One-Time HNP or Attacks on a Flawed El Gamal Revisited

Tomáš Rosa

trosa@ebanka.cz

eBanka, a.s.
Václavské náměstí 43
110 00 Prague 1, Czech Republic, EU

Abstract

We present a modification of the well-known *hidden number problem* (HNP) which we refer to as a *one-time* HNP (OT-HNP). We also present an algorithm for solving such a problem together with its formal analysis. We show then that carefully designed instances of OT-HNP can be used to break certain flawed implementations of public key schemes efficiently. We work, for instance, with Nguyen's attack on El Gamal's signature scheme in the GNU Privacy Guard of version 1.2.3. The technique employed there was not based on HNP, since it was supposed that more than one signature would be necessary, which seemed to be a wastage. We will see, however, that by using OT-HNP one signature is still far enough, while retaining certain elegance of the HNP approach. We also present an experimental confirmation of this result.

Keywords: cryptanalysis, public key cryptography, El Gamal, DSA, hidden number problem, lattice, implementation, side channels.

1 Introduction

It is a well-known fact that cryptanalytical techniques can only get better. And as we can see, they already do. Sometimes faster, sometimes only in, let us say, formal improvements. One of those areas still deserving a close attention are cryptanalytical methods aimed at breaking El Gamal's [5] and the other DSA-like signature schemes [10] through their extraordinary sensitivity to a partial information leakage of secret temporary nonces used during a signing operation. An attacker can gain this information in either a-priori way due to their disturbed probabilistic distribution or in a-posteriori way due to their leakage through various side channels. If we ignore the discrete-log part of these schemes, we can say that cryptanalytical techniques (cf. mainly [13], [8], [3]) exist allowing an attacker to practically break the scheme almost as soon as the corresponding mathematical problem becomes solvable from an information-theoretic point of view. Looking on it from security architects perspective, such a property shall be alarming.

There seem to be basically two elementary kinds of algorithms used: The lattice-based ones [13], [8], and special searching techniques [3], [2]. Special techniques currently hold the public record in how small amount of information is enough – it is 1 bit of each DSA temporary nonce [2]. However, their drawback is the amount of signatures needed – it is a number of order 2^{24}. This far more than what is needed for lattice-based methods which, on the other hand, are consuming more information – they seem to need at least 3 bits of each nonce [13]. The main cryptanalytical advances we can expect here are lowering the amount of information or the number of signatures needed. This seems to be a difficult task, since, for instance, the minimum information needed for the lattice-based methods is mainly given by the fact that we can only get some approximate solutions to the lattice problems employed. On the other hand, some bright new ideas on how to arrange the corresponding cryptanalytical problems can make many things possible.

In the following text, we expect that the reader is familiar with the elementary geometric number theory [9] together with basic properties of lattices and their applications in cryptanalysis [14]. Very interesting lattice-based methods that seem to be perfectly adjusted for solving the cryptanalytical problems arising with flawed implementations of DSA-like signature schemes are connected with the *hidden number problem* (HNP). Roughly speaking, the task is to find a secret integer x satisfying certain approximations in a form $(t_i x - u_i) \bmod N < \delta_i$. For the first time, it was formulated and used in 1996 by Boneh and Venkatesan [4] to prove security of the most significant bits for Diffie-Hellman key agreement protocol. In 1999, Nguyen pointed out [11] a connection between solving HNP and breaking flawed implementations of DSA [6]. The work was then extended together with Shparlinski in [13]. Our article contributes to this area by introducing a modification of HNP called one-time HNP (OT-HNP) that allows us to utilise a direct information on x itself, which was not covered by HNP. OT-HNP can be regarded as a tool for finding a small integral solution x of a single modular inequality in the form $(tx - u) \bmod N < \delta$, where $0 < x < X$. Note that HNP assumes only that $0 < x < N$, which leads to an unnecessary loss of information when $X < N$. Allowing x itself to be bounded by a certain estimate X, $X < N$, together with using only one modular inequality are the main ideas behind the transition from HNP to OT-HNP. Roughly saying, provided that $\delta X < N^{1-\varepsilon}$ (cf. Theorem 1 for precise statement), we can expect such a solution to be probably unique and findable by an algorithm presented here (cf. experimental results presented in §3.3). We then show how OT-HNP connects with the cryptanalytical problem discussed in [12].

