On blindness of several ElGamal-type blind signatures

Alexandra Babueva$^{1[0000-0003-4223-7746]}$, Liliya Akhmetzyanova$^{1[0000-0001-6544-9288]}$, Evgeny Alekseev$^{1[0000-0002-1279-0359]}$, and Oleg Taraskin$^{2[0000-0001-7967-1525]}$

[1] CryptoPro LLC, Moscow, Russia
{babueva, lah, alekseev}@cryptopro.ru
[2] Waves, Moscow, Russia
otaraskin@web3tech.ru

Abstract. Blind signature schemes are the essential element of many complex information systems such as e-cash and e-voting systems. They should provide two security properties: unforgeability and blindness. The former one is standard for all signature schemes and ensures that a valid signature can be generated only during the interaction with the secret signing key holder. The latter one is more specific for this class of signature schemes and means that there is no way to link a (message, signature) pair to the certain execution of the signing protocol. In the current paper we discuss the blindness property and various security notions formalizing this property. We analyze several ElGamal-type blind signature schemes regarding blindness. We present effective attacks violating blindness on three schemes. All the presented attacks may be performed by any external observer and do not require signing key knowledge. One of the schemes conceivably became broken due to an incorrect understanding of blindness property.

Keywords: Blind signature scheme · Blindness.

1 Introduction

The blind signature mechanism was originally proposed by Chaum in 1982 in [1] for e-cash systems. Signature issuing protocol is the interactive protocol that runs between two parties: a Signer and a Requester. As the result, the Requester obtains the signature for some message, at that the Signer does not receive any information about either the message or the signature value. The application of blind signature schemes includes electronic voting systems, anonymous e-cash systems, direct anonymous attestation, anonymous credentials, etc.

Blind signature schemes should provide two security properties: unforgeability and blindness. The first one is standard for all signature schemes and ensures that a valid signature can be generated only during the interaction with the secret signing key holder. The second property is more specific for this class of signature schemes and provides that a Signer learns no additional information during the protocol execution. However, the way to determine this information is not obvious. Intuitively it seems that the message to be signed should be hidden from Signer, but it turns out that this is not enough.

In the current paper we discuss the blindness property and analyze several ElGamal-type blind signature schemes regarding this property. We present attacks violating blindness on schemes introduced in [2–4]. It seems that one of them [3] became broken due to an incorrect understanding of blindness property.

2 Blindness property

Before talking about blindness we recall the definition of blind signature scheme. It is determined by three algorithms:

– $(sk, pk) \leftarrow KeyGen(\ )$: a key generation algorithm that outputs a secret key sk and a public key pk;
– $(b, \sigma) \leftarrow \langle Signer(sk), Requester(pk, m)\rangle$: an interactive signing protocol that is run between a Signer with a secret key sk and a Requester with a public key pk and a message m; the Signer outputs $b = 1$ if the interaction completes successfully and $b = 0$ otherwise, while the Requester outputs a signature σ if it terminates correctly, and a fail indicator $\perp$ otherwise.
– $b \leftarrow Verify(pk, m, \sigma)$: a (deterministic) verification algorithm that takes a public key pk, a message m, and a signature σ, and returns 1 if σ is valid on m under pk and 0 otherwise.

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 other words, the blindness is broken if the particular protocol execution for some fixed message leads to fixing the signature value in an unambiguous way or at least to significant narrowing the set of possible signature values. In other words, it means that for each protocol transcription and message there exists only the small set of valid signature values (and hence, blinding factors values) that could be produced during such protocol execution.

For deeper understanding we consider the example of using blind signature schemes in e-voting systems. Suppose, that the authenticated voter performs a blind signature protocol with the Registrar and receives a signature for his ballot (the ballot acts as the message in this scenario). Note that in this case the transcription of the protocol is tied to a specific person, his full name and personal information. After receiving the signature, the voter sends a signed ballot to the ballot box anonymously. Thus, if one can link the protocol transcription to the (message, signature) pair, then he can link the ballot to the specific person and violate anonymity.

