(state_S, msg_{U,1})$$

   and makes a query $(0, msg^0_{S,2})$ to its own $User_2$ oracle, receiving the $\varepsilon$ value.
   Note that $\sigma_{b_0} \neq \perp$ due to the correctness property of the blind signature scheme.
4) Then the adversary $\mathcal{C}$ intercepts all queries of $\mathcal{C}'$ and simply passes them to $sess_1$:
   a) intercepting from $\mathcal{C}'$ a query $msg_1$ to the $User_1$ oracle, $\mathcal{C}$ makes a query $(1, msg^1_{S,1})$, where $msg^1_{S,1} = msg_1$, to its own $User_1$ oracle and directly transmits the response $msg^1_{U,1}$ to $\mathcal{C}'$;
   б) intercepting from $\mathcal{C}'$ a query $msg_2$ to the $User_2$ oracle, $\mathcal{C}$ makes a query $(1, msg^1_{S,2})$, where $msg^1_{S,2} = msg_2$, to its own $User_2$ oracle. $\mathcal{C}$ receives $(\sigma_0, \sigma_1)$ and returns to $\mathcal{C}'$ the fist component $\sigma_0$. Note that $(\sigma_0, \sigma_1)$ can be $(\perp, \perp)$.
5) $\mathcal{C}$ returns the same bit as $\mathcal{C}'$ returns.

If the $\mathcal{C}$ interacts with the experimentator $\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},1}$ ($\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},0}$), then $\sigma_0 = \sigma_{b_1}$ ($\sigma_0 = \sigma_{b_0}$). Moreover, $\mathcal{C}$ returns $\perp$ at stage 4 iff $\mathcal{C}'$ provokes error in $sess_1$ that perfectly coincides with $\mathbf{Exp}_{\mathsf{BS}}^{4}$. Thus,

$$\Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{4,1}(\mathcal{C}') \to 1\right] = \Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},1}(\mathcal{C}) \to 1\right],$$
$$\Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{4,0}(\mathcal{C}') \to 1\right] = \Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},0}(\mathcal{C}) \to 1\right].$$

Summing up,

$$\Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{0}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{1}(\mathcal{A}) \to 1\right] = q \cdot k \left(\Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{4,1}(\mathcal{C}') \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{4,0}(\mathcal{C}') \to 1\right]\right) =$$
$$= q \cdot k \left(\Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},1}(\mathcal{C}) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathsf{BS}}^{\text{HS-Blind},0}(\mathcal{C}) \to 1\right]\right) = q \cdot k \cdot \mathsf{Adv}_{\mathsf{BS}}^{\text{HS-Blind}}(\mathcal{C}).$$

The theorem 1 is proven. ■

### 4.2. Honest-but-curious unforgeability / Security against external adversary

Here we define the particular class of blind signature schemes based on ElGamal signature equation that provides the honest-but-curious unforgeability. Namely, for such schemes we construct the security reduction to the unforgeability of the base ElGamal signature scheme. Note that all known ElGamal blind signature schemes do not provide strong unforgeability [11].

At first, let us introduce the required notations. We denote the group of points of the elliptic curve over the prime field as $\mathbb{G}$, the order of the prime subgroup of $\mathbb{G}$ as $q$, an elliptic curve point of order $q$ as $P$ and zero point as $\mathcal{O}$. We denote by $H$ the hash function that maps binary strings to elements from $\mathbb{Z}_q$ and assume that all field operations are performed modulo $q$.

**ElGamal blind signature scheme**

The generalized ElGamal signature scheme was introduced in [12] and further extended in [13], we denote it by GenEG scheme. A key generation algorithm in this scheme involves picking random $d$ uniformly from $\mathbb{Z}_q^*$ (secret signing key) and defining $Q = dP$ (public verifying key). A signature for message $m$ is a pair $(r, s)$, where $r = (kP).x \bmod q$ for some $k$ picked uniformly at random from $\mathbb{Z}_q^*$ and $s$ is computed from the ElGamal signature equation $EG$:

$$EG(d, k, r, e, s) = 0,$$

where $e = H(m)$. All possible $EG$ equations are listed in [12]. To ensure functionality and security, certain $r, e, s$ values need to be excluded.

