Dual Discrete Logarithms

Benne de Weger

Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands.
E-mail: b.m.m.d.weger@tue.nl

version 1.0, October 15, 2007

To Henk van Tilborg, on the occasion of his 60th birthday.

1 Introduction

WHAT'S THIS? YOU'VE INTERCEPTED ONE OF THE ENEMY'S SECRET MESSAGES!
WE ATTACK AT MIDNIGHT!
HMM
OH, REALLY? BY GOLLY, YOU'RE RIGHT!
WE'VE BROKEN THEIR CODE!!

Recently Hung-Min Sun, Mu-En Wu, Wei-Chi Ting and M. Jason Hinek [SWTH] introduced *Dual RSA* as a variant of plain RSA, in which two key pairs share the public and private exponents, but have different moduli. Their main motivation for this variant is to enable storage savings for users who operate multiple key pairs on the same machine. A similar motivation was behind *Twin RSA*, introduced in 2005 by Arjen Lenstra and the present author [LdW2], where a construction was given for secure RSA moduli that differ by 2 only (à la *twin primes*).

(As a side remark we notice that the idea for Twin RSA emerged from the desire to construct pairs of secure RSA moduli with a prescribed difference, related to the construction of colliding public key certificates based on hash collisions, see [LdW1].)

In [LdW1], [LdW2], corresponding "twin" problems for Discrete Logarithms were mentioned, i.e. it was asked whether it would be feasible to construct what might be called *Twin Discrete Logarithm* instances. An example is, for given prime p and generator g of a subgroup of $(\mathbb{Z}/p\mathbb{Z})^*$, to find elements g^{x_1}, g^{x_2} that satisfy $g^{x_1} - g^{x_2} = 1$.

Those problems might be quite nontrivial, and as far as we know no ideas for the efficient construction of Twin Discrete Logarithm instances have emerged, whereas the Twin RSA problem was readily solved.

Similar to the Twin RSA case one might wonder about the Discrete Logarithm analogue of Dual RSA, and that is exactly what is done in this note. It will be shown that in some sense the situation is exactly the other way around compared to the Twin case, in the sense that the construction of instances is somewhat cumbersome (though not infeasible, as shown in [SWTH]) for Dual RSA, but quite easy for Dual Discrete Logarithms. This short note will only introduce some easy concepts and obvious constructions, and will not deeply analyse all sorts of variants and their security. Instead a list of open questions is given, to stimulate research for anyone who, for whatever reason, might care to spend some time on this.

2 Dual Discrete Logarithms and Dual Diffie Hellman key pairs

In this section we discuss the generation of instances of Dual Diffie Hellman key pairs. To generate Dual RSA instances, the paper [SWTH] fixes the public and private exponents d, e, and asks for a pair of secure RSA moduli $n_1 = p_1q_1$ and $n_2 = p_2q_2$ such that the RSA key equations $ed \equiv 1 \pmod{\phi(n_1)}$ and $ed \equiv 1 \pmod{\phi(n_2)}$ both hold.

In the Discrete Logarithm setting (we only look at the case of prime fields $\mathbb{Z}/p\mathbb{Z}$), there may be different ways of fixing some parameters among Diffie Hellman key pair pairs. A completely trivial and uninteresting way is to fix a generator g and a prime p, and take different private and public keys $x_1, x_2, y_1 \equiv g^{x_1}, y_2 \equiv g^{x_2} \pmod{p}$. Another equally trivial way is to fix a generator g and a private key x, and to find two random different primes p_1, p_2 and public keys $y_1 = g^x \pmod{p_1}$ and $y_2 = g^x \pmod{p_2}$. Or to fix a prime p and a private key x, and to find two random generators g_1, g_2 and public keys $y_1 = g_1^x \pmod{p}$ and $y_2 = g_2^x \pmod{p}$.

Slightly less uninteresting is to fix a generator g and a public key y, and to find two random different primes p_1, p_2 and private keys x_1, x_2 such that $y = g^{x_1} \pmod{p_1}$ and $y = g^{x_2} \pmod{p_2}$. When first two primes are chosen, the problem of finding the private keys is just two instances of the Discrete Logarithm Problem, and thus can be supposed to be hard and therefore uninteresting. When first two private keys x_1, x_2 is chosen, it's not clear how to find in an efficient way primes p_1, p_2 dividing $g^{x_1} - y$ and $g^{x_2} - y$, unless trivial x_1, x_2 were chosen. But maybe something clever can be found here.

The most close to Dual RSA seems to be the case where we fix the public and private keys x and y (which we'll view as elements of $\mathbb{Z}$), and ask for a pair of system parameter sets $\{p_1, g_1\}$ and $\{p_2, g_2\}$, where p_1 and p_2 are distinct primes of about the same magnitude, such that x and y form a Diffie Hellman key pair in both settings, i.e. the following Diffie Hellman key equations hold simultaneously:

$$\begin{cases} g_1^x \equiv y \pmod{p_1}, \\ g_2^x \equiv y \pmod{p_2}. \end{cases}$$

Clearly, for a user who operates multiple Diffie Hellman key pairs with different system parameters on the same machine, such an instance of Dual Diffie Hellman key pairs enables a saving of 50% in storing the private as well as the public keys.

