Provable Security of Digital Signatures in the Tamper-Proof Device Model

Nick Varnovsky

Information Security Institute,
Lomonosov University, Moscow

1 Introduction

Provable security of practical electronic signature schemes[1] remains one of the most intriguing open problems in mathematical cryptography. Naor and Yung [9] proved that existence of one-way permutations implies existence of e-signature schemes that are secure against existential forgery with respect to chosen-message attack. Rompel [11] showed that for such a security arbitrary one-way function suffices. It is evident that existence of one-way functions is also a necessary condition, therefore from the theoretical point of view the problem reached its final solution. Namely, secure e-signature schemes exist iff there exist one-way functions.

However, e-signature schemes proposed in the papers cited above are far from be practical.

Proofs of security for practical e-signature schemes are known in idealized models only. There exist two such models, with random oracle and with generic group. Random oracle model comes back to Fiat and Shamir [6] who noted that security proof techniques proposed for interactive authentication protocols could be applied to e-signature schemes if a hash function is substituted by a random function. Later this idea was formalized as a random oracle model. In this model any participant of a protocol can query an oracle for a value of random function in any point of its range.

Most of the proposed so far e-signature schemes were shown to be secure in the random oracle model (the first such proof is due to Pointecheval and

[1]There exists commonly used but somewhat misleading term *digital signature*. Note that hand-written signature needs not be something like the signer's name. It can be any picture instead, for instance a sequence of digits. We coin the term *electronic signature* (e-signature for short) which emphasizes the fact that both the signature and the document it is attached to exist in electronic form only. This term is in the same line as e-mail, e-cash etc. The ancient term *digital* is better be voided nowadays. The term *electronic signature* should not be misused for other primitives that exhibit certain similarities with signatures. The key property of a signature, both hand-written and electronic, is that the signer intentionally attached it to a given document to authorize it. This is not the case, e. g. with fingerprints.

Stern [10]). Moreover, random oracle model was successfully applied to cryptographic protocols of other types and nowadays a substantial amount of security proofs is given in this model.

However the random oracle model remains to be an idealization only. Attempts to instantiate random oracle with cryptographic primitives resulted in no success. Moreover, Canetti et al. [4] constructed an e-signature scheme secure in the random oracle model, but insecure when random oracle is instantiated with any efficient function.

In the generic group model the group operation is available to adversary only through calls to oracle. This means that adversary can run only generic algorithms. Brown [3] proved security of ECDSA in the generic group model. This model is also an idealizations only. Dent [5] constructed cryptographic schemes secure in the generic group model but insecure when instantiated with any efficiently realizable group.

Another line of research was initiated by Varnovsky [1] who studied tamper-proof device model. Instead of access to random oracle each participant is provided with a tamper-proof device implementing a private-key cryptosystem. In this model was demonstrated security of a somewhat modified former GOST[2] e-signature scheme [2]. In that variant of GOST they used a group of residues modulo a prime. The security guarantee was based on three conjectures: computational intractability of discrete logarithm problem, collision resistance of a hash function, and security of private-key cryptosystem. The last conjecture was rather unusual.

In the present paper we study the current version of GOST e-signature scheme with the following modifications:

- parameters of the scheme depend on a growing security parameter, i. e. we consider as is usual in mathematical cryptography an infinite family of schemes;
- hash value $H(m)$ of a message m to be signed is submitted to tamper-proof device for encryption. In the signature generation algorithm one uses encrypted hash value $E_K(H(m))$, where K is a private key and E is the encryption function.

For modified this way GOST e-signature scheme we prove existential unforgeability with respect to chosen-message attack. The security guarantee is based on the following assumptions:

- physical assumption on tamper-proof device. Private key K is the same for all the tamper-proof devices and supposed to be physically shielded;
- IND–CPA security of the private-key cryptosystem (indistinguishabilty under chosen-plaintext attack);

[2] Not to be confused with ghost! In Russian GOST is just an acronym for National Standard.

- collision resistance of a hash function;
- intractability of discrete semi-logarithm problem.

