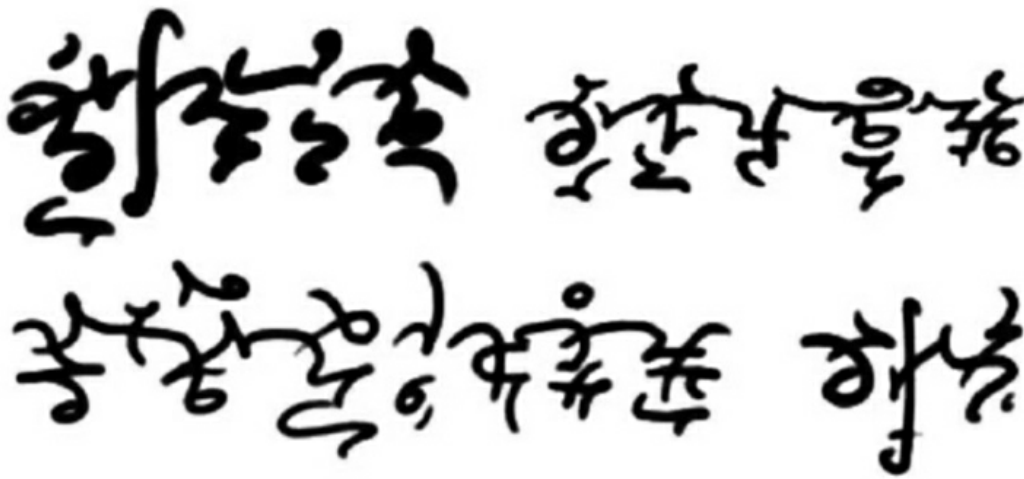


Advanced Handwriting Cryptography

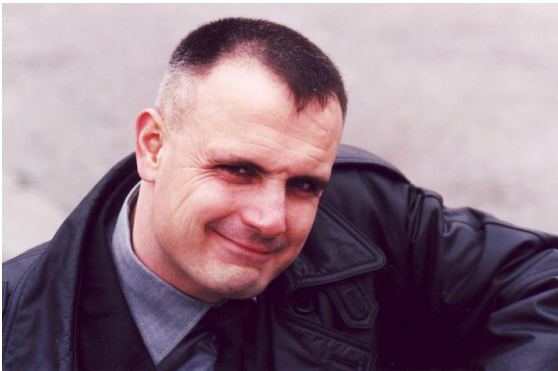


Alex A. O. Kobold

Advanced Handwriting Cryptography

(Full Edited Version 2016.09.18)

by Alex A. O. Kobold



Legal Notice.

The author of this book have used his best efforts in preparing this book. The author makes no representation or warranties with respect to the accuracy, applicability, fitness, or completeness of the contents of this book. The information contained in this book is strictly for educational purposes. Therefore, if you wish to apply ideas contained in this book, you are taking full responsibility for your actions.

The author disclaims any warranties (express or implied), merchantability, or fitness for any particular purpose. The author shall in no event be held liable to any party for any direct, indirect, punitive, special, incidental or other consequential damages arising directly or indirectly from any use of this material, which is provided “as is”, and without warranties.

1 Introduction.

Have you ever been in a situation where you needed to write up some private data in ways no-one else could read? Have you ever forgotten any access codes to your mails, logins to websites, codes of bank accounts, etc.? If not yet then perhaps the logins and codes you're using are very simple and easy to crack, or you're using them often enough not to forget them. As there's a growing number of the services that require to remember certain access data, and certainly you're not going to use all of them whole the time, there's coming a day when you'll be in trouble while trying to access one of the sites you urgently need to use, but have forgotten your password.. or even the login itself. So it would be very handy to have your own secure way of writing down the personal access data, a kind of personal secret language.

In this book I'm going to explain the basic principles for creating a highly secure personalized handwriting code, an individual cryptography, and how to operate it securely without compromising it by your own unwilling giveaways of the code. At first it may seem as a complicated thing to take on, especially considering that I have spent over thirty years creating and improving the encryption method, but while you learn it you'll see how easy it actually is. The reason it took so long time to find a practical solution is that it was hard to come up with a method which is easy to learn and simple to use, possible to personalize with little effort, yet secure enough against attacks by those trying to access your private data.

The advanced handwriting cryptography method described in this book is safer (can be safer when used properly) than any computer encryption can ever be. The reason for this is that no online interceptions, no screen shots of your computer created by hidden programs and accessed by hackers, no tapped keyboards, no spy cameras hidden around your places of work, nothing will reveal the messages hidden in the cipher text -- your private sensitive information will be encrypted/decrypted in your head on the go and only encoded information will see the daylight. The code will be totally safe even if there is the plaintext out there, of something you wish to encrypt, if you are careful enough to separate plaintext from the cipher text never keeping the two in the same locations, and never relating the two in the ways someone could guess the match in the messages.