ElGamal blind signature scheme, denoted by GenEG-BS, was introduced in [11]. A key generation and verification algorithms in GenEG-BS scheme are the same as in the base GenEG scheme. An interactive signing protocol assumes that the Signer performs ElGamal signature generating algorithm for the $e$ value received from the User, the User algorithm is not determined and can be arbitrary. The parameters of the signing protocol are the base point $P$, public key $Q$, and the message $m$, we denote them by $par$.

We impose the additional requirements on the algorithm performed by the User:

— all blinding factors (we denote them by $rnd$) used by the User are selected according to some distribution $\mathcal{D}$ that is independent on the values received from the Signer;

— the first component of the signature $r'$ is the $x$-coordinate of the $R'$ point, which is computed as a result of applying the function parameterized by the $par$ value (we denote it by $\mathcal{L}_1^{par}$) that takes as arguments the $R$ point received from the Signer and $rnd$ values. This function is linear by $R$ for all $rnd$ values generated according to the protocol;
— the second component of the signature $s'$ is computed as a result of applying the function parameterized by the $par$ value (we denote it by $\mathcal{L}_2^{par}$) that takes as arguments the $s$ value received from the Signer, $rnd$ values, and point $R$. This function is linear by $s$ for all $rnd$ and $R$ values generated according to the protocol.

We denote such a scheme by $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$ scheme. The corresponding signing protocol is illustrated in Fig. 1.

The signing protocol

| **Signer**$(d)$ | | **User**$(Q, m)$ |
|---|---|---|
| $k \xleftarrow{\mathcal{U}} \mathbb{Z}_q^*$ | | |
| $R \leftarrow kP$ | $\xrightarrow{\quad R \quad}$ | $rnd \xleftarrow{\mathcal{D}}$ |
| | | $R' \leftarrow \mathcal{L}_1^{par}(R, rnd)$ |
| | | $r' \leftarrow R'.x \bmod q$ |
| $r \leftarrow R.x \bmod q$ | $\xleftarrow{\quad e \quad}$ | compute $e$ |
| **if** $\exists!\ s\colon EG(d, k, r, e, s) = 0$ find $s$ | | |
| **else** : **return** 0 | $\xrightarrow{\quad s \quad}$ | |
| | | $s' \leftarrow \mathcal{L}_2^{par}(s, rnd, R)$ |
| **return** 1 | | **return** $(r', s')$ |

Fig. 1. The signing protocol in $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$ scheme

Let us show that the $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$ scheme is indeed the blind version of the GenEG scheme, i.e., provides the same distribution on the signature values. The distribution on GenEG signatures is defined by the uniform distribution on $k$ values. The distribution on $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$ signatures is defined by the distribution on $k'$ values, where $k'$ is such that $(k'P).x \bmod q = r'$. The $k'$ value is linear by $k$ since $R'$ value is linear by $R$ and $rnd$ values are chosen independently on $R$. Thus, the distribution on $k'$ values is also uniform.

Note that the User view in the $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$ scheme consists of the incoming messages $R, s$ and the blinding factors $rnd$ sampled by the User.

Now we are ready to construct the security reduction to the unforgeability of the conventional ElGamal signature scheme.

**Theorem 2.** For any adversary $\mathcal{A}$ for $\mathsf{GenEG\text{-}BS}_{\mathcal{L}}$ scheme in the HBC-UF model with time complexity at most $t$ making at most $q$ queries to the signing oracle, there exist an adversary $\mathcal{B}$ for the conventional GenEG scheme in the sUF-CMA model with the same time complexity at most $t$ making at most $q$ queries to the signing oracle such that

$$\mathsf{Adv}^{\text{HBC-UF}}_{\mathsf{GenEG\text{-}BS}_{\mathcal{L}}}(\mathcal{A}) \leqslant \mathsf{Adv}^{\text{sUF-CMA}}_{\mathsf{GenEG}}(\mathcal{B}).$$

***Proof.*** Let construct the adversary $\mathcal{B}$ for the conventional GenEG scheme. The adversary $\mathcal{B}$ uses the adversary $\mathcal{A}$ as a black box. It intercepts the queries of the adversary $\mathcal{A}$ to the signing oracle and process them by itself using its own signing oracle in the following way.