A formal description of OT-HNP together with an algorithm to solve it and its basic analytical properties are presented in §2. In §3, we show how to use OT-HNP as an alternative method to break flawed implementation of El Gamal's signature scheme investigated in [12]. Finally, we conclude in §4.

2 One-Time HNP

Definition 1 One-time hidden number problem (OT-HNP). *Let x be a particular secret integer satisfying $0 < x < X$, where X, $X \in \mathbf{Q}$, is known. Furthermore, let us be given a quadruplet (t, u, δ, N), where $t, u \in \mathbf{Z}$, $N \in \mathbf{N}$, $X < N$, and $\delta \in \mathbf{Q}$, satisfying $(tx - u) \bmod N < \delta$, $t \bmod N \neq 0$. The one-time hidden number problem is then to find x and its particular instance is specified as the quintuplet (t, u, δ, N, X).*

There is a straightforward generalization of OT-HNP to its multidimensional variant where we are given a set of d inequalities, each determined by a particular quadruplet $(t_i, u_i, \delta_i, N_i)$, $1 \leq i \leq d$. The following algorithm together with its justification can be then extended for such a generalization as well. For the sake of simplicity and clear connection with solving the cryptanalytical problem in §3, we will, however, do without this generalization here. Even the words "one-time" in the name of the problem express our motivation to develop an HNP-rooted method for solving the problems like the one stated in §3 having only one signature, i.e. only one independent inequality for x.

Definition 2 $L(t, N, \gamma)$. *Let $t \in \mathbf{Z}$, $N \in \mathbf{N}$, and $\delta \in \mathbf{Q}$. By $L(t, N, \gamma)$ we mean a two-dimensional full-rank lattice spanned by the base vectors $\boldsymbol{b}_1 = (N, 0)$ and $\boldsymbol{b}_2 = (t, \gamma)$, i.e. $L(t, N, \gamma) = \{z_1 * \boldsymbol{b}_1 + z_2 * \boldsymbol{b}_2 : (z_1, z_2) \in \mathbf{Z}^2\}$.*

Algorithm 1 Solving OT-HNP.

Input: Instance of OT-HNP specified by (t, u, δ, N, X).

Output: Solution candidate x'.

Computation:

```
1. Let $\gamma = \delta/X$.
2. Set up a rational vector $\mathbf{v} = (v_1, v_2)$, where $v_1 = u + \delta/2$, $v_2 = \delta/2$.
3. Compute $\mathbf{w} \in L(t, N, \gamma)$, $\mathbf{w} = (w_1, w_2)$, which is the (approximately) closest vector of $L(t, N, \gamma)$ to $\mathbf{v}$.
4. Let $x' = (w_2/\gamma) \bmod N$, where we apply a standard division on $\mathbf{Q}$ and expect the result to be in $\mathbf{Z}$.
5. Return $x'$.
```

■

As was stated in the description of Algorithm 1, its result is to be regarded as a solution *candidate* of the given instance of OT-HNP. The following statements elaborate its correctness formally. Their purpose is mainly to illustrate the reasoning behind the composition of Algorithm 1 together with showing the connection between HNP and OT-HNP. Therefore, our elaboration corresponds closely with the description given by Nguyen and Shparlinski in [13] and somehow also with the first study of HNP by Boneh and Venkatesan in [4]. Note that our algorithm relies heavily on an access to a method for solving the approximate closest vector problem on a certain lattice (cf. [14] for an overview). One can use, for instance, the well-known Babai's nearest plane algorithm [1]. Our expectation on the particular approximation factor of such an algorithm is stated by the following proposition. Note that the overall time complexity of Algorithm 1 depends mainly on the algorithm used for the closest vector approximation and can be essentially regarded as being polynomial.

Proposition 1 Approximate solution of the closest vector problem [1]. *Let L be a lattice of dimension d and let us be given a vector $\mathbf{v} \in \mathbf{R}^d$. We assume there exists a polynomial time algorithm which finds a lattice vector $\mathbf{w} \in L$, such that $\| \mathbf{v} - \mathbf{w} \| \leq 2^{d/4} \min_{\mathbf{b} \in L} \| \mathbf{v} - \mathbf{b} \|$.*

In the following elaboration, we will work with an assumption that the modulus N is a prime. This is because we want to stay focused on showing clearly the main ideas behind OT-HNP, while the formal elaboration when N is a composite would require deeper justifications of the probability distributions used in Lemma 1 together with a slightly different approach to a uniqueness of the solution in Theorem 1. On the other hand, we can expect that very similar results will also hold for composite moduli, as well. Experiments done in §3 fully support this hypothesis.