When using the method only yourself it is useful for encrypting..

- .. passwords/access codes of your credit/debit cards;
- .. logins and access codes of online services;
- .. private financial deals;
- .. places, times and tasks of confidential character;
- .. names and contacts of selected people for their security;
- .. personal opinions about your business partners in your contacts;
- .. parts of personal diary with delicate information;
- .. written information of a great value to your adversaries;
- .. any written information for your personal safe use only.

There are additional benefits of the method while sharing encrypted messages with others, particularly with several recipients. These can be hard to implement, requiring some work in developing the code, but once the code is created and shared with trusted individuals then it can be used during entire lifetime. The specific benefits of sharing encrypted letters with others, compared to sharing unencrypted letters, are the integrity of the messages (no part of the messages can be removed without receiving parties noticing it), safety against loss of messages as every party involved can have the same letter (with exactly the same cipher text but differently encrypted message to each, with a different content when needed), and more. Details are discussed in the book.

On top of that the method can be used for marketing, entertainment, commercial and public safety needs, which are described in the book:

- product promotion;
- games and lotteries;
- cryptic signatures;
- cryptic orders;
- safe message boards;
- .. and much more.



2 A bit of history.

As I mentioned it took quite a long time to work out the method – over thirty years. In fact I didn't work on the project whole the time, not the 30 years straight. The encryption was my hobby since around the age of 10-12 y.o., with some very basic encoding methods for kids. In earlier times I tried to come up with a personal code or method of encryption just sometimes, on free time besides other hobbies like music, philosophy of mind, psychology, martial arts, programming and travel. Later I dedicated on it much more time as it was becoming more and more of a challenge.. I almost new “for fact” the solution is out there in the basic principles I have already learned, discovered and created, but the “perfect” method still remained elusive for me. The total net time I spent working on the handwriting cryptography may be around a few years of whole day work. During that time I have created perhaps a dozen fully complete codes, but each time I wasn't happy enough with the result. After using yet another method for some time it was becoming so easy to encode and read the code that it created in me doubts about the security of it. So each time I made a more complex method of encryption. The most complex code had thousands of encryption symbols to remember, so I had to repeat all of them in a systematic manner, not to forget. It became an obstacle by itself.

In 2004, while in Japan, it took me over half a year to create the most complex version I had so far, from beginning to completion. After the method was complete I encrypted my pocket notebook (the paper one, not electronic) with all the names, addresses, telephone numbers etc., with hundreds of contacts from around the world, destroying the original notebook. The encryption process of the notebook took around a week or two, don't remember exactly, including the time of verification for eliminating any errors. When I moved back to France where I lived before going to Japan it took me about two months of work every day six to eight hours to decrypt it all back. It took so long time mostly because by this time I had forgotten some of the symbols of the code and I needed to go through the process of recovering the meaning of them. It was a proof that this method of encryption wasn't practical for use when so much work was needed to keep the code secure. For substitution of real symbols it included several sets of code letters, code numbers and other encryption symbols for writing down different things: for encoding names, logins and codes I used separate sets of symbols, for telephone numbers, dates, counting etc. yet more sets of symbols and so on, for making it impossible to deduce one thing by another if one part of the script, login or code for example, became somehow known to someone. To make the code yet stronger I used several encoding symbols to represent each actual symbol even within the separate sets. I also had several reserve sets of encryption symbols which I actually never used but had ready just in case, to write down some data requiring extreme precaution in handling, so it wouldn't be possible to deduce the data having known all the logins, codes, names and telephone numbers from third party sources.

It seems a lot of stuff to remember but later I'll show you the methods of systematization of symbols used, so it becomes not hard at all. Still, as it requires thousands of symbols to remember, even though the forgotten symbols are possible to derive by your own systematization methods, it becomes impractical for everyday use.

After forgetting the meaning of just one symbol, a number represented by given symbol for instance (remember, for security each number was substituted by several symbols even within separate set), it may take you a few hours or even days of work to recover the meaning of a symbol from the rest of encrypted data. It's because to find out the value of the forgotten number you first need to find in the encrypted text the other symbols belonging to the system of that set and then derive the value of the forgotten symbol. For safety different sets are never in similar systems in an obviously comparable way -- every set must have its own unique system to memorize/derive symbols within. Thus, basically I didn't use the method anymore after seeing how much time it takes to decrypt the data if some of the symbols got forgotten (in case of codes and phone numbers). The other reason for not using the method anymore was, of course, that I compromised the code myself by keeping both of the data sets – encrypted notebook and decrypted notebook – in my computer, while working on it, scanning the notebook into secure drives. Regardless the computer encryption I used was extremely strong there's never (and will never be) full security when it comes to computers. To create a new system as complex as I had created was not an option, I had to come up with a better solution.