Receiving the query $m$, adversary $\mathcal{B}$ forwards $m$ to its own oracle and receives the signature $(r', s')$. Then it reconstructs $R'$ point from the verification algorithm and selects $rnd$ value according to the distribution $\mathcal{D}$. After that, it calculates the $R$ value using $\mathcal{L}_1^{-1}$ function and $s$ value using $\mathcal{L}_2^{-1}$ function. It returns as an answer the signature $(r', s')$ and the $view = (R, s, rnd)$.

Note that $\mathcal{B}$ generates exactly the same distribution on signature values since GenEG-BS$_{\mathcal{L}}$ scheme is the blind version of the GenEG scheme. The $rnd$ value is chosen as in the honest execution of blind signature protocol, $R$ and $s$ values are also computed as in the honest execution, since $\mathcal{L}_1$ and $\mathcal{L}_2$ functions are unambiguously invertible.

When $\mathcal{A}$ returns a forgery, $\mathcal{B}$ translates it to its own challenger and stops. Obviously, if $\mathcal{A}$ wins, then $\mathcal{B}$ wins, whence follows the statement of the theorem. ■

**Remark 6.** The same result may be formulated for the Schnorr signature scheme and its blind version defined in [5]. The proof of the theorem is conducted in the same way.

## 5. GOST-based blind signature scheme

We propose to use the concrete blind signature scheme in case of building the protection for GOST signature scheme [4]. This scheme was proposed in [6] in 1994 and is commonly referred to as the Camenisch scheme. We provide the definition of this scheme in terms of elliptic group notation.

The key generation algorithm is the same as in the general ElGamal signature scheme and assumes picking secret key $d$ uniformly from $\mathbb{Z}_q^*$ and defining public key $Q$ as $dP$. The signing protocol is defined in Fig. 2. The verification procedure for the message $m$ and the signature $(r', s')$ assumes checking $r' \neq 0$ and checking the equality $r' = R'.x \bmod q$, where $R' = (e')^{-1}(s'P - r'Q)$, $e'$ is equal to $H(m)$, if $H(m) \neq 0$, and to 1 otherwise. Note that the signing protocol in Fig. 2 is defined for the case of using the elliptic curves of the prime order. Nevertheless, it can be slightly modified by adding the additional checks for use with non-prime order curves, e.g. with Edwards curves.

This scheme provides perfect blindness [6, Theorem 2], but does not provide unforgeability in the strong sense. In [11] it was shown that it is vulnerable to the ROS-style attack, which is possible if the adversary acting as a User is given the opportunity to open $\ell \geqslant \lceil \log q \rceil$ parallel sessions of signing protocol. However, providing such strong unforgeability is not required for our application, our purpose is the honest-but-curious unforgeability.

Camenisch scheme is the particular case of the GenEG-BS$_{\mathcal{L}}$ scheme defined in Section 4.2. Indeed, the distribution $\mathcal{D}$ in this scheme is a uniform distribution on $\mathbb{Z}_q^* \times \mathbb{Z}_q^*$ that is independent on $R$; $\mathcal{L}_1^{(P,Q,m)}$ and $\mathcal{L}_2^{(P,Q,m)}$ are defined as follows:

$$\mathcal{L}_1^{(P,Q,m)}(R, (\alpha, \beta)) = \alpha R + \beta P, \quad \mathcal{L}_2^{(P,Q,m)}(s, (\alpha, \beta), R) = sr'r^{-1} + \beta e',$$

where $e' = H(m)$, $r = R.x \bmod q$, $r' = (\alpha R + \beta P).x \bmod q$. These functions are linear by $R$ and $s$ values respectively for all possible $rnd$ values. Moreover, zero $r$ and $e$ values are excluded by the corresponding checks on the Signer side as in the GOST signature scheme. Therefore, the result of Theorem 2 is applied to the Camenisch scheme, which means that it provides honest-but-curious unforgeability under the assumption that GOST