Although the resulting e-signature scheme is not practical its security guarantee does not use any idealized models. All the four assumptions are standard for cryptography. For instance, such companies as General Electric and IBM use in their bank systems tamper-proof devices (see e. g. [12]). IND–CPA security is achievable even for probabilistic public-key cryptosystems [7]. The last two assumptions are also necessary conditions for security of e-signature schemes.

Moreover, it is well known that IND–CPA secure private-key cryptosystems exist iff there exist one-way functions. Since one-way functions are necessary for the existence of secure e-signature schemes the results of the present paper could be interpreted as follows. In our setting we prove modulo the physical assumption a necessary and sufficient condition for security of the e-signature scheme.

Fiat–Shamir paradigm allows one to prove security of e-signature schemes under assumption that hash function is "as secure as" a random function. We take a moderate step towards practice. In our paradigm an e-signature scheme could be shown secure if a hash function is "as secure as" an encryption function of an IND–CPA secure private-key cryptosystem.

In section 2 we specify a modification of the GOST e-signature scheme to be considered. Sections 3 to 5 are devoted to conjectures used to study its security. In section 6 we state a definition of security of an e-signature scheme and prove the main result.

2 e-signature scheme

Let n be a security parameter which is defined to be the binary length of a private key.

For each n there exists a finite set of instantiations of an e-signature scheme. Each of these instantiations is defined by a certain group $\langle g \rangle$ and a set X. We use multiplicative notation for the group $\langle g \rangle$.

A private key x is drawn from uniform probability distribution over the set X. A public key is $y = g^x$.

In the GOST e-signature scheme $\langle g \rangle$ is an order q cyclic subgroup of an elliptic curve group, X being the set $\{1, \ldots, q-1\}$. Thus, the security parameter is the binary length of a prime q.

The e-signature scheme makes use of two additional cryptographic primitives, hash function H and encryption function $E_K(\cdot)$ of a private-key cryptosystem. Private key K of this cryptosystem is stored in tamper-proof devices and is anavailable to all participants of the scheme.

To define an e-signature scheme it suffices to specify signature verification algorithm. In the modified GOST scheme at hand this algorithm is as follows.

Let (r, s) be a purported signature for a message m.

1. Compute $H(m)$.

2. Hash value $H(m)$ is submitted to the tamper-proof device which returns an encrypted hash value $h = E_K(H(m))$.

3. Encrypted hash value h and signature (r, s) are substituted into the signature verification relation $r = g^{sh^{-1}} y^{-rh^{-1}}$.

4. If the equality holds the signature is accepted (valid signature), otherwise it is rejected.

From now on the equality $V(r, s, m) = 1$ means that a pair (r, s) is a valid signature for a message m.

To sign a message m the owner of a private key has to compute encrypted hash value $h = E_K(H(m))$, choose session private key k uniformly at random in the set X, compute session public key $r = g^k$ and finally compute $s = rx + kh$.

This description uses certain simplifying conventions. In the GOST e-signature scheme session public key r is an elliptic curve point. To use this value in operations over the set X one needs a transformation mapping $f : \langle g \rangle \to X$. In the GOST scheme the transformation mapping f picks the first coordinate of the point and reduces it modulo q. To simplify notation we omit in what follows any references to the transformation mapping f and all reductions modulo q.

3 Discrete semi-logarithm problem

Definition 1. *Discrete semi-logarithm of an element $z \in \langle g \rangle$ is any pair (t, u) such that $t = g^u z^{-t}$.*

Thus if one considers z as a public key of an e-signature scheme then its discrete semi-logarithm is a signature for a message m such that $H(m) = 1$.

It is evident that the problem of finding discrete semi-logarithms is not harder than the discrete logarithm problem.

From now on PPT is a shorthand for probabilistic polynomial Turing machine.

Conjecture 1. *Let z be a random element of the group $\langle g \rangle$. Then for any polynomial p, for any PPT A*

$$\Pr\{A(g, z) = (t, u) : t = g^u z^{-t}\} < 1/p(n)$$

for all sufficiently large n.