While trying to find the solution I created several more codes for practicing the script while the encryption symbols basically remained the same – I only changed the meaning of the symbols and made some of the symbols more advanced giving them a kind of 'flexibility' in representation. By 2007-2008 the possible total number of cryptic symbols to use, in my way of writing, was in hundreds of thousands and even millions while used in context (the 'flexibility' increased ways of writing greatly - later I'll show you how), so I only needed to pick for using the ones I liked by design for particular meanings, the rest of them could have been used as an information noise at any time. After the extensive memory training and brain exercise in Japan the following versions of encoding I was able to create from zero to completion in just a week time, and the simplest (but still somewhat safe) codes in matter of just a few hours. I can do it at any time again – it's not a complicated process if you know what to do. But I wasn't happy enough. Each process of creation required writing down on paper large quantities of symbols and destroying the paper after memorizing all of the symbols.. and repeating the symbols regularly. By around 2009-2010 I was not always writing down the symbols anymore but quite often representing each symbol with a dot on a paper in a certain system, so I knew what the dot should mean. During the recalling process of the symbols I painted a matrix (out of actual dimensions when I was in a public place, not to give away the real size of the matrix of symbols I used) and marked dots to the squares for the corresponding symbols which I remembered (and of course to the extra spaces in the out of dimensions matrix). If I couldn't remember a symbol after a while then I created a new one instead of it until I could remember all of them.

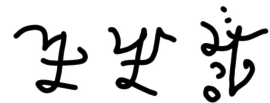
By 2011, as it was obvious that the simpler encoding methods were not becoming secure enough, or should need to include large amount of information noise which would make the code bulky and impractical, I began a completely new process of creation. I started with the matrix of actual symbols I would need to use but left the writing part completely out until I could come up with the method of writing satisfying all the requirements, including:

- simple to write and to remember the code symbols;
- easy to adapt for individual needs, to personalize;
- very secure against attacks by cryptanalysis.

While creating the new matrix of symbols I also choose a completely new approach – I decided to never-ever write down anywhere the matrix of symbols for the newest code, completely doing everything in my mind only. It took me two years to complete the process. First of all I wasn't in hurry, waiting for the idea to come into my mind how to write simply and while keeping the simplicity also in a secure way, so I began with little steps.. I decided about the dimensions of the matrix, then placed letters of the languages I speak into the matrix, then numbers, then mathematical symbols I use or may need to use one day, then punctuation symbols, then other common representations of ideas (symbols like arrows, squares, etc.), then rules for highlighting parts of the script, then several grammatical rules for flexibility of the code, then several encryption rules to modify the code, to correct errors and to switch between different encoding patterns during encryption. When that was done I added some common words and some other useful stuff to represent an idea or a process with a single encryption symbol, and after each step I took a break. If after a while, sometimes after several months, I was able to recall the symbols and was still happy with the position of them in the matrix, then I followed with other symbols.. if not, then I repositioned the stuff in mind and gave it a break again.

By now my own matrix of symbols is complete, and encoding rules are complete including all the commands/orders, modifiers and switches, never being written down anywhere, with several sets of symbols in the matrix for encoding information in different levels of security. And as now the rules are complete it obviously means I have also found a solution for the writing system which differs from the previously used methods, easier to memorize and to use. The beauty of it is in the possibility to write different things in different methods of encryption the way that they're almost indistinguishable from each other. Embedding encoded 'switches' in the script it is possible to write as a normal text with cryptic symbols, including information noise for security, as a highly sophisticated encryption, and also as a mathematically encoded text giving maximum security. All of the encryption methods can be used in the same script moving smoothly from one method to another at any point of the script, without visible indication of which method is used.

In this book, in one of the last chapters, you will find out what the latest, the best and the highest level encryption method is, which I'm using myself, but you need some patience going through the examples of basic level and medium level codes, then you can understand what the highest level encryption method will be if you wish to create something like that for yourself as well.



3 Why handwriting?

The reason to keep developing the methods of handwriting cryptography even after the beginning of computer security age is the impossibility to rely on computer encryption only. Whatever a company may claim about impossibility of cracking their code the only secure method is the one having no third party involved (or if shared between several people then no unwanted parties). You can only rely on the method of encryption you yourself have created or customized an existing one. My goal was to develop a method which everyone can modify, to create their own encryption systems while knowing how to do it. I think I have succeeded in the goal, surpassing my own expectations on the security of the method, and you're going to see it for yourself.

On the market there are available encryption tools for hiding messages in a picture for example, so someone may even not know where to look for a message to begin decryption with. This method is called steganography, which is the art of hiding messages in plain sight, so to say a 'branch' of cryptography. Still, as there are used certain algorithms of encryption created by unknown to you people you can never be sure that these pictures won't carry a specific signature allowing another algorithm to find the pictures where a message is hidden. With my method of encryption, after you customize the code and add some original tools, you can paint a picture yourself including in it a hidden message, while no-one can ever guess is it a hidden message, an art, or you simply tried your pen.