Towards formalizing. Let describe the regular blindness security notion introduced in [5, 6]. An adversary acts as a malicious Signer and is powered to run the signing protocol with the Requester twice. It is assumed that the Requester behaves correctly (according to the protocol). After two successful interactions the Requester outputs two (message, signature) pairs simultaneously. If at least one of the interactions failed, the Requester outputs fail indicator.

The adversary's task (threat) is to link the transcription to the corresponding (message, signature) pair with success probability significantly greater than 1/2. A strong and a weak attacks may be also distinguished by the following criteria [7]:

– by key generation way (weak attack — the adversary generates key pair according to the protocol, strong — in the malicious way),
– by method of choosing messages, the signature for which the adversary should distinguish (weak attack — the messages are chosen by the Requester, strong — by the adversary).

Note that regular blindness assumes that all interactions terminates successfully. However, extended security notions, that allow an adversary to initiate aborts, were also introduced: a-posteriori blindness [8], selective-failure blindness [9]. The latter notion was also extended to the multiple interaction case [10]. A-posteriori blindness originally considers blindness of multiple executions between the Signer and the Requester, and guarantees unlinkability of execution with (message, signature) pairs only for non-aborted sessions. An adversary is powered to control the distribution on the signed messages, but not to choose them. However, a-posteriori blindness does not imply ordinary blindness and vice versa [8]. Selective-failure blindness guarantees that adversary could not force Requester to abort the signing protocol because of a certain property of the Requester message, which would disclose some information about the message to the adversary. Selective-failure blindness is a strictly stronger notion than regular blindness [10].

3 Broken schemes

This section presents three ElGamal-type blind signature schemes that do not provide blindness and the corresponding attacks. To address specific schemes we name them by the authors' initials and the date of paper publication.

All considered schemes are based on the elliptic curve discrete logarithm problem. If p is a prime number then the set $\mathbb{Z}_p$ is a finite field with characteristic p. We assume the canonic representation of the elements in $\mathbb{Z}_p$ as a natural number in the interval $[0 \dots p-1]$.We define $\mathbb{Z}_p^*$ as the set $\mathbb{Z}_p$ without zero element. We denote the group of points of elliptic curve over the field $\mathbb{Z}_p$ by $\mathbb{G}$, the order of the prime subgroup of $\mathbb{G}$ by q and elliptic curve point of order q by P. For simplicity we assume that $p < q$. A key generation algorithm $KeyGen$ in all schemes involves picking random d from $\mathbb{Z}_q^*$ (secret signing key) and defining $Q = dP$ (public verifying key). 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.

To avoid trivial attacks we assume that during the signing protocol both the Signer and the Requester check that field elements are nonzero, points belong to the used elliptic curve and are not equal to the zero point. Moreover, the Requester should always check that the values obtained from the Signer are valid for its query. If one of these checks fails, the participant should abort the protocol with fail indicator.

All the proposed attacks are applied in the weak security model:

- key pair is generated correctly;
- Requester chooses the messages for signing on its own;
- an adversary does not need to know secret signing key;
- an adversary does not need to initiate aborts on the Requester side.

In fact, all these attacks may be performed by any external observer, not only the Signer.

3.1 BS_GYP16 schemes

Four blind signature schemes, based on ECDSA, GDSA, KCDSA and DSTU schemes, were proposed in [2] in 2016. We will now present the definition of ECDSA-based scheme and attack on it. The attacks on other schemes are constructed similarly.

Scheme description. The signing protocol is defined at Figure 1.

The verify procedure for the message m and the signature (r, s) assumes computing point $R = s^{-1}(rQ + eP)$, where $e = H(m)$, and verifying the equality $R.x \bmod q = r$.

Attack. We show that for fixed protocol transcription and message there exists only the small set of valid signature values that could be produced during the given protocol execution. Indeed, if the protocol transcription (R, e, s) and message m are fixed, then the $r = R.x \bmod q$ and $e' = H(m)$ values are also fixed. The line (1) allows to define the r' component of the signature unambiguously as $r' = re^{-1}e'$ and thus R' point is fixed up to sign. For each possible R' value there exists the unique α value such that $R' = \alpha R$. But α values are selected uniformly at random, so the probability to choose α, such that $(\alpha R).x \bmod q = re^{-1}e'$, during several protocol executions is negligible. Therefore, with overwhelming probability there exist only one signature with r' component satisfied the condition in line (1).