The probability is over the random choice of z, random choices of algorithm A and, in general, random choice of a group $\langle g \rangle$ from the set of all groups corresponding to a given security parameter n.

The next lemma shows that Conjecture 1 provides a necessary condition for security of e-signature scheme.

Lemma 1. *Suppose Conjecture 1 does not hold, i. e. there exist a polynomial p and a PPT A such that*

$$\Pr\{A(g, z) = (t, u) : t = g^u z^{-t}\} \geq 1/p(n)$$

for infinitely many n. Then there exists a PPT B such that

$$\Pr\{B(g,y,m)=(r,s):V(r,s,m)=1\}\geq 1/p(n)$$

for infinitely many n.

Proof. On input (g,y,m) the machine B computes $h=E_K(H(m))$ and $z=y^{h^{-1}}$. Then it calls A as a subroutine feeding it with input (g,z). Since y is a random element of the group $\langle g\rangle$, z is a random element of this group as well. Therefore the pair (g,z) generated by B has the same probability distribution as the pair of input values of A in the supposition of the lemma. By this supposition A finds for infinitely many n a discrete semi-logarithm (t,u) of z with probability at least $1/p(n)$.

Let (t,u) be a pair returned by A. The PPT B verifies whether the equality $t=g^u z^{-t}$ holds and, if so, outputs (t,uh) and halts.

It is clear that B proceeds in polynomial time.

If (t,u) is a discrete semi-logarithm of z then

$$g^{(uh)h^{-1}}y^{-th^{-1}}=g^u z^{-t}=t,$$

i. e. the pair (t,uh) is a valid signature for the message m. □

4 Hash function

Hash function H can be defined as a family of hash functions $\{H_n\}$ where the function H_n maps messages of arbitrary length into the set $\{1,\ldots,2^n-1\}$. An index n is always clear from the context and therefore omitted for simplicity.

The required cryptographic properties of a hash function are stated in the next conjecture.

Conjecture 2. *For any PPT A, for any polynomial p and all sufficiently large n*

$$\Pr\{A(1^n)=(m,m'):m\neq m'\ \ \&\ \ H(m)=H(m')\}<1/p(n).$$

This requirement is standard for cryptographic hash functions. In mathematical cryptography it is formalized by the notion of a family of collision-intractable hash functions. However this conjecture is the most problematic one. Collision intractability seems to be too much to require from individual function. For instance, if one turns to non-uniform computation model than the set of hash functions satisfying an analogous conjecture is evidently empty

The next lemma shows that for e-signature schemes based on individual hash-functions Conjecture 2 provides a necessary condition for security. For definition of existential unforgeability with respect to chosen-message attack the reader is referred to section 6.

Lemma 2. *Suppose the Conjecture 2 does not hold, i. e. there exist a PPT A and a polynomial p such that*

$$\Pr\{A(1^n)=(m,m'),m\neq m'\ \ \&\ \ H(m)=H(m')\}\geq 1/p(n)$$

for infinitely many n. Then there exists a PPT B such that

$$\Pr\{B(g,y) = (m,r,s) : V(r,s,m) = 1\} \geq 1/p(n)$$

for infinitely many n.

The PPT B can mount a chosen-message attack on the e-signature scheme.

Proof. Machine B can call A as a subroutine, get from it a collision (m, m') and then obtain a signature (r,s) for the message m' using a chosen-message attack. It is clear that the pair (r,s) is also a valid signature for the message m. □

5 Tamper-proof device

In the setting being considered each participant of e-signature protocol has in his possession a tamper-proof device implementing an encryption function $E_K(\cdot)$ of a private-key cryptosystem. The private key K is chosen at random by the key generation algorithm and is the same for all tamper-proof devices.

Since an adversary is assumed to have a tamper-proof device in his possession the contents of this device should be shielded. More precisely, an adversary is supposed to be ignorant of the value of the private key K. In general there are many ways to formalize this requirement. To obtain a security guarantee for the e-signature scheme in question it suffices to require that tamper-proof device shielding provides for the IND-CPA security of the cryptosystem.

