Chapter 9

Migrating to code land

Of all the chapters of this book, this one will probably be obsolete first. Technologies, laws and behaviors evolve so fast today, especially around computers and networks that by the time you read this parts if not all of what follows will be false, nonsens, possibly unlawful. Yet paper's qualities, as a static, non updatable media compare favorably to its shortcomings as a static, non updatable media.

The change in a single paragraph in common dictionaries reveals the extent of the change brought about in our lives by the quantum leap of active code (see previous chapter) at the dawn of the third millennium. Fifty years ago, dictionaries defined a "computer" as a human being who performed calculations. It's now defined as an electronic machine and there is no reference to human beings whatsoever. Active code has taken over.

Coded Golems

Active code is the best promoted product humanity has ever sold itself. However slow, cumbersome and limited they were in reality, as soon as computers came into existence in the 1950's, they were exalted and became invested with the most extraordinary possibilities. Although, at the time, they could hardly do more than store data and sort it as high speed, they were credited with awesome intelligence. Businesses and governments trusted blindly their imagined powers and invested in their development. "Computer assisted" became a synonym for being equipped with a magic wand. No serious activity could do without one as its tool. Armies, administrations and businesses empowered them as their most precious adjunct. Companies were often endangered or even collapsed on coding their accounts departments, yet no one thought to blame the holy computers.

This irresistible attraction stemmed from the combined seductions of hidden content and rapid management. Rapidity has always been the key quality of superior machinery. A hidden and coded content immediately draws on the poetic substance of all the romantic power and mystery accumulated by codes and ciphers over the centuries. Symbolically, coded data is more than mere data: what coding does is that it transmutes data - lifts it into a superior dimension where obscure mythical processors then yield golden results.

When personal computers became available in the 80's, we could hardly wait to equip every single individual with the wonder tool. Twenty years on, billions of people have, in their pockets, a machine, familiarly called a phone, which is, in fact, more powerful than a strategic computer of the sixties and no one is cheated: pocket computers do, indeed, deliver the goods - speed, concealment and the wonder of entertainment.

A virtual Eden

All this laid the ground for the most ambitious epic feat humanity had ever undertaken. For once, instead of discovering new territory, we created it. Spontaneously, irresistibly, it simply happened over a matter of a few years.

In the early nineties, before the magic link of hypertext, even, the simple magic of exchanging text and pictures through modems and phone lines was seducing those geeks who could access it. They began to invest the embryonic internet with a spatial existence of new dimensions. It was the dream come true. Users were disembodied entities meeting with other disembodied entities and sharing free material, more often than not wild erotica. Sheriffs were nowhere to be seen and neither were there any real brides you could meet. Imagination was all powerful. The first generation of internauts didn't mind having to type the exact address of a site or search through the frustratingly textual directories of servers. The era of File Transfer Protocol was a golden age for geeks. They thought of the keyboard as magic contact with code. It naturally excluded the crowd of straight users who did not enjoy typing their way through code land. Enthusiastic geeks fed the network with data - you could say they "enjoyed every bit of it".

Born again internauts

The magic click of hypertext was the final ingredient needed to turn the Internet into a virtual Eden. Clicking on a word and being teleported to some other page, anywhere, at some unspecified point in the virtual sphere, enhanced the feeling of freedom.

Internet was freely delivering what millenniums of brotherhoods had painstakingly worked at achieving through their secret rites. Internauts were not only disembodied, they were born anew into a fresh, new life with the immediate benefit of unlimited travel through an utterly free universe. Abandoning their bodies on the old world for hours, they wander in the virtual world of cost-free data.

This Virtual Eden aspect of the Internet was strongly supported and fed by a founding myth: that of Freedom from Hierarchy. The Internet was an Eden free of an overseer, an all-powerful deity. Anyone could bite into the fruit of knowledge without risk. Indeed, for irrelevant reasons, due to a military past which had

produced but now had no further connection with its present, the Internet had no hierarchical structure. The myth was actually based on solid reality. In contrast to what humanity has always known and experienced, there is no central power controlling the Internet. Heritage of the Cold War, by-product of past issues of survival through a world war, the network was engineered so as to survive all local disasters. There is no center whose disappearance could cause the whole network to collapse.

A phenomenon similar to an emigration started to take place. The first step - preparation for the voyage - started in the 80's and early 90's. Humanity had begun packing its suitcase, so to speak. We'd been storing our data, intellectual work and entertainment on the far side of the screen, in the closed box of the hard disks buried in our computers.