Lemma 1 On short vectors in $L(t, N, \gamma)$. *Let N be a prime, t integer uniformly distributed on* $\langle 1, N)$, *and* $\gamma, \delta \in \mathbf{Q}$, *such that* $0 < \delta < N/3$, $0 < \gamma$, *and* $9\delta^2\gamma^{-1} < (N - 1)$. *Then with a nonzero probability* $P \geq 1 - 9\delta^2(\gamma(N - 1))^{-1}$ *all* $\boldsymbol{\Delta} \in L(t, N, \gamma)$ *satisfying* $\| \boldsymbol{\Delta} \|_\infty \leq 3\delta/2$ *are in the form* $\boldsymbol{\Delta} = (0, zN\gamma)$, *where* $z \in \mathbf{Z}$.

Proof. Let us recall that by Definition 2 it holds $\boldsymbol{\Delta} = (\alpha N + \beta t, \beta\gamma)$, where $(\alpha, \beta) \in \mathbf{Z}^2$.

Next, we observe that $\beta = zN$ implies $\alpha N + \beta t = 0$, since it must hold $| \alpha N + \beta t | = | N(\alpha + zt) | \leq 3\delta/2 < N/2$. Therefore, we can focus only on elaborating the probability P of the event $\beta = zN$. We have $P = 1 - \Pr[\beta \bmod N \neq 0]$. If $\beta \bmod N \neq 0$ then the following inequalities must hold:

$$0 < | \alpha N + \beta t | \leq 3\delta/2 < N/2,$$

$$0 < | \beta\gamma | \leq 3\delta/2, \text{ i.e. } \beta \in \boldsymbol{I} = \langle -3\gamma^{-1}\delta/2, 0) \cup (0, 3\gamma^{-1}\delta/2\rangle.$$

We select an integer β from the interval $\boldsymbol{I}$ and denote $p(\beta)$ the probability that the first inequality is satisfied. If there is no such integer β on $\boldsymbol{I}$, we set $p(\beta) = 0$. Otherwise, let us observe that the first inequality implies $(\beta t \bmod N) \in (0, 3\delta/2\rangle \cup \langle N - 3\delta/2, N)$, since βt must be within a distance of up to $3\delta/2$ from an integral multiple of N. Furthermore, we assume t to be uniformly distributed on $\langle 1, N)$, which implies the uniform distribution of $(\beta t \bmod N)$ on the same interval, since $\gcd(\beta, N) = 1$. From here, we can write $p(\beta) \leq 3\delta/(N - 1)$. Finally, we do a rough but sufficient estimation $\Pr[\beta \bmod N \neq 0] \leq \sum_{(\beta' \in \boldsymbol{I})} p(\beta') \leq 9\delta^2(\gamma(N - 1))^{-1}$ giving $P \geq 1 - 9\delta^2(\gamma(N - 1))^{-1}$.

■

Theorem 1 On the solution candidates returned by Algorithm 1. *Let us be given an OT-HNP instance* (t, u, δ, N, X), *where N is a prime, t is uniformly distributed on* $\langle 1, N)$, *and X and* δ *satisfy* $\delta < N/3$ *and* $9\delta X < (N - 1)$. *Then with a nonzero probability* $P \geq 1 - 9\delta X(N - 1)^{-1}$ *Algorithm* 1 *returns the correct unique solution of the OT-HNP problem instance.*

Proof. By the definition of OT-HNP, there must be $c, x \in \mathbf{Z}$ satisfying $0 \leq tx - u + cN < \delta$, so $| tx - u - \delta/2 + cN | \leq \delta/2$. Furthermore, we have $0 < x < X$, so $| \gamma x - \delta/2 | < \delta/2$, where $\gamma = \delta/X$.