In the case an adversary manages to succeed in reverse-engineering and obtains the private key, this would not lead to any fatal consequences. It is not clear whether knowledge of the private key facilitates signature forgering. Intuitively, it seems that signature forgering with known private key K is no easier than the same task for original (without tamper-proof devices) e-signature scheme. However we were unable to justify this intuition and pose this as an open problem.

An adversary having access to a tamper-proof device is able to mount a chosen-plaintext attack on the cryptosystem. This means that an adversary can choose plaintexts $m_1, \ldots, m_t$ and obtain the corresponding ciphertexts $c_1, \ldots, c_t$, where $c_i = E_K(m_i)$, $i = 1, \ldots, t$. An attack may be adaptive, i. e. when choosing a current plaintext m_i an adversary knows ciphertexts $c_1, \ldots, c_{i-1}$.

We consider ciphertext distinguishability threat: an adversary chooses two plaintexts m^0, m^1 and gets a ciphertext of one of them chosen at random. The threat is that an adversary can distinguish ciphertexts of plaintexts m^0, m^1 of her choice. The following scenario is allowed: after choosing plaintexts m^0, m^1 and obtaining a ciphertext c an adversary proceeds with the chosen-plaintext attack. But in any case it is required that $m^0, m^1 \neq m_i$ for all $i = 1, \ldots, t$.

Formally, an adversary is an oracle PPT A^E. An input word to this machine is the security parameter n in unary. The oracle E chooses a private key K using the cryptosystem key generation algorithm. PPT A can submit to the oracle E two kinds of queries:

- regular queries of the form $(1, m_i)$. The oracle answers to this query with ciphertext $c_i = E_K(m_i)$;
- special query of the form $(2, m^0, m^1)$. The oracle chooses a random bit σ and returns a ciphertext $c = E_K(m^\sigma)$.

Only one special query is allowed and this can be issued at any time moment of adversary's choice. It is required that $m_i \neq m^0$ and $m_i \neq m^1$ for any i.

By $A^E(1^n) = \sigma$ we denote the following event: PPT A after getting a ciphertext c in return to its special query outputs a bit b such that $b = \sigma$ and halts.

Definition 2. *A cryptosystem is IND–CPA secure if for any oracle PPT A^E, for any polynomial p and any sufficiently large n*

$$|\Pr\{A^E(1^n) = \sigma\} - 1/2| < 1/p(n).$$

The probability is over random choices of the algorithm A, and random choices of the private key K and bit σ.

Conjecture 3. *The cryptosystem implemented in tamper-proof devices is IND–CPA secure.*

6 Security of the e-signature scheme

We consider security of e-signature scheme against existential forgery based on (adaptive) chosen-message attack. For classification of attacks and threats that can be defined for e-signature schemes the reader is referred to [8].

Formally a chosen-message attack is modelled by allowing an adversary to access an oracle S. A query is defined to be a message m and the oracle responds with a pair (r, s) such that $V(r, s, m) = 1$.

An adversary is defined as an oracle PPT A^S. Input to this machine is a pair (g, y). The PPT A, also has access to the oracle E for the encryption function implemented in a tamper-proof device.

To simplify notation for machines with two oracles we sometimes omit one of them in superscripts.

Let $m_1, \ldots, m_t$ be the set of all messages submitted by A as queries to the oracle S. Parameter $t = t(n)$ is a function upper bounded by a polynomial due to the time complexity of A.

Definition 3. *e-signature scheme is existentially unforgeable with respect to the chosen message attack if for any oracle PPT $A^{S,E}$ for any polynomial p and all sufficiently large n*

$$\Pr\{A^{S,E}(g, y) = (m, r, s) : m \neq m_i, i = 1, \ldots, t \ \ \& \ \ V(r, s, m) = 1\} < 1/p(n).$$

The probability is defined dy random choices of parameters and keys of e-signature scheme, random choice of cryptosystem private key and random choices of algorithms A and S.

The next theorem addresses the e-signature scheme defined in section 2 and assumes that Conjectures 1–3 hold.