The handwriting encryption also provides you with greater stability for recovery of the messages compared to the ever-changing computer systems and their encryption methods. I have 'lost' lots of data, including collected material for one of my books I was planning to write: hundreds of pages of philosophical thoughts from great thinkers (including myself of course, obviously). The collected ideas are still there somewhere on an encrypted backup disc, but to recover the information I must buy an old computer (with the same system which was on stolen computer), to install the same encryption program (which doesn't work on new systems) and so on. It would have been easier to recover the data encrypted in handwriting.

A stylized signature or logo in Chinese characters, written in a bold, expressive cursive style.

4 Relevance.

I could simply give the principles of the latest encryption method, to explain the basics and the ways to personalize the code, but in that case you may not be aware of the precautions of use. It could be possible to explain the method in an hour to someone used to intellectual exercises, to regular challenges of the brain and to applying logic with common sense in daily life. But, there may be important points which are better to explain in detail right in the beginning, not letting it to wait till someone gets the point later, maybe even too late.

You can make use of the encryption method with hieroglyphs only if you know the basic principles of most classic encryption methods (also the weaker ones), the possible attacks on the codes and so on. You can find general information about the attacks on codes on the internet, by reading the history of encryption and how the historic codes have been broken. I'm going to explain the principles of my methods of encryption by showing you the way I've gone through and you'll see why I have discarded some previous methods as not the best. I consider it important because if I'll show you only the latest and most advanced method you may come across the idea to simplify it. So, by showing you the simpler methods you will see why some of these may be not a good idea to use for writing down codes or other important data in every occasion. At the same time, it is also good to know them because these simpler methods are still handy for writing down some less critical data, like personal diary or something, which you wouldn't want everyone to understand by accidentally leaving it around. Also, the simpler methods are a good training for your brain, to get used to the kind of hieroglyphs involved in the encryption.

For my surprise, while making overall research on cryptography in 2011, trying to find on the internet any news about advanced handwriting codes, I didn't find any. They were still talking there about centuries old methods, outdated and cracked long time ago, and about very complex ones (or very inconvenient ones) used by the military during world wars and during the cold war era. And lately all the talk has concentrated around the internet security, while the handwriting cryptography has obviously been forgotten. Well, everyone today knows that internet will never be secure in terms of protecting your private information, even if it's technically possible to secure, thus creating your very own code is more relevant now than ever before.



5 The method.

The handwriting cryptography method which I've created is one of those methods of encryption which is not entirely affected by just a few errors, and this makes it particularly hard to attack by cryptanalysis because you can leave the errors in, yet being capable to read the text later. For encrypting the data which wouldn't support any errors (for example codes, telephone numbers) there's possible to write it down several times without repeating itself, in different ways, thus not repeating the encryption error. I have also developed a method for encrypting a single symbol in a manner that it verifies itself against an error of encryption, but this method requires more complex data processing in mind which is not the optimal solution when you want to write up some data fast. It's better to write up something twice in different ways – if one gets wrong then hopefully another one will be right.. in any case it's easier to recall a code when most of the symbols can be recovered, against the situation when it has completely escaped you what was the login, password or a name of your private business contact.

In fact, believe it or not (later I'll show it), with my latest method of encryption is possible to encode a single letter, number or any other symbol in hundreds of ways with the help of a simple set of extensions to the script, making it the most secure handwriting cryptography method I have created, and yet it is more practical to use than my previous methods as it requires less information to remember. It opened the possibility to learn the method by those who for using securely previous methods would have needed special memory training for remembering all of those hundreds and even thousands of different symbols. Now you can be satisfied with just a handful of rules without the need to memorize too many different 'hieroglyphs'. So far I haven't seen any other handwriting code which comes even close to my method by its flexibility in design, simplicity in use and the level of security it provides, all in one package.

Basically, while a single symbol can be written in hundreds of ways then a two letter word can be written in tens of thousands of ways, a three letter word can be written in millions of ways and so on.. can it get any better than that? And you can do the same in your own way, so neither I nor a professional cryptanalyst with the help of an array of most powerful computers can crack your code if you use all the necessary rules and precautions properly. When you customize existing encryption tools and for addition create your own tricks embedded and hidden in your script then no human can ever guess and input all the possible tricks into computer programs for cracking your code.. it all comes down to your imagination in creating your script.

I'm quite confident that a handwriting encryption with your own symbols and rules can give you way much better security than any computer encryption will ever be able to provide, when it comes to encrypting symbols of course – I have yet to come up with an idea how to encrypt pictures in handwriting and I'm not taking on the task any time soon I guess. Computer has a fixed set of symbols and rigid rules of encryption while you can create unlimited set of symbols and rules – only you will know which symbols or parts of a symbol carry information and which are just a meaningless noise.. no computer will ever be able to go through all the possible combinations of human imagination to crack a well created handwriting code.