Hence, the line (1) provides the criteria to break the blindness property. The exact transcription (R, e, s) corresponds to the certain message m with signature (r', s') iff the following condition holds:

$$e = r(r')^{-1}e',$$

where $e' = H(m)$.

The signing protocol

$$
\begin{array}{lcl}
\textbf{Signer}(d, Q) & & \textbf{Requester}(Q, m) \\
k \leftarrow_{\$} \mathbb{Z}_q^* & & \\
R \leftarrow kP & & \\
 & \xrightarrow{\quad R \quad} & \\
 & & \alpha \leftarrow_{\$} \mathbb{Z}_q^* \\
 & & R' \leftarrow \alpha R \\
 & & r \leftarrow R.x \bmod q \\
 & & r' \leftarrow R'.x \bmod q \\
 & & e' \leftarrow H(m) \\
 & & e \leftarrow r(r')^{-1}e' \qquad (1) \\
 & \xleftarrow{\quad e \quad} & \\
r \leftarrow R.x \bmod q & & \\
s \leftarrow k^{-1}(dr + e) & & \\
 & \xrightarrow{\quad s \quad} & \\
 & & s' \leftarrow s\alpha^{-1}r'r^{-1} \\
 & & \sigma \leftarrow (r', s')
\end{array}
$$

Fig. 1. BS_GYP16 scheme: the signing protocol.

3.2 BS_R00 scheme

Two blind signature schemes, based on Schnorr and ElGamal (specifically, GOST) signatures, were proposed in [3] in 2000. Both of them are vulnerable to the same attack. Let us show it on the GOST-based blind signature example.

Further we assume that elliptic curve points can be represented as binary strings (corresponding to their coordinates) and therefore may be passed as input to the hash function H.

Scheme description. The signing protocol is defined at Figure 2.

The verify procedure for the message m and the signature (R, s) assumes verifying the equality $sP = H(R)Q + eR$, where $e = H(m)$.

Attack. Similar to the previous scheme, we show that for fixed protocol transcription and message there exists only few valid signatures that could be produced during the given protocol execution. Indeed, if the protocol transcription (R, e, s) and message m are fixed, then the $r = H(R)$ and $e' = H(m)$ values are

The signing protocol

$$\begin{array}{lcll}
\textbf{Signer}(d, Q) & & \textbf{Requester}(Q, m) & \\
k \leftarrow_{\$} \mathbb{Z}_q^* & & & \\
R \leftarrow kP & & & \\
r \leftarrow H(R) & & & \\
 & \xrightarrow{\quad R \quad} & & \\
 & & \alpha \leftarrow_{\$} \mathbb{Z}_q^* & \\
 & & R' \leftarrow \alpha R & (1) \\
 & & r' \leftarrow H(R') & (2) \\
 & & r \leftarrow H(R) & \\
 & & \beta \leftarrow r' r^{-1} & (3) \\
 & & e' \leftarrow H(m) & \\
 & & e \leftarrow \alpha \beta^{-1} e' & (4) \\
 & \xleftarrow{\quad e \quad} & & \\
s \leftarrow ke + dr & & & \\
 & \xrightarrow{\quad s \quad} & & \\
 & & s' \leftarrow s\beta & \\
 & & \sigma \leftarrow (R', s') &
\end{array}$$

Fig. 2. BS_R00 scheme: the signing protocol.

also fixed. Consider the line (4) of the protocol keeping in mind the relations from lines (1)–(3):

$$e = \alpha\beta^{-1}e' = \alpha(r'r^{-1})^{-1}e' = \alpha(r')^{-1}re' = \alpha H(\alpha R)^{-1}re'.$$

The equation $e = \alpha H(\alpha R)^{-1}re'$ for α has only few roots. However, α values are selected uniformly at random, so the probability to choose α, that satisfies the equation above, during several protocol executions is negligible. Therefore, with overwhelming probability there exist only one signature with $R' = \alpha R$ component, for which α satisfies the condition in line (4).