Let us set $\boldsymbol{v} = (u + \delta/2, \delta/2)$ and consider the lattice $L(t, N, \gamma)$. Observe that there is a vector $\boldsymbol{h} \in L(t, N, \gamma)$, such that $\boldsymbol{h} = (tx + cN, \gamma x)$ and $\| \boldsymbol{h} - \boldsymbol{v} \|_\infty \leq \delta/2$. In [13], they call $\boldsymbol{h}$ as a *hidden vector*, since it discloses the value of the hidden number x directly. The next step is to use a suitable algorithm for solving the closest vector problem for $L(t, N, \gamma)$ and $\boldsymbol{v}$. Let the result be denoted as $\boldsymbol{w}$. According to Proposition 1, we assume that $\| \boldsymbol{v} - \boldsymbol{w} \| \leq 2^{1/2} \min_{\boldsymbol{b} \in L(t, N, \gamma)} \| \boldsymbol{v} - \boldsymbol{b} \| \leq 2^{1/2} \| \boldsymbol{v} - \boldsymbol{h} \| \leq 2^{1/2} * 2^{1/2} * \| \boldsymbol{v} - \boldsymbol{h} \|_\infty \leq \delta$. Furthermore, there exists $\boldsymbol{\Delta} \in L(t, N, \gamma)$, $\boldsymbol{\Delta} = \boldsymbol{h} - \boldsymbol{w}$. By the triangle inequality, we get $\| \boldsymbol{\Delta} \|_\infty \leq \| \boldsymbol{h} - \boldsymbol{v} \|_\infty + \| \boldsymbol{v} - \boldsymbol{w} \|_\infty \leq \delta/2 + \delta \leq 3\delta/2$. Applying Lemma 1, we get that with a probability $P \geq 1 - 9\delta X(N - 1)^{-1}$ the lattice $L(t, N, \gamma)$ has such a structure that each $\boldsymbol{\Delta}$, $\| \boldsymbol{\Delta} \|_\infty \leq 3\delta/2$, satisfies $\boldsymbol{\Delta} = (0, zN\gamma)$ which implies $\gamma x - w_2 = h_2 - w_2 = \Delta_2 = zN\gamma$. Therefore, with the probability P, we can recover the hidden number x from $x = (w_2/\gamma) \bmod N$. This is done in step 4 of Algorithm 1. Note that according to the definition of $L(t, N, \gamma)$ it must hold $(w_2/\gamma) \in \mathbf{Z}$.

For the purpose of uniqueness proving, let us recall that we are assuming the event that each $\boldsymbol{\Delta} \in L(t, N, \gamma)$, such that $\| \boldsymbol{\Delta} \|_\infty \leq 3\delta/2$, is in the form $\boldsymbol{\Delta} = (0, zN\gamma)$. Now, let us consider that there is another solution described by c', x' together with its corresponding hidden vector $\boldsymbol{h}'$. Using the definition of OT-HNP, it is easy to see that then $\| \boldsymbol{h} - \boldsymbol{h}' \|_\infty < \delta$ which under the event (structure of $L(t, N, \gamma)$) assumed implies $\boldsymbol{h} - \boldsymbol{h}' = (0, zN\gamma)$, so we get $\gamma x - \gamma x' = zN\gamma$ further implying $x \equiv x' \pmod{N}$. By the definition of OT-HNP we have $x, x' \in (0, N)$, so it holds directly that $x = x'$.

■

3 GPG-Flawed El Gamal Revisited

3.1 Motivation

Let us briefly recall basic properties of El Gamal's signature scheme [5], [10]. Its public parameters consist of a prime p and an integer g which is a generator of $\boldsymbol{Z}_p^*$. The private key is an integer x, $0 < x < p - 1$, and the public key is computed as $y = g^x \bmod p$. To sign a message m, we compute its hash code and format it as an integer H satisfying $0 < H < p - 1$. Specifically, in the implementation attacked in [12], the formatting rules from PKCS#1 v1.5 [15] were used, but it is unimportant for us here. It suffices to note that there is no secret input during the computation of H, so anybody who knows m is able to compute H trivially. Next, we generate a secret random number k, $0 < k < p - 1$, such that $\gcd(k, p - 1) = 1$. Since we need a freshly generated k for each signature, we usually call it a nonce (as a number-used-once). The signature itself is then a pair of integers (r, s), such that:

$$r = g^k \bmod p,$$

$$s = (H - xr)k^{-1} \bmod (p - 1), \text{ where } kk^{-1} \equiv 1 \pmod{p - 1}.$$

The main issue of El Gamal's signature scheme implementation in GNU Privacy Guard (also denoted as GPG) of versions 1.0.2 – 1.2.3 was the following: Aiming for speeding up certain private key operations, architects decided to lower the sizes of nonces k and the private key x significantly. Specifically, their maximum bitlengths were both restricted to $3q_{\text{bit}}/2$, where q_{bit} was a function of the length of prime p and it was primarily meant to be a threshold for prime factors in $(p - 1)/2$. The particular values are presented in Table 1 [12].