The main question we address is how to efficiently create instances of Dual Diffie Hellman key pairs. One might wonder in the first place whether such instances exist at all, but as we allow a lot of freedom in all the parameters p_1, p_2, g_1, g_2, it is readily seen that there should be sufficiently many solutions. We only have to find them. Here is a baby example.

Let us take $p_1 = 3\,646\,060\,591$ and $p_2 = 4\,186\,435\,763$ as prime numbers, and $g_1 = 2\,454\,186\,096$ and $g_2 = 754\,870\,076$ as generators. With $x = 2\,101\,907\,279$ and $y = 1\,420\,880\,381$ we now indeed have

$$\begin{aligned} g_1^x &= 2\,454\,186\,096^{2\,101\,907\,279} \equiv 1\,420\,880\,381 = y \pmod{p_1}, \\ g_2^x &= 754\,870\,076^{2\,101\,907\,279} \equiv 1\,420\,880\,381 = y \pmod{p_2}. \end{aligned}$$

One easy construction is as follows. We start with random primes p_1 and p_2 with $p_1 < p_2$, a random integer g_1 with $2 \leq g_1 \leq p_1 - 2$ as first generator, and a random integer x with $2 \leq x \leq p_1 - 2$ and $\gcd(x, p_2 - 1) = 1$, as private key. Then the public key is computed as $y = g_1^x \pmod{p_1}$, where y is interpreted as an integer with $0 < y < p_1 - 1$. Next $z = x^{-1} \pmod{p_2 - 1}$ is computed, and now we can compute $g_2 = y^z \pmod{p_2}$.

A second construction, just as easy, is to start with random primes p_1 and p_2 with $p_1 < p_2$, and random integers x and y with $2 \leq x \leq p_1 - 1$ and $2 \leq y \leq p_1 - 2$ and $\gcd(x, p_1 - 1) = \gcd(x, p_2 - 1) = 1$, as private and public keys. Then the two generators are computed via $z_1 = x^{-1} \pmod{p_1 - 1}, g_1 = y^{z_1} \pmod{p_1}$ and $z_2 = x^{-1} \pmod{p_2 - 1}, g_2 = y^{z_2} \pmod{p_2}$.

Interestingly, the second construction can be related to RSA encryption and decryption. If we take $n = p_1 p_2$ as RSA modulus and $e = x^{-1} \pmod{\phi(n)}$ as RSA public exponent, then the solution of Chinese Remaindering $g_1 \pmod{p_1}$ and $g_2 \pmod{p_2}$ is just the RSA ciphertext obtained by encrypting the public key y. Unfortunately it's totally unclear what the use of this observation is.

It is interesting to note that our constructions have a lot of freedom built in, seemingly not imposing any structure that could be used in a priori making the security of the constructed Diffie Hellman key pairs questionable. We think that in this respect the constructions outlined here have a more robust feel to them than the construction of Dual RSA instances proposed in [SWTH], which does add a lot of additional structure (such as a rather big common divisor of $\phi(n_1)$ and $\phi(n_2)$).

The last case to be studied is that of fixing a prime p and a public key y, and looking for generators g_1, g_2 and private keys x_1, x_2 such that

$$\begin{cases} g_1^{x_1} \equiv y \pmod{p}, \\ g_2^{x_2} \equiv y \pmod{p}. \end{cases}$$

Clearly this cannot be done by first generating random generators (unless one can solve Discrete Logarithms), but it should be done by first generating the private keys, and then the generators by extracting the roots as in the construction described above. We leave further details of this case to the reader.

3 Open questions

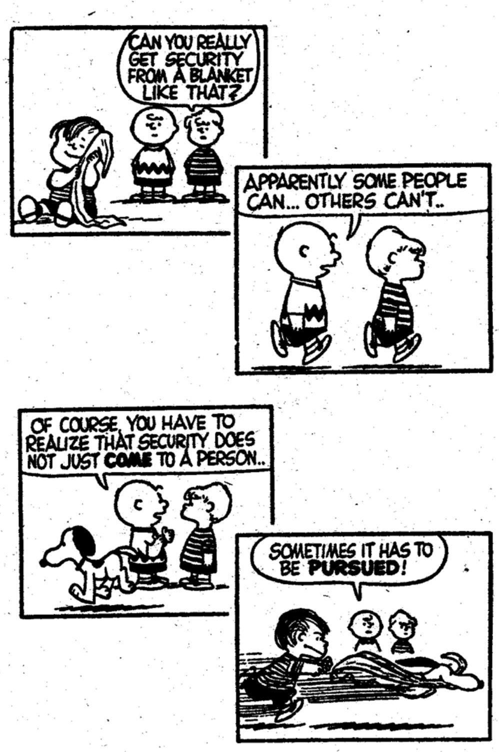
Of course the first question to ask is about the security of Dual Diffie Hellman key pairs. We could formulate the *Dual Discrete Logarithm Problem*, or *DuDL Problem*, as follows:

The DuDL Problem
Given primes p_1, p_2 and generators g_1, g_2 and a positive integer $y < \min\{p_1, p_2\}$,
find the positive integer $x < \min\{p_1, p_2\}$ (if it exists) such that simultaneously
$g_1^x \equiv y \pmod{p_1}$ and $g_2^x \equiv y \pmod{p_2}$.