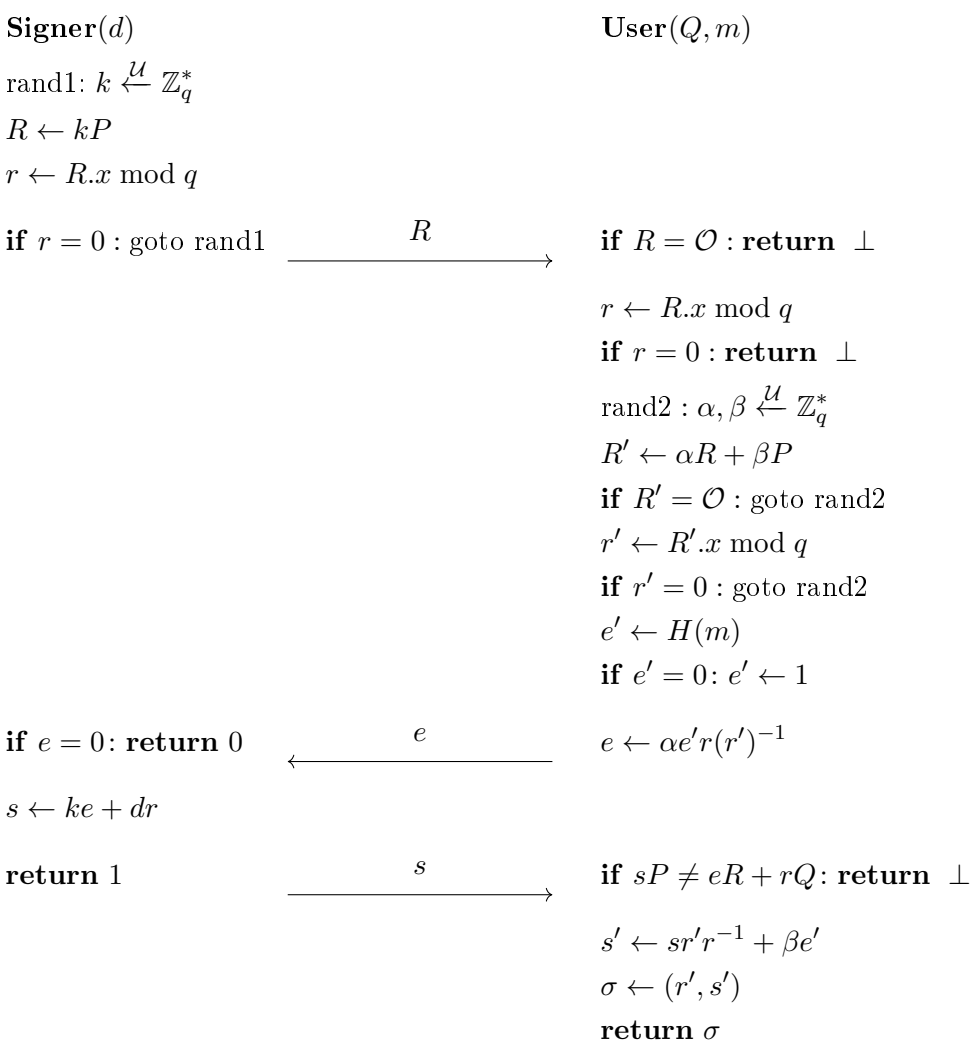


Fig. 2. The signing protocol in Camenisch scheme

scheme provides unforgeability. The security of the Camenisch scheme in the sUF-CMA model, in its turn, directly follows from the honest-but-curious unforgeability.

Thus, the Camenisch scheme is a blind version of the GOST scheme and can be applied in the systems realizing the GOST signature as the protection against backdoors in smart cards. It provides the security against external adversary and adversary with agent only by the security of the GOST signature scheme. Note that such solution, in contrast to the solution from [2], does not need any additional assumptions about the smart card such as correct implementation of low-level arithmetic operations and the absence of failures. Moreover, it requires less computations on the smart card side.

## 6. Conclusion

The paper addressed the security issues that arise in signing systems when the smart card used for key storage and signing is believed to contain backdoors. A novel approach based on blind signature schemes to protect against backdoors has been proposed. It has been proven that weak versions of standard security properties (honest-signer blindness and honest-but-curious unforgeability) of blind signature scheme imply security against backdoors in smart cards.

Moreover, the concrete solution in case of using the GOST signature scheme has been proposed. This solution is the well known Camenisch blind signature scheme that provides perfect blindness. It was shown that the target security is held under the sole assumption that the GOST signature scheme provides standard security, i.e., is unforgeable under chosen message attack.