The second step - migration into the virtual space of the network - is still going on now. It's already reached such a point of accomplishment as to the be irreversible. Our hard disks themselves are migrating, following our mail and our business.

A more critical issue is that the virtual world hosts coded counterparts of our identities and property, and that our physical existence and property are now so closely linked to their virtual counterparts as to be inseparable.

Privacy and security

These days, network communication security concerns everyone, individuals and businesses, governments and administrations. Cipher code is back in the forefront, expected to solve all the protection problems created by its twin active code. The present scenario is a veritable carnival of contradictory demands for protection: individuals wanting to be protected against criminals and against administrations abusive of their power, administrations seeking protection against cheating individuals, business wanting to be protected against thieves, governments against enemies, foreign or otherwise, etc, etc. The list is endless.

Balancing out secrecy

Fortunately, though played out here on the coded scene of the virtual world, the problem of freedom in a contradictory world of conflicting interests is an old one. Thanks to the philosophers of the 18th century, we know that there is a solu-

tion in a balancing of forces between competing powers. This is the cornerstone of all democracies and it works in the virtual world as well. It's not perfect, but it does help maintain a more-or-less livable society.

In the world of code we live in, the key to this balance is a numerical cipher key. Any user, private or communal, has free access to highly sophisticated encryption software of a quality and efficiency no Venetian diplomat of the seventeenth century could have dreamed of. Anyone can instantly encipher a text which may then be equally instantly deciphered by a chosen correspondent simply through their having the correct numerical "key" and without either of them possessing the least knowledge of cryptography. PGP, Pretty Good Privacy, was the emblematic original tool, free software now developed into many and diverse directions.

Numerical keys can, of course, be broken, but this requires the brute force of a super computer and can be achieved only with expensive machines and at the cost of a non-negligible delay. It is this expense and delay that are the keys of today's security balance.

The issue is of course controversial.

The Diffie-Hellman-Merkle paradoxical key exchange

It sounds like an impossible magic trick: how do you exchange a secret number with your partner while everybody's looking at you and watching your hands?

As with most cipher systems since the time of Caesar, internet cryptography is based on the exchange of a secret key between trusted partners. For Caesar's code, the key was an alphabet jump, a number between 1 and 25. For Vigénère, it was the secret word repeated beneath the message. Here, the key is large number, 128 or 256 digits long.

For anyone who doesn't know the simple arithmetic hidden in the system, it looks like magic. Let's say you and I want to exchange secret information through email. First we agree on two numbers we don't bother to hide. Then, even though the first numbers are perfectly visible to eaves-droppers and we keep on using emails anyone can read, we use those numbers to produce together a new number so secret that we can use it as a key to protect all further exchanges with a high degree of security.

Here is how our exchange might look:

A: Hi. I suggest we base our encryption on 3 and 10

B: OK. 3 and 10 are fine by me. No problem

A: Good. Note this number: 11

B: OK, fine. You note number 14, too

A: Thank you for your 14, I can now compute our secret numerical key

B: Thank you for your 11. I have computed our numerical key, too, and I'm perfectly sure we have the same key and that no one else knows it. Here's a text encrypted with that key.

A: Roger. I've deciphered your text and read it perfectly clearly.

A and B have simply and quite openly exchanged only four numbers: 3, 10, 11 and 14. They haven't hidden them yet they're certain they can use them to produce a common key number nobody else will know. Better still: they've done the trick according to rules everyone knows. Roughly, the "magic" comes from the fact that nobody but A knows exactly how he "seeded" the system to produce his number, Na, and nobody but B knows exactly how he "seeded" the system to produce his, Nb. The security is exactly as good as A's and B's ability to keep their own seeds secret.

The mathematical magic involved was the work of three independent mathematicians, Whitfield Diffie, Martin Hellman and Ralf Merkle, who first worked separately but eventually joined forces. Their combined genius produced a system that instantly superseded years of research by established research laboratories.

The system key relies on two number gimmicks easily understandable with nothing more than high school math, namely the prime numbers and their powers we've already seen at work along with Gödel numbers in chapter 2. It also makes use of the "modulo" function (number p modulo number q is the remainder of a division of p by q; for instance, 43 modulo 10 is 3 and 22 modulo 7 is 1).

The first number exchanged is the "base". All numbers exchanged thereafter will be powers of that base. Had the base, which must be a prime number, been 3, the conversation would have dealt with the powers of 3: 9, 27, 81, etc.

The second number exchanged is the "modulo" reference. If that number is 7, then all numbers exchanged after this would be remainders of their division by 7.