So, let's get started with studying the advanced handwriting encryption methods which I'm promoting as the best in the world -- so far anyway -- at the time of working on this book there was no competition of the same level. You can do your learning the way you like, but I will describe you here the way it should work the best for you, in my opinion.

It's better not to read the book like a mathematical manual in clearly described steps. The book isn't written that way and thus it is not supposed to be consumed intellectually as a "how to do" manual in classical terms. Because this cryptography method isn't simply to copy and to use.. you must get the big picture of the possibilities of the encryption technique and then create your own code with your own cryptic symbols and your own other keys of encryption (the matrix of real symbols and the rules of access of the real symbols). Thus, take the book as a book of art.. read it, look it, get an idea, and then read it again and again, each time with a better insight to the method and better set of tools in your mind, picking up things you missed before.

5.1 Getting an idea.

Before continuing with the next chapter, on the basic principles of handwriting cryptography, take a tour through the book looking carefully at all the pictures of the cryptic symbols you'll find -- all the hieroglyphs and all the elements of the glyphs. Don't try to understand them -- you will not. Just look like an artwork paying attention to details and see if you can find similar details in several places of the 'art', a kind of characteristic signature of the artist. Try to make sense of the art as pictures, and that's it in the beginning -- you only need to get a general idea what's going on in these pictures, without guessing what's the message the artist is trying to convey.

Before going through all the pictures of the hieroglyphs don't read the book yet more than till the end of this chapter, 'the method'. From curiosity you can peek into the text of some chapters or sections deeper as well, it's not forbidden of course, but you'll be better off if you will continue with the next chapter, 'basic principles of handwriting encryption', later. The point here is that you will get much better understanding of descriptions, explanations and the terminology used in the following chapters when you have in your mind an overview of the many ways of writing the glyphs.

5.2 Seeking the information.

After you have gone through the book studying the pictures of the hieroglyphs, read the basic principles of handwriting encryption and all the following chapters. Read the book thoroughly till the end, but don't study everything in detail yet. Learn the general message in the book, to know in which parts of the book you can get an additional information and explanations when needed, when you'll be studying the method properly on the second read.

In the first read of the book you don't need to understand everything. No need to go back and read again if something isn't quite clear. But you do need to go through whole

the book, the parts which deal with the cryptography method, to have a picture of different sections of the book which deal with specific elements of the method.

It is way easier to learn the method later when you have an overview what is where to be found. Some chapters of the book are easier to understand when you have already been through later chapters, with the technique of encoding already understood in general, but without all the details yet. Of course you should try to understand everything you can, even from the first read, but don't enforce on yourself the condition of not going to the next chapter before all is clear, because some details may just be missing yet for the full comprehension. Or to put it differently -- even if you get everything from the first time you will definitely be short of understanding of all the context in details and possibilities of interaction within the cipher text.

5.3 Learning all the details.

When you're familiar with the layout of the book, gone through all of the chapters and ready to begin a proper study, take it with small steps, few chapters at time but not whole the book. Why? Because in a quick study the new information won't settle in your brain, the necessary connections in your brain cells won't become strong enough (there will have formed not sufficient amount of supporting connections between your brain cells for the learned information), thus if you learn fast new and again new information, although you can get a quick understanding you'll forget half of the information the next day. In cryptography, forgetting half of the things is a big deal. Even one thing forgotten may ruin your life's work if that one thing happens to be of a great importance for the security of the method.

Read few smaller chapters at time, and bigger chapters by smaller parts at time. Leave the book for a day or few, and then go through the last studied chapters again, to make sure you haven't missed anything. You must understand all the important parts of the method by heart, by logic. If you don't, then seek supportive information in other parts of the book, from previous or later chapters, and then read the chapter again till you get it entirely. And so on, little by little, whole the book, even if it takes several months or a year to study. It's better to spend a year on learning the method among other things you do in your life, than learning the encryption method fast missing some important information that the book is trying to teach you. That's one reason why some elements of the method are repeating throughout the book -- it's because they're important and you should not miss them.

There's also a possibility that I will have not explained something in a necessary detail, because for me all is clear but I'm not inside the mind of a reader. With all the great effort to end up with the best explanation I'm also aware of the possibility of creating a boring read if I should take the readers as not sufficiently intelligent. Thus, in the compromise of creating the book compact and interesting while explaining everything necessary, I have left some elements for your own imagination. In certain parts I point it clearly out -- your own creative work is needed for the safety of the code. Thus, there is a possibility that you will read a chapter over and over again but won't get the point. Although sad then, but in this case just skip the part and read on, relying on your own creativity and logic in the future, to close the gap on the missed part of the book. If in the

end something still wasn't clear then most likely it wasn't that important, because all the important parts of the encryption technique are explained in great detail in each level, and repeated throughout the book when necessary in the context of specific sections.