One of the most interesting directions for future research is the security analysis of our solution with regard to a stronger external adversary — an active adversary that has an access to a smart card signing API (e.g. in a case when the smart card is not protected with a password or is connected to a malicious terminal).

This case corresponds to the standard unforgeability notion of the blind signatures, where the user side is treated as a fully active adversary. There are two types of unforgeability notion differing on whether the adversary can open sessions in parallel or not. In our application scenario, where the signer side is executed by low resource device, it is fairly enough to consider the adversary's capability to open sessions sequentially only (this refers to the SEQ-OMUF notion [14]).

Note that the SEQ-OMUF-security of the Camenish scheme is still an open question (as well as for the most ElGamal blind signature schemes), although there have been some positive results [14] for the Schnorr blind signature scheme.

## REFERENCES

1. *Alekseev E. K., Akhmetzyanova L. R., Oshkin I. B., and Smyshlyaev S. V.* Obzor uyazvimostey nekotorykh protokolov vyrabotki obshchego klyucha s autentifikatsiey na osnove parolya i printsipy postroeniya protokola SESPAKE [A review of the password authenticated key exchange protocols vulnerabilities and principles of the SESPAKE protocol construction]. Matematicheskie Voprosy Kriptografii, 2016, vol. 7, iss. 4, pp. 7–28. (in Russian)
2. *Alekseev E. K., Akhmetzyanova L. R., Bozhko A. A., and Smyshlyaev S. V.* Bezopasnaya realizatsiya elektronnoy podpisi s ispol'zovaniem slabodoverennogo vychislitelya [Secure implementation of digital signature using semi-trusted computational core]. Matematicheskie Voprosy Kriptografii, 2021, vol. 12, iss. 4, pp. 5–23. (in Russian)
3. *Wang Y.* Password protected smart card and memory stick authentication against off-line dictionary attacks. D. Critzalis, S. Furnell, and M. Theoharidou (eds.), Information Security and Privacy Research, Berlin, Heidelberg, Springer, 2012, pp. 489–500.
4. GOST R 34.10-2012. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Protsessy formirovaniya i proverki elektronnoy tsifrovoy podpisi. [GOST R 34.10-2012. Information Technology. Cryptographic Data Security. Signature and Verification Processes of Electronic Digital Signature]. Moscow, Standartinform Publ., 2012. (in Russian)
5. *Chaum D.* Blind signatures for untraceable payments. D. Chaum, R. L. Rivest, and A. T. Sherman (eds.) Advances in Cryptology. Boston, MA, Springer, 1983. pp. 199–203.
6. *Camenisch J. L., Piveteau J. M., and Stadler M. A.* Blind signatures based on the discrete logarithm problem. LNCS, 1995, vol. 950, pp. 428–432.
7. *Bellare M. and Rogaway P.* The security of triple encryption and a framework for code-based game-playing proofs. LNCS, 2006, vol. 4004, pp. 409–426.
8. *Tessaro S. and Zhu C.* Short pairing-free blind signatures with exponential security. LNCS, 2022, vol. 13276, pp. 782–811.
9. *Juels A., Luby M., and Ostrovsky R.* Security of blind digital signatures. LNCS, 1997, vol. 1294, pp. 150–164.
10. *Fischlin M. and Mittelbach A.* An Overview of the Hybrid Argument. Cryptology ePrint Archive, paper 2021/088, `https://eprint.iacr.org/2021/088`, 2021.

11. *Akhmetzyanova L., Alekseev E., Babueva A., and Smyshlyaev S.* On the (im)possibility of ElGamal blind signatures. Cryptology ePrint Archive, paper 2022/1128, `https://eprint.iacr.org/2022/1128`, 2022.
12. *Harn L. and Xu Y.* Design of generalised ElGamal type digital signature schemes based on discrete logarithm. Electronics Letters, 1994, vol. 30, pp. 2025–2026.
13. *Fersch M.* The provable security of Elgamal-type signature schemes. Doctoral Thesis, Ruhr-Universität Bochum, 2018.
14. *Kastner J., Loss J., and Xu J.* On pairing-free blind signature schemes in the algebraic group model. LNCS, 2022, vol. 13178, pp. 468–497.