Bitlength of p	512	768	1024	1280	1536	1792	2048	2304	2560	2816	3072	3328	3584	3840
q_{bit}	119	145	165	183	198	212	225	237	249	259	269	279	288	296
$\lceil 3q_{\text{bit}}/2 \rceil$	179	218	248	275	297	318	338	356	374	389	404	419	432	444

Table 1: Security boundaries used for El Gamal's signature computation.

It was deemed that such a modification will not affect security of the signature scheme, since the private key together with the nonces seemed to be still long enough. However, as was shown in [12], this "relative enough" was definitely not enough from a cryptanalytical viewpoint. An attack was presented allowing computing the whole private key from just only one signature known. The technique used for the attack was also built on lattice algorithms, *however a different approach than an HNP-based one was employed*. On one hand, this is a quite common situation, since there is a plenty of lattice-based techniques, so we can choose freely what we just want. On the other hand, it was a bit surprising, since HNP encapsulates benefits of several other techniques very nicely in a way that is perfectly tailored for the cryptanalytical problems arising in flawed implementations of El Gamal's and other DSA-like signature schemes. The reason was, perhaps, that existing HNP-based techniques were unable to use partial knowledge of the private key x itself, so it seemed that more than one signature would be necessary. As we will show, however, a small extension of HNP to OT-HNP allows solving the problem with just only one signature, too. Furthermore, we believe that retaining the connection of the attack with the HNP-based methods is beneficial from both theoretical and educational viewpoints.

3.2 Using OT-HNP

The cryptanalytical problem which stays behind breaking the implementation we recalled in §3.1 is stated as follows: We are given an El Gamal signature (r, s) of a known message m computed according to the description given in §3.1. Furthermore, we know that, because of a certain unfortunate optimization, it holds that $0 < k < B$, and $0 < x < B$, where B is given by Table 1 and generally $B < p^{3/8}$. The task is to find the private key x. We will show how OT-HNP can be used for such a purpose.

We start with the basic congruence for s:

$$H - xr \equiv sk \pmod{p - 1},$$

where H, r, and s are known values. Using the extended Euclidean algorithm, we find integers a, b satisfying $as + b(p - 1) = \gcd(s, p - 1) = c$. Multiplying the congruence by a, we get

$$a(H - xr) \equiv ck \pmod{p - 1},$$

where $ck < cB$ and with a high probability still $cB < p^{3/8}$. Setting $t = -arx \bmod (p - 1)$, $u = -aH \bmod (p - 1)$, $N = p - 1$, $\delta = cB$, and $X = B$, we can finally write:

$$(tx - u) \bmod N < \delta, \ 0 < x < X.$$

In this way, we get the OT-HNP instance (t, u, δ, N, X), where $\delta X < (N + 1)^{3/4} << N$, which we then try to solve by Algorithm 1 from §2. With respect to the formal elaboration given there, we have two obstacles: The first one is that N is not a prime and the second one that we did not prove a uniform distribution of t. Therefore, we regard Theorem 1 as a heuristic argument only, here. On the other hand, it is reasonable to expect similar theorem to be provable for the instances of OT-HNP we have. However, we decided to use an experimental verification of our approach instead, partly due to a well-known and important fact that lattice algorithms perform much better on a typical cryptanalytical problems in this area than what is guaranteed formally. The results presented in §3.3 fully support the hypothesis, that the instances of OT-HNP we have here, are practically solvable almost as soon as it holds $\delta X < N$.

3.3 Practical Experiments

The results obtained for a randomly generated El Gamal instances and signatures according to a flawed implementation described in §3.1 are presented in Table 2. Next, a rough study on how close to the size of the value of $p - 1$ can the sum of bitlengths of x and k be is given in Table 3. As we can see, we can even go over the size of $p - 1$. However, as we can expect, a closer inspection shows that the performance is then roughly the same as trying to guess several most significant bits of x and k directly and then to solve the “downsized” problem for each guess separately. The experiments were programmed in C++ supported by the well-known Shoup’s NTL library [16]. Basic parameters of the computing platform employed were the following: Windows 2000/SP 4, Intel Celeron/2.20 GHz, 256 MB of RAM.

Bitlength of *p*	**Bitlength of *x*, *k***	**Probability of Success**	**Time (in seconds)**
512	179	1	0.26
768	218	1	0.88
1024	248	1	2.23
1280	275	1	4.28
1536	297	1	9.26
1792	318	1	14.53
2048	338	1	22.47
2304	356	1	44.02
2506	374	1	66.36
2816	389	1	84.4
3072	404	1	116.67
3328	419	1	151.56
3584	432	1	333.12
3840	444	1	356.01

Table 2: Experimental verification of OT-HNP-based attack on the GPG-flawed El Gamal.