Hence, the criteria for breaking blindness can be constructed from the lines (1)–(4). The exact transcription (R, e, s) corresponds to the certain message m with hash-value e' and signature (R', s') iff the following condition holds:

$$\alpha R = R',$$

where $\alpha = e(e')^{-1}H(R')H(R)^{-1}$.

The attack on Schnorr-based blind signature, proposed in [3], is defined using the same considerations.

Blindness understanding. The attack seems to become possible due to incorrect understanding of blindness property. The authors of [3] considered blindness as the resistance to the attacks that lead to the disclosure of message m after the protocol execution. However, blindness property is much wider. Indeed, the protocol transcription may leak information about the signature value that also may violate blindness.

3.3 BS_TNHV18 scheme

The similar attack is applicable to the aggregate blind signature scheme, that was proposed in 2018 in [4] (more precisely, two cases of Signing protocol differring on the Requester side were proposed). It is also GOST-based scheme. Without loss of generality, we omit aggregation property and present the description of the scheme in case of one Signer. Indeed, the following attack does not need the secret key knowledge and can be performed by anyone who can view the set of protocol transcriptions and the set of generated (message, signature) pairs.

Scheme description. The signing protocol is defined at Figure 3.

The verify procedure for message m and signature (r, s) in both cases assumes computing point $R = e^{-1}sP - e^{-1}rQ$, where $e = H(m)$, and verifying the equality $R.x = r \bmod q$.

Attack. Consider first case of the scheme. As usual, we show that for fixed protocol transcription and message there exists only few valid signatures that could be produced during the given protocol execution. If the protocol transcription (R, r, e, s) and message m are fixed, then the $e' = H(m)$ value is also fixed. Consider the line (4) of the protocol keeping in mind the relations from lines (1)–(3):

$$r = r'\beta^{-1}\alpha = (R'.x \bmod q)\beta^{-1}e(e')^{-1} = ((\beta R + \alpha P).x \bmod q)\beta^{-1}e(e')^{-1} = \\ = ((\beta R + e(e')^{-1}P).x \bmod q)\beta^{-1}e(e')^{-1}.$$

The equation

$$r = ((\beta R + e(e')^{-1}P).x \bmod q)\beta^{-1}e(e')^{-1}$$

for β has only few roots. However, β values are selected uniformly at random, so the probability to choose β, such that the equation above is satisfied, during several protocol executions is negligible. Therefore, with overwhelming probability there exist only one signature with r' component equal to $(\beta R + e(e')^{-1}P).x \bmod q$, for which β satisfies the condition in line (4).

Hence, lines (1)–(4) provide the following criteria for breaking blindness. The exact transcription (R, r, e, s) corresponds to the certain message m with hash-value e' and signature (r', s') iff the following condition holds:

$$R'.x \bmod q = r',$$

The signing protocol

Signer(d, Q)		**Requester**(Q, m)		
		Case 1	Case 2	
$k \leftarrow_{\$} \mathbb{Z}_q^*$				
$R \leftarrow kP$				
	$\xrightarrow{R}$			
		$\alpha, \beta \leftarrow_{\$} \mathbb{Z}_q^*$	$\alpha, \beta \leftarrow_{\$} \mathbb{Z}_q^*$	
		$e' \leftarrow H(m)$	$e' \leftarrow H(m)$	
		$e \leftarrow \alpha e'$	$e \leftarrow \beta e'$	(1)
		$R' \leftarrow \beta R + \alpha P$	$R' \leftarrow \alpha^{-1} R + P + Q$	(2)
		$r' \leftarrow R'.x \bmod q$	$r' \leftarrow R'.x \bmod q$	(3)
		$r \leftarrow r' \beta^{-1} \alpha$	$r \leftarrow \alpha\beta(r' + e')$	(4)
	$\xleftarrow{r, e}$			
$s \leftarrow ke + dr$				
	$\xrightarrow{s}$			
		$s' \leftarrow \beta\alpha^{-1} s + \alpha e'$	$s' \leftarrow \beta^{-1}\alpha^{-1} s + e'$	
		$\sigma \leftarrow (r', s')$	$\sigma \leftarrow (r', s')$	

Fig. 3. BS_TNHV18 scheme: the signing protocol.

where $R' = \beta R + \alpha P,\ \alpha = e(e')^{-1},\ \beta = r' r^{-1} \alpha$.