To start the real exchange, A picks a prime number nobody else, not even B,

will know. He raises the base to the power of that number then computes the result by the chosen modulo. Let's say he chooses 11. He then computes

311 = 177 147
and applies the modulo:
177147 modulo 10 = 7
and sends 7.
Likewise, B chooses 14. Then he computes
314 = 4 782 969
and applies the modulo:
4 782 969 modulo 10 = 9
and sends 9
A computes with his secret key 11:
911 modulo 10 = 31 381 059 609 modulo 10 = 9.
B computes with his own secret key 11:
714 modulo 10 = 678 223 072 849 modulo 10 = 9

In both computations each secret number is input only once, raising the base to its power exactly once. The modulo operation is there to enhance security. It is transparent for exponentials and serves merely to hide the original numbers. The modulo step presents no way backward. The modulo function cuts off the way back. It's impossible to determine what number any given number is the modulo of: there are too many possibilities. 3 could just as easily be 23 modulo 10 as 293 modulo 10.

The key is

(base)(Na + Nb) modulo 10.

Of course, in this example the numbers are too small. Simple trial and error would solve such a key rapidly. With larger numbers and a larger modulo, hence a key with a higher number of figures, a trial and error method would require long and expensive computation.

The random gang

Present day secret codes depend on a strange gang: random numbers. These would have thoroughly confused Pythagoras because you can use them but you

can't see them, and whenever a random number is identified as such, it vanishes as a random number.

We resort to random numbers because of a basic strategy principle. When an opponent knows all our weapons and tactics to the point of being able to anticipate our moves, our only way out is to surprise him by choosing moves in a random manner. This strategy is beautifully described by Vladimir Nabokov in his novel "The Luzhin Defense" where chess champion Luzhin plays random moves on the board and in real life in a desperate attempt to confuse adversaries.

Here, in the open/hidden key exchange, security in its entirety rests on the secret choice of our own key subsequently to be concealed in large integers and their modulos. If an eavesdropper can predict and guess our key, the whole security system collapses, so, to avoid such a risk, the system relies on random numbers. Somewhere in our computer, a chunk of code generates a random number every time we need a numerical key.

Absolute pure randomness is a mere dream, out of the reach of human hands and certainly out of the reach of algorithms such as computer programs. We approximate randomness every time we throw a dice or shuffle a deck of cards. Computer programs resort to Pseudo Random Numbers, an approximation of the real thing. They're the weak link in the security chain, a potential backdoor.

Coding humans

In an unexpected loop, the Pythagorean philosophical system eventually arrives back on the scene to install a pure Pythagorean world. The Pythagorean view that only numbers are real and that all the rest is illusion is fast becoming a realistic description of our present day world. We depend on our digital, coded counterparts in the networks. Administration and businesses deal less and less with our physical bodies and more and more with our digital representations. Our freedom is now defined more by what our codes can do than by what our physical bodies can. Bodies follow codes. That part of our life which depends on codes and dwells on the net is growing. Digital impersonation on the web is becoming as dangerous as kidnapping and murder in the physical world.

An illustration of this is the new and widespread criminal activity, identity theft. Using "phishing", the rapidly developing activity of fishing for personal data on the web, criminals get hold of credit card numbers, social security num-

bers and Internet passwords and then impersonating their victims in many situations. Victims lose homes and belongings. Some are even prosecuted for criminal acts committed in their name. In these cases, the process of identity retrieval is long and difficult. Showing up in person is of little help: numbers tend to prevail over physical bodies.

The Turing test, level 2

A side issue of this situation is an ironic twist of the famous Turing Test proposed by Alan Turing for testing for artificial intelligence. These days, in contrast, machines need a test to be able to tell machines from humans.

Crowds of software robots roam the Internet in a continual attempt to access protected sites. Because of their ability to try out millions of name and password combinations in a split second, these are far more dangerous than ordinary dishonest humans.

Dealing with such situations, we've probably all been asked to copy a "captcha", a word written on a background designed to confuse automatic scanning. Lucky for us there still are activities that can help differentiate us humans from machines!

The Cantor sieve

Simple property theft is not the worst of developments in our emigration to Code Land. After all, it concerns only our link to earthly goods and, as most religions teach us, we'd be better off without those, especially as we all know we're eventually going to have to leave them behind us when we quit this world.

The deeper, more insidious danger comes from digitizing the physical world into the virtual one and then trying to manage it through this digitized counterpart. This threatens to downgrade reality to a sort of sub-world.