5.4 Working on your own.

After your study of the encryption method is properly done and you feel that you're ready to create your own code, you should try first with simpler methods of encoding. It is obviously depending on the level of understanding you have achieved. I am quite sure that not everyone who reads the book even several times, and thinks that all is clear, is ready to embark in the more complex levels of encoding. The more complex methods require simultaneous interplay of many details, none of which can be created wrong. And you should not put everything down on paper for the safety of the code, which I'm explaining in several parts of the book. That ability to have in mind without any errors the whole picture of the key to the code, even if it is your own code, comes with practice. It's like with learning to play a musical instrument -- little by little your attention to detail during musical handling of an instrument (playing) becomes automatic and you can take up ever more sophisticated tricks, up to the point when it all becomes so easy that you can create a piece of music having it all in your mind completed before even touching the instrument. Don't try to become a 'maestro' of handwriting cryptography from a first try -- take it one step at the time, beginning with simpler codes first.

After you have successfully created a simple code (with the tools and ways of writing provided in this book, if you like), verified it against errors through practical trials, then you can begin making a layout of your very own original method, a complex one at this point, but still you better practice with the simpler methods as well, created with your own original tools and style of writing. Your perfect complex method will take time to mature, thus playing around with simpler methods provide you with some fun, give you new ideas for including in your advanced method, and also will help you at times in making notes in a secure way before your advanced method is complete.

Most importantly, creating a simple code takes less time which is a way to achieve satisfaction from your creative work, to get the feeling that something has been done and is useful. It will be very frustrating to end up with no results if you get right on to the most complex version of encryption and won't succeed to complete it for a long time. Trust me, you'll be better off creating simpler codes first even if later you discard them from use.

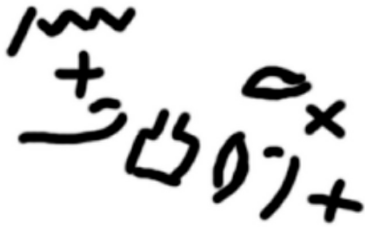
Your handwriting code is complete when you're ready to use it for storing data without worries about it being possible to crack nor about losing the data yourself due to confusion in the elements, all the keys of encryption in your mind and no traces of the keys left behind.

In the end of the book (chapter 28) I give you some selected samples of writing in different alphabets, hieroglyphs and shorthand (from Wikipedia and other public sources), used in the past and in use at the present time, for visual help in creation of your own style of cryptic writing. It is very important to detach you from my way of cryptographic writing, to give you some different ideas how writing can be done. But that you will need

in the end -- before that you shall learn how it's done in my way of writing.

I will not be surprised if you will find my way of writing of the glyphs the most convenient one, because I've been through many different trials with other types of glyphs as well, in the first years of my search, until I found that the current way, presented in the book, is the only way to do it. This way provides security, flexibility, clarity, precision and the result will look beautiful as well.

Here's an example how you would do without an assistance..



..actually that's a specimen of Proto-Sinaitic script, one of the earliest (if not the very first) phonemic scripts. (No copyright on this picture).

Anyway, as you have read the book this far by now, before to continue reading on, with the next chapter waiting, now is the time to pause reading and to get familiar with all the pictures of the cryptic writing in this book. Have fun.



6 Basic principles of handwriting encryption.

The following principles or 'rules' are not necessarily in the order of priority – all of them are important. I'm numbering the rules just for easy following and in the order they come to my mind.

Rule 1:

Never keep plaintext and encrypted text together. Never show off anyone how beautiful is the encryption method representing your name or else. If it's unavoidable for whatever social reasons then show it generally writing down a wrong thing in a wrong way.. and if asked much later what does it means then simply say you probably made a mistake and can't understand.

I can imagine that after creating a personal code the excitement of the achievement may trigger the emotions to show it someone, but it may cost you lost privacy and/or extra work to create a new code. You may only show the exact encrypted writing to someone you're planning to share the particular code with.

Rule 2:

You should always encrypt as little text as reasonably possible, encrypting only the parts which need to be encrypted, the related parts of text which would give out the information about context of the encrypted text, possible sources of the encrypted text and other content having references to the encrypted text.

The less data you encrypt and fewer references you leave about the encrypted text the less vulnerable it will be for attacks by cryptanalysis. When encrypting whole the context is impractical then don't insert the encrypted part directly into plaintext – instead insert an encrypted reference to the place where you hold the actual encrypted part belonging to the document.