Theorem 1. *The e-signature scheme is existentially unforgeable with respect to the chosen-message attack.*

Proof. Suppose to the contrary that there exist an oracle PPT $A^{S,E}$ and a polynomial p such that

$$\Pr\{A^{S,E}(g,y) = (m,r,s) : m \neq m_i, i = 1,\dots,t \ \ \& \ \ V(r,s,m) = 1\} \geq 1/p(n)$$

for infinitely many n.

Let $\varepsilon = \varepsilon(n) = 1/p(n)$. From now on in probability estimates we omit additive negligible terms fot simplicity.

Suppose that $h_1,\dots,h_t$ is the set of all hash values signed during the chosen-message attack, i. e. $h_i = E_K(H(m_i))$, $i = 1,\dots,t$.

It is clear that at least one of the following cases occurs with probability at least $\varepsilon/2$ infinitely often:

- there exists $i \in \{1,\dots,t\}$ such that $h_i = h$, where $h = E_K(H(m))$;
- $h_i \neq h$ for all $i = 1,\dots,t$.

We handle these two cases separately.

1. For any given key K the function E_K is one-to-one, therefore the equality $h_i = h$ implies that $H(m_i) = H(m)$. Thus A can be used to find collisions of the hash-function H. To this end we construct a PPT B (hash-function adversary), which calls A as an oracle.

The only minor technical problem is as follows. Machine A has itself access to two oracles S and E, therefore B should be able to intercept and process all the queries to these oracles.

Given an input 1^n B generates public parameters and keys of the e-signature scheme according to algorithms of this scheme, and generates private key K using the key generation algorithm of the cryptosystem implemented in the tamper-proof device.

Then B calls A feeding it with input (g,y). It is evident that B is able to answer all the queries of A to oracles. Moreover, all the random variables will have the same probability distributions as in the above supposition. Hence a collision will be found with probability at least $\varepsilon/2$.

2. The case when h is a collision with none of the h_i is further divided into two subcases:

- the PPT A infinitely often with probability at least $\varepsilon/4$ forges a signature for a message m without querying the tamper-proof device on the hash value $H(m)$;

- the PPT A infinitely often with probability at least $\varepsilon/4$ forges a signature for a message m for which the encrypted hash value $h = E_K(H(m))$ was obtained as a result of a call to the tamper-proof device.

2.1. The PPT A without querying the tamper-proof device on the hash value $H(m)$ forged a signature for a message m, i. e. generated a pair (r, s) such that $V(r, s, m) = 1$. This can be used to construct an algorithm contradicting Conjecture 3 on the IND–CPA security of the cryptosystem.

Define a PPT B_1 (an adversary for the cryptosystem) as follows. Given an input 1^n machine B_1 generates public parameters and keys of the e-signature scheme according to algorithms of this scheme. Then B_1 calls A on input (g, y). The PPT A has access to two oracles, S and E. Queries to these oracles are intercepted by B_1. All the queries to the oracle E are answered using chosen-plaintext attack, while all the queries to the oracle S can be answered due to the knowledge of a private key of the e-signature scheme.

When A outputs a message m and a corresponding signature (r, s) the PPT B_1 produces a query $(2, H(m^0), H(m^1))$, where $m^0 = m$, m^1 is a random message and submits this query to the oracle. After receiving the oracle answer $h = E_K(H(m^\sigma))$, where σ is a random bit, B_1 runs the signature verification algorithm using this value h to verify the signature (r, s). If the verification passes B outputs 0, otherwise it outputs 1 and in either case halts.

It is clear that

$$\Pr\{B_1^E(1^n) = \sigma\} - 1/2 \geq \varepsilon/4.$$

2.2. Main case. The PPT A forges a signature (r, s) for a message m such that the tamper-proof device was queried on the hash value $H(m)$. Now we define a PPT B_2 (algorithm for the discrete semi-logarithm problem) whose existence would contradict Conjecture 1. The key idea is to substitute $h = E_K(H(m))$ obtained by A as a result of a query by $h' = E_K(H(m'))$ where m' is a random message.

If A generates a valid signature for h' with probability less than $\varepsilon/4$ this can be used to reach a contradiction with Conjecture 3.