In chapter 2 we saw that Georg Cantor had demonstrated the existence of at least two different infinities. Aleph-0 is the infinity of whole numbers, Aleph-1 is a larger infinity, the set of points in a 3-D universe for instance. There may be other infinities between these, but that doesn't concern us here. The more important point is that Cantor proved that there can be no one-to-one correspondence between the worlds of Aleph-0 and Aleph-1.

Today, since they are based on integers and their siblings, the rational num-

bers, Aleph-0 is everything our codes and our computers can manage, but we are and live in Aleph-1.

We should probably dismiss the problem as a preposterous waste of time, and yet it takes us straight back to Pythagoras. Two is in Aleph-0, the square root of two is in Aleph-1 and never the twain shall meet! A ratio or decimal number can approximate the square root of two as closely as we like, but it will never BE the square root of two.

Are we humans, residents of Aleph-1 (and perhaps even higher if we take our imagination into account) being treated fairly through our Aleph-0 shadows? Can Aleph-0 code be expected to adequately manage the Aleph-1 world?

We living creatures bask in the upper world of Aleph-1, enjoying each second of the limitless spread of organic life and of our thoughts and potentials. When tagged with Aleph-0 numbers, we seem to become conveniently predictable, an easily sortable database. Yet our trace in Aleph-0 is the merest shade of what we are in Aleph-1. If Aleph-0 is used only by computers trying to study and understand human beings, the situation is not so bad. Only our superficial numerical Aleph-0 skin is concerned and our Aleph-1 freedom remains intact. If it goes further and Aleph-0 is used to manage our lives, we could be in deep trouble.

Aleph-0 is to Aleph-1 what a DVD is to a live concert.

The fifth frontier

Software developers are currently fighting on an elusive frontier: code perfection. Their goal - producing bug-free codes - seems simple enough. Bugs, however, take many shapes. They can be an overlooked logical loop that crashes a product or an open backdoor letting malevolent code install itself on the computer. To the constant surprise of both users and publishers, no code ever comes out on the market bug-free. Apparently unavoidably, imperfections do crawl into code-land itself, for all that it's expected to be free of all flaws of the traditional human realm. Internauts are now used to receiving a stream of patches fixing this or that in nearly all software. Usually described as enhancements of the user's comfort or security, in fact all they do is eliminate bugs.

The solution seems simple enough and easy to implement from the outside. A precise analysis of projects, clear distribution of tasks and coding by profes-

sional programmers verified by further professionals should yield perfect products. Nevertheless, no matter how many top-level dedicated geeks a software company uses on a project, the next thing they know, other geeks are exposing bugs and trap doors and advertising them. Patches, etc., follow.

This doesn't mean that the first geeks were idiots and the second lot geniuses. It's much more important in that it reveals the present state of the code world. It means a new condition - interaction - is gaining in importance. Once released, and "out there" in the world of networks and users, software is immersed in a soup of rich interactivity with other software. The best-tested code leaps into a state of uncertainty greater than coders and testers can manage. It seems as if conceivers of code are going to have to take a new dimension into account. Beyond the three dimensions of space, beyond time, which plays its role too, what we're talking about is the dimension of interaction.

In this fifth dimension, a code-versus-code competition that started timidly with World War II's Enigma and its competitors takes place. We still are the basic artisans, the hands on the keyboard, but an overall understanding of the general mass of interactions is already somewhat beyond our grasp.

The MMOG bride

As if irresistibly, a bachelor machine situation has emerged on the Internet (see chapter 8). The myth - freedom from hierarchy - was there from the start. The program and involvement of human users were, so to speak, built in. The fourth ingredient, the dimension of public show, came last. It first appeared as textual activity in forums. Internauts loved examining one another's opinions on every possible subject.

With better computers and a faster network, the show has set in with full scale sound and image. Certain Internauts have installed online cameras so the whole world can look at them 24/7. Currently the most popular and profitable sites either let Internauts see each other or let them exchange music and videos. The ultimate spectacular codes are the "persistent" worlds. Wearing sophisticated masks, Internauts go there to see and be seen and enjoy a lawless life beyond even the basic lawlessness of the Internet.

The first creators of such worlds thought they needed a gaming incentive and offered complex worlds of war and competition. They called their worlds "Massively Multiplayer Online Games". They soon discovered that most

Internauts were there only to look around and meet each other. That has given birth to a new generation of persistent worlds oriented only toward offering a new life, a new deal in Code Land. Internauts own land and houses, do business, create objects and earn virtual money that is actually convertible into dollars.

Persistent virtual worlds are parallel worlds to ours, but with a big difference: the internauts are to be seen, they're part of the general show. Probably the ultimate, lived-in, bachelor machine.