Bitlength of *p*	**Bitlength of *x*, *k***	**Probability of Success**	**Time (in seconds)**
1024	500	1	4.97
1024	508	0.99	5.08

1024	509	0.96	5.15
1024	510	0.88	5.25
1024	511	0.7	5.35
1024	512	0.37	4.99
1024	513	0.09	5.38
1024	514	0.03	4.97
1024	515	0.003	4.99
1024	516	0.001	5.19

Table 3: Experimenting with the size of *x* and *k*.

4 Conclusions

One-time HNP is a useful modification of HNP presented in [4] and further refined in [13]. It shows an easy and efficient way on how to utilise direct partial information on the hidden number being search for, too. Inability to use such information was outperforming, otherwise very elegant and efficient, HNP-based attacks when the task was to break certain flawed implementations of El Gamal's signature scheme using just only one signature [12]. We show that OT-HNP overcomes this obstacle. It is reasonable to expect that similar results are basically achievable for other DSA-like schemes which implementations are flawed in a similar way as well. Furthermore, OT-HNP also uncovers new promising directions for further refinements of the HNP-based approach itself.

5 Acknowledgements

The author would like to thank to Katarína Machálková for her technical help with the experiments, to Martin Hlaváč for pointing out typos in previous version of the paper, and to the members of the program committee of MKB 2005 for their comments.

References

[1] Babai, L.: On Lovász' Lattice Reduction and the Nearest Lattice Point Problem, *Combinatorica*, 6:1-13, 1986.

[2] Bleichenbacher, D.: Experiments With DSA, *CRYPTO 2005 – Rump Session*, Santa Barbara, USA CA, 2005.

[3] Bleichenbacher, D.: On the Generation of DSS One-Time Keys, *manuscript*, the result was presented at the Monteverita workshop in March 2001.

[4] Boneh, D., and Venkatesan, R.: Hardness of Computing the Most Significant Bits of Secret Keys in Diffie-Hellman and Related Schemes, in *Proc. of CRYPTO '96*, pp. 129-142, Springer-Verlag, 1996.

[5] El Gamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Transactions on Information Theory*, Vol. 31, pp. 469–472, IEEE, 1985.

[6] FIPS PUB 186-2: *Digital Signature Standard (DSS)*, National Institute of Standards and Technology, January 27, 2000, update: October 5, 2001.

[7] GPG – The GNU Privacy Guard, http://www.gnupgp.org.

[8] Howgrave-Graham, N.-A., and Smart, N.P.: Lattice Attacks on Digital Signature Schemes, *Design, Codes and Cryptography*, 23:283-290, 2001.

[9] Hlawka, E., Schoissengeier, J., Taschner, R.: *Geometric and Analytic Number Theory*, Springer-Verlag, 1986.

[10] Menezes, A.-J., van Oorschot, P.-C., and Vanstone, S.-A.: *Handbook of Applied Cryptography*, CRC Press, 1997.

[11] Nguyen, P.-Q.: The Dark Side of the Hidden Number Problem: Lattice Attacks on DSA, in *Proc. of the Workshop on Cryptography and Computational Number Theory (CCNT '99)*, pp. 321-330, Birkhäuser, Basel, 2001.

[12] Nguyen, P.-Q.: Can We Trust Cryptographic Software? Cryptographic Flaws in GNU Privacy Guard v1.2.3, in *Proc. of Eurocrypt '04*, pp. 151-176, Springer-Verlag, 2004.

[13] Nguyen, P.-Q., and Shparlinski, I.-E.: The Insecurity of the Digital Signature Algorithm with Partially Known Nonces, *Journal of Cryptology*, Vol. 15, No. 3, pp. 151-176, Springer-Verlag, 2002.

[14] Nguyen, P.-Q., and Stern, J.: The Two Faces of Lattices in Cryptology, in *Proc. of Cryptography and Lattices – CALC'01*, pp. 146-180, Springer-Verlag, 2001.

[15] PKCS#1 v2.1: *RSA Cryptography Standard*, RSA Labs, DRAFT2, January 5 2001.

[16] Shoup, V.: *Number Theory C++ Library (NTL)*, http://www.shoup.net/ntl/.