In the opposite case the PPT B_2 will be successful with nonnegligible probability.

The PPT B_2 on input (g, z) generates private key K of the cryptosystem using the key generation algorithm. Then B_2 chooses random $\alpha \in \{0, 1\}^n$ and computes $e = E_K(\alpha)$. Next it computes $y = z^e$ and calls A feeding it with input (g, y).

Machine A is provided access to two oracles, E and S, therefore B_2 should process queries to both. Queries to oracle E (chosen plaintext attack on the cryptosystem) are responded easily since B_2 knows cryptosystem private key. Queries to oracle S (chosen message attack on the e-signature scheme) require ability to generate valid signatures. Note that B_2 does not know private key corresponding to the public key y, i. e. its discrete logarithm.

To forge signatures B_2 uses the following trick. It chooses $\lambda, \mu \in_R X$ and computes $r = g^\lambda y^\mu$. Next it puts $h = -r\mu^{-1}$ and takes h as a substitute for encrypted hash value $E_K(H(m))$ of a given message m. Then B_2 computes $s = \lambda h$. It is evident that (r, s) is a valid signature for a message m' such that $h = E_K(H(m'))$. Indeed,

$$g^{sh^{-1}} y^{-rh^{-1}} = g^{-\lambda r\mu^{-1}(-r\mu^{-1})^{-1}} y^{r(r\mu^{-1})^{-1}} = g^{\lambda} y^{\mu} = r.$$

Next we describe how B_2 processes queries of A to oracles S and E. To distinguish messages from plaintext the latters from now on will be denoted by Greek letters.

The PPT B_2 maintains a list of replies to queries. Initially this list is empty.

Given a message m (a query to the oracle S) or plaintext α (a query to the oracle E) B_2 checks whether the list of replies contains an entry with such a value α where in the first case $\alpha = H(m)$. If so, outputs either a pair (r, s) (query to the oracle S) or h (query to the oracle E).

Otherwise B_2 chooses random λ, μ, computes r, s and h as above and fixes h as a substitute for $E_K(H(m))$. Either a pair (r, s) (query to the oracle S) or a value h is returned to A. Then B_2 adds to its list of replies new entry (α, h, r, s), where in the case of a query to the oracle S, $\alpha = H(m)$.

Thus B_2 responds to queries to the oracle S by substituting true encrypted hash value $E_K(H(m))$ by a randomly generated value h. Now we show that on these "false" values h A must exhibit virtually the same behavior as on true ones.

Let $p_1 = p_1(n)$ be the probability that A forges a signature for whatever message when having access to oracles E and S. We have supposed that $p_1 \geq \varepsilon/4$. Let $p_2 = p_2(n)$ be the probability of the same event in the case when oracles E and S are emulated by B_2 as above.

Suppose that $p_1 - p_2 > \varepsilon/8$. To show that this contradicts Conjecture 3 we use well-known hybrid argument. Let $t = t(n)$ be the total number of queries to oracles E and S issued by A. For each $i = 0, 1, \ldots, t$ consider the i-th hybrid G_i. By definition, G_i is the string of replies to queries of A to oracles in the case when all the queries up to the i-th one are emulated by B_2 while the queries $i + 1, \ldots, t$ are responded by oracles E and S.

Let p^i be the probability of successful forgering when A's queries to oracles are responded with hybrid G_i. It is clear that $p_1 = p^t$, $p_2 = p^0$. Therefore there are two adjacent hybrids G_i and G_{i+1} such that $p^{i+1} - p^i \geq \varepsilon/8t$.

Now we construct a PPT B_3 (an adversary for the cryptosystem) which calls A as a subroutine. The PPT B_3 is provided access to the oracle E. On input 1^n it generates keys and public parameters of e-signature scheme. Then B_3 chooses a random number i in the set $\{1, \ldots, t-1\}$ and responds to queries $1, \ldots, i$ of A using access to the oracle E or keys of e-signature scheme as appropriate. Starting with the $(i + 1)$-th query B_3 proceeds as described above for B_2 but for the following modification.