The attack on the second case of the scheme is justified similarly. The exact transcription (R, r, e, s) corresponds to the certain message m with hash-value e' and signature (r', s') iff the following condition holds:

$$R'.x \bmod q = r',$$

where $R' = \alpha^{-1} R + P + Q,\ \alpha = r\beta^{-1}(r' + e')^{-1},\ \beta = e(e')^{-1}$.

References

1. Chaum, D.: Blind Signatures for Untraceable Payments. In: Chaum, D., Rivest, R. L., Sherman, A. T. (eds.) Advances in Cryptology. Springer, Boston, MA, pp. 199-203 (1983). https://doi.org/10.1007/978-1-4757-0602-4_18
2. Gorbenko, I., Yesina, M., Ponomar, V.: Anonymous electronic signature method. In: 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T). IEEE Press, Kharkiv, Ukraine, pp. 47-50 (2016). https://doi.org/10.1109/INFOCOMMST.2016.7905332

3. Rostovtsev, A. G.: Podpis' "vslepuju" na jellipticheskoj krivoj dlja jelektronnyh deneg [Blind signature on elliptic curve for e-cash]. Information Security Problems. Computer Systems. (1), 40-45 (2000). [in Russian]
4. Tan, D. N., Nam, H. N., Hieu, M. N., Van, H. N.: New Blind Muti-signature Schemes based on ECDLP. International Journal of Electrical and Computer Engineering. 8(2), 1074-1083 (2018). https://doi.org/10.11591/ijece.v8i2.pp1074-1083
5. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures. In: Kaliski, B. S. (eds.) Advances in Cryptology ? CRYPTO '97. CRYPTO 1997. Lecture Notes in Computer Science, vol. 1294. Springer, Berlin, Heidelberg, pp. 150-164 (1997). https://doi.org/10.1007/BFb0052233
6. Pointcheval, D., Stern, J.: Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology. 13(3), 361-396 (2000). https://doi.org/10.1007/s001450010003
7. Okamoto, T.: Efficient Blind and Partially Blind Signatures Without Random Oracles. In: Halevi, S., Rabin, T. (eds.) Theory of Cryptography. TCC 2006. Lecture Notes in Computer Science, vol. 3876. Springer, Berlin, Heidelberg, pp. 80-99 (2006). https://doi.org/10.1007/11681878_5
8. Hazay, C., Katz, J., Koo, C. Y., Lindell, Y.: Concurrently-Secure Blind Signatures Without Random Oracles or Setup Assumptions. In: Vadhan, S. P. (eds) Theory of Cryptography. TCC 2007. Lecture Notes in Computer Science, vol 4392. Springer, Berlin, Heidelberg, pp. 323-341 (2007). https://doi.org/10.1007/978-3-540-70936-7_18
9. Camenisch, J., Neven, G., Shelat, A.: Simulatable Adaptive Oblivious Transfer. In: Naor, M. (eds.) Advances in Cryptology - EUROCRYPT 2007. EUROCRYPT 2007. Lecture Notes in Computer Science, vol. 4515. Springer, Berlin, Heidelberg, pp. 573-590 (2007). https://doi.org/10.1007/978-3-540-72540-4_33
10. Fischlin, M., Schroder, D.: Security of Blind Signatures under Aborts. In: Jarecki, S., Tsudik, G. (eds.) Public Key Cryptography – PKC 2009. PKC 2009. Lecture Notes in Computer Science, vol. 5443. Springer, Berlin, Heidelberg, pp. 297-316 (2009). https://doi.org/10.1007/978-3-642-00468-1_17