Rule 3:

Never write a script without including at least a minimum amount of information noise – the symbols and extensions which actually mean nothing. Write into a cipher text about 1/5...1/10...1/15 of symbols as a noise and varying the amount of the noise from place to place.

Using the noise level and quality of it properly makes exponentially harder to crack the code by cryptanalysis while still keeping the script easy to read without affecting the process of decryption.

Rule 4:

Never write down text, specific names and codes in the same script using the same encryption method/pattern or set of symbols.

Having an accidental access to one of them will make it easier to crack the other part of the script if encoded with the same method and/or set of symbols. It doesn't mean you will always need to write down different data with completely different encryption methods, you need at least to use different keys of encryption within the same method.

Rule 5:

Create a simplified “public” set of symbols to write down stuff temporarily but fast and still quite securely while others are watching and may guess the content.

Later you can rewrite the data in another place with “private” and highly secure code while destroying the “public” version of the same script.

Rule 6:

Never correct encryption errors by writing a correct symbol above the wrong one. Include in your encoding method hidden (also encrypted) tools for backward deletion, insertion, replacement and other correction symbols, and special signs indicating errors if placed in the right position but otherwise used as noise. If an error isn't important leave the error uncorrected. If not possible to leave an error in and to correct it then write the data again correctly with other symbols leaving the error untouched. If none of the above works for you then destroy the wrong symbols leaving no traces of them before writing correct versions above.. or destroy whole the script and write again.

Indicating your errors is a good help for cryptanalysts, leaving the errors into the script is the cryptanalysts' nightmare.

Rule 7:

Never use classical grammatical signs within encrypted script. All the signs like periods, commas, apostrophes, dashes and any other signs, also real positions of spaces, line breaks and other grammatical rules must be encrypted and hidden within encryption. An encryption must give no hint whatsoever what it is about. Don't even use any classical signs in wrong places as a noise – use only random false spacing for breaking and combining 'words' into the blocks with not the real length.

Placing classical signs within encrypted script is an excellent help for cryptanalysts to guess the content. In some cases you may use the classical signs as a noise (in wrong places), to confuse others, but if you're not careful enough you may give away some information unwillingly just because of your habits (dropping accidentally the signs into right locations).

Rule 8:

Never highlight any part of the encrypted script in classical style for easy following. For highlighting use encryption rules you could easily spot if necessary but which wouldn't look special to others.

Highlighting parts of the encrypted text, like underlined or bold script, is an excellent help for cryptanalysts to guess the content. If you do need to include highlights for the text you're encrypting then you must create corresponding, visually hidden rules and code symbols for use within the encryption, including code representations for colors, size, font, background, etc.

Rule 9:

Never use within encryption the actual length of encrypted words and even sentences to make it look better and easier to read.

Writing encrypted script with the actual length of encrypted words is the best help

for cryptanalysts to guess the content and will greatly diminish the strength of encryption. In some cases, writing long pieces of text, it will render the use of almost all the other precaution rules practically pointless -- by the frequency of words and the rules of grammar in a language a computer program will easily be able to make sense of your encrypted text.

Rule 10:

Never back up any part of your personal encryption method in a computer during or after creation of it – all keys must be destroyed. You must rely on help of paper notes during development of your method and on your memory alone. You should never leave around any traces of the keys and the systematization used in your encoding method.

That's an obvious one you may guess, but there are still people out there thinking they can trust their computers – do not, that I can assure you. If you do need to write something down for visualization, before you learn to do it in your mind, then you must destroy the paper each time you stop working on the code. To work on the code the next time you will write it all down from memory and will continue (that's an excellent training by the way). You can storage ready scripts (cipher texts) in computer with no worries, but never together with the plaintext related to the script as it may significantly weaken your code making it vulnerable for further attacks.

Rule 11:

Create your cryptic symbols different from any known to you symbols (not borrowed from any language), with several layers where the separate parts are difficult to distinguish and not easily understood by others, so that they're not possible to input into a computer program automatically.

Your cryptic symbols must be complex enough requiring human work on each symbol trying to take it apart. If you create single layer symbols, with each corresponding to one actual symbol, then a scanned document can easily be translated into a computer language and a program can figure out the actual symbols by the frequency analysis.. and to crack the code in no time.

Rule 12:

Always test your every newly created encryption method with a reasonable amount of cipher text, storing some complex, difficult to remember data which you can verify later. After you have forgotten the complex data sets, successfully deciphered the code and verified against errors, only then you can begin encrypting important data which you can't afford to lose.

The length of the cipher text for trial depends on the complexity of the code -- it should include all of the essential elements of your code several times, to make sure you won't be confused yourself while deciphering the test data. The plaintext used for testing must not be kept anywhere together with cipher text or separately -- the plaintext for testing must be a part of a larger volume of text/data but the location of the part used for testing must be only for you to know, never highlighted within the volume.