Processing a current query of A to oracle B_3 chooses $\alpha \in_R \{0, 1\}^n$, computes $h = E_K(\alpha)$ and checks whether $h \in X$. If so B_3 chooses $k \in_R X$, computes $r = g^k$ and finds μ and λ from relations $h = -r\mu^{-1}$ and $\lambda + x\mu = k$. It is easy to see that parameters λ, μ and the private key x have the same probability distributions for PPT's B_2 and B_3. Indeed, each of these parameters is chosen uniformly at random and independently in the set X. Probability that $E_K(\alpha) \in X$ for $\alpha \in_R \{0,1\}^n$ is nonnegligible (for instance, it is $\geq 1/2$ for the GOST scheme). Using well-known probability amplification techniques one can ensure that after polynomially many attempts the probability of not hitting the set X is exponentially vanishing. Therefore the probability distributions of triples (r, s, h) generated by PPT's B_2 and B_3 are statistically close.

To get a contradiction with Conjecture 3 take as plaintexts α^0 and α^1 for a special query the plaintext of i-th query of A and the value of α generated by B_3 when processing the i-th query. Finally, B_3 outputs 1 if A forged a signature for whatever message and outputs 0 otherwise. It is easy to see that B_3 distinguishes ciphertexts of plaintexts α^0 and α^1 with nonnegligible probability.

Now let $p_2 \geq \varepsilon/8$. Then one can use B_2 to find discrete semi-logarithms. On input (g, z) the PPT B_2 forged a signature (r, s) for a random message m. Machine PPT B_2 is modified to pick a random $j \in \{1, \ldots, t\}$ and to respond the j-th query to oracle E with the value of e. Recall that $y = z^e$. Then the signature will be forged just for this encrypted hash value with probability at least $\varepsilon/8t$.

Let (r, s) be a signature for the encrypted hash value e. Then (r, se^{-1}) is the discrete semi-logarithm for (g, z). Indeed,

$$g^{se^{-1}} y^{-re^{-1}} = g^{se^{-1}} z^{-r}. \quad \square$$

References

[1] Varnovsky N., Security of e-signature schemes in tamper-proof device model, Manuscript, 1996 (in Russian)

[2] Varnovsky N., Security of e-signature schemes in tamper-proof device model, Technical Report, Russian Academy of Cryptography, 1998 (in Russian)

[3] Brown D., On the provable security of ECDSA, Designs, Codes and Cryptography, **v. 35**, N 1, 2005, 119–152

[4] Canetti R., Goldreich O., Halevi S., The random oracle methodology revisited, Proc. 30th Annu. Symp. on Theory of Computing, 1998, 209–218

[5] Dent A., Adapting the weaknesses of the random oracle model to the generic group model, Proc. ASIACRYPT 2002, Lect. Notes in Comput. Sci., **v. 2501**, 2002, 95–104

[6] Fiat A., Shamir A., How to prove yourself: practical solutions to identification and signature problems, Proc. CRYPTO'86, Lect. Notes in Comput. Sci., **v. 263**, 1987, 186–194

[7] Goldwasser S., Micali S., Probabilistic encryption, J. of Computer and System Sciences, **v. 28**, N 2, 1984, 270–299

[8] Goldwasser S., Micali S., Rivest R., A secure digital signature scheme, SIAM J. on Computing, **v. 17**, N 2, 1988, 281–308

[9] Naor M., Yung M., Universal one-way hash functions and their cryptographic applications, Proc. 21st Annu. Symp. on Theory of Computing, 1989, 33–43

[10] Pointcheval D., Stern J., Security proofs for signature schemes, Proc. EUROCRYPT'96, Lect. Notes in Comput. Sci., v. 1070, 1996, 387–398

[11] Rompel J., One-way functions are necessary and sufficient for secure signatures, Proc 22nd Annu. Symp. on Theory of Computing, 1990, 387–394

[12] Shain M., Security in electronic funds transfer: message integrity in money transfer and bond settlements through GE information services' global network, Computers and Security, **v. 8**, 1989, 209–221