Another version is to leave y out of the problem.

The Second DuDL Problem
Given primes p_1, p_2 and generators g_1, g_2,
find a positive integer $x < \min\{p_1, p_2\}$ (if it exists) such that
$g_1^x \pmod{p_1}$ and $g_2^x \pmod{p_2}$, seen as positive integers $< \min\{p_1, p_2\}$, are equal.

Clearly, if one can solve the Discrete Logarithm Problem (for any of the primes p_1, p_2), then one can solve the DuDL Problem, and if one can solve the DuDL Problem and indeed a solution is found, then one can solve the Second DuDL Problem.


CAN YOU REALLY GET SECURITY FROM A BLANKET LIKE THAT?
APPARENTLY SOME PEOPLE CAN... OTHERS CAN'T..
OF COURSE, YOU HAVE TO REALIZE THAT SECURITY DOES NOT JUST COME TO A PERSON..
SOMETIMES IT HAS TO BE PURSUED!

One gets the feeling that the DuDL Problem should be essentially easier than the Discrete Logarithm Problem, as twice as much information is given away. Can this be exploited, e.g. in a Pollard-ρ-type method, to get a better running time than the square root of the prime size? Is there reason to believe that, say, the DuDL Problem may allow a fourth root Pollard-ρ attack? And what about the subexponential methods: can they take advantage of the additional information?

And what about the difficulty of the Second DuDL Problem?

What about constructions for special cases of Dual Diffie Hellman key pairs, such as $g_1 = g_2$? Do such instances exist at all, and if yes, can one find some?

Then of course the question comes up about DuDL in other Discrete Log settings, such as finite fields of characteristic 2, subgroups of $\mathbb{Z}/p\mathbb{Z}$ with large cofactors, and (hyper)elliptic curves. Let us describe the case of elliptic curves.

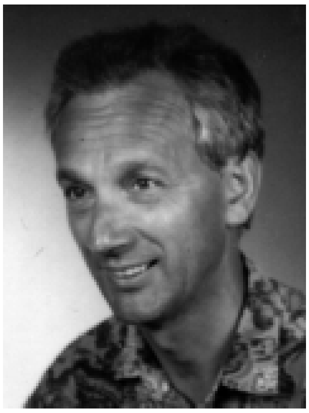
One variant that comes to mind is the following. Let E_1, E_2 be two elliptic curves over prime fields $\mathbb{F}_{p_1}, \mathbb{F}_{p_2}$ respectively, being nice groups of appropriate sizes and without any special structure that might be exploited by a ECDL solver. Suppose that we know the orders of the full Elliptic Curve groups. Then the second construction given above translates as follows: first find random points Q_1, Q_2 that share their x-coordinates (that's easy), and generate a random private ECDH key $k \in \mathbb{Z}$ of the proper bitsize, that is coprime to the group orders. Then compute the inverses of k modulo the group orders, and thus we can easily find base points P_1, P_2 such that the corresponding public ECDH keys are $Q_1 = [k]P_1$ and $Q_2 = [k]P_2$. By the point compression technique this would again imply an almost 50% saving in storing both the private and public keys.

Finally an interesting open question is: is all this really useful? Are there any nice applications, other than the somewhat uninteresting storage saving?

Acknowledgements

The author thanks Arjen Lenstra for comments on this note, and the Dutch Railways for kindly providing the mobile office space in which most of this note was devised and written.
Cartoons by Charles M. Schulz, © United Feature Syndicate, Inc.

Much more importantly, the author is very grateful to Henk van Tilborg for creating a wonderful working enviroment and atmosphere in the Eindhoven crypto group.



The picture above shows Henk as he looked at least 10 years ago.

References

[LdW1] A.K. Lenstra and B.M.M. de Weger, "On the possibility of constructing meaningful hash collisions for public keys", in C. Boyd and J.M. Gonzalez Nieto (Eds.), *ACISP 2005*, LNCS 3574, pp. 267-279, 2005.

[LdW2] Arjen K. Lenstra and Benjamin M.M. de Weger, "Twin RSA", in E. Dawson and S. Vaudenay (Eds.), *MyCrypt 2005*, LNCS 3715, pp. 222-228, 2005.

[SWTH] Hung-Min Sun, Mu-En Wu, Wei-Chi Ting, and M. Jason Hinek, "Dual RSA and its Security Analysis", *IEEE Transactions on Information Theory* 53 [2007], 2922–2933.