Rule 13:

Read again the first rule of the basic principles of handwriting encryption. If

you're a very social person and like to share things then for social needs you can create a code for fun which you never use for encrypting important data. Your own personal code shall remain the murkiest mystery for others.

Cryptography is fun, and fun is always better when it is shared with others, thus creating a simple code just for fun is a reasonable thing to do.

In addition to these, certain rules and principles are pointed out in other parts of the book when they fit into the context and are more suitable for that particular section than using separately from the context. Many principles are the precaution rules of the encryption itself used in conjunction with the encryption process. Where suitable these rules are highlighted.



7 Techniques of encryption.

The method of encryption I'm presenting here contains some classical techniques of code creation like confusion, diffusion, noise and others, which may be not original in principle, but like all languages have some common principles of syntax these are just principles – the method lies in how to use them. The main originality and security of the method is in the way of using the principles of encoding in combination, and particularly in the method of writing which gives enormous flexibility in customization. After that it all depends what you like to include while you're modifying the method for your needs creating your own code – either you like to make it more secure, or with less security but more easy and fast to write and read. One technique is obviously unavoidable – confusion (substitution of common symbols with different ones) – this cannot be left out when it comes to handwriting encryption as it is the strongest power providing the most significant part of security. Still you shouldn't rely on it entirely – your habits of writing may give you out.

I'm using the word 'symbol' quite often and in different meanings. For understanding it correctly in the context I make a little distinction here: there are symbols which are common, and symbols of encryption. For clarification I will sometimes refer to common symbols as text symbols (TS), and original symbols (created by you or someone else for the encryption) as code symbols (CS). The TS will mean all the common symbols in use which a computer can generate, be it letters, numbers or other signs, while CS will mean all the handwriting symbols used for encryption and not having a standard computer alternative. The word 'script' I'm using in the meaning of 'cipher text' – hopefully it's obvious already.

7.1 Confusion / substitution.

Substitution is when you replace symbols (TS) with other symbols (TS or CS) either in a systematic way or randomly, having the 'key' (or several keys) memorized.. or written up somewhere if it needs to be given to other parties to decrypt messages.

Using the same symbols during substitution (the TS-TS method) simply reordering them in the alphabet provides almost no security. It's pointless to create a handwriting system using classical letters, numbers and other common symbols as then it only needs to be scanned into a computer and let a program to work out the system you use. When you create your own symbols, if not too simple of course, it needs to be first understood where is the information in these symbols, which part of them could be translated into the symbols that a computer can read. This requires human work and makes an attack on the script enormously more difficult.. each symbol (CS) must be analyzed by a human while there's never certainty that it is correctly understood, or isn't an error of encryption, or even worse – a noise. The more noise a cryptanalyst reads out from a script and inputs into a program as a symbol for cracking the code the further away he/she goes from solving the code.. at the same time an analyst cannot exclude any of the "maybe just noise"

in your script as it may contain information – excluding even a little piece which is real part of an information block makes it harder again to crack the code as it may change the whole meaning of the entire block. Thus confusion (substitution of classic symbols with original complex symbols) is the most important part of the method, it must be given particular attention while learning all the tricks, and creating new ones.

In this book I'm going to show how to confuse others properly yet being capable easily to write and read the script. In order to do that I must show the writing technique of the hieroglyphs from the very beginning – when you learn to see an order in a seeming mess then you'll be able to reorder the system for confusing others. As you need to use the symbols of the language(s) which you encode while confusing others you need to substitute each symbol (TS) with a different one (CS) which have no meaning to others (cannot be pronounced or easily described). To make the confusion stronger you must create several different symbols to represent each actual symbol, the more the better. It all depends on your imagination and on the capacity to memorize. For creating more substitution symbols and not to forget them you also need to systematize your own symbols a way that this system wouldn't be obvious to others.

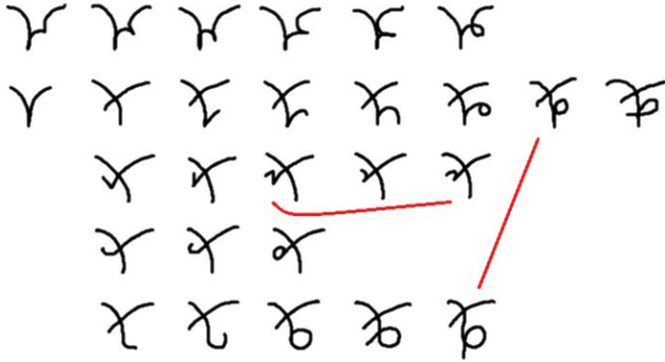
Note: I'm talking here (so far) about a simple method, even not yet the most basic version of the advanced handwriting cryptography. It's just an introduction, a method where code symbols are fixed to the common symbols. It can only be made somewhat secure if you create several substitution symbols for each actual symbol and add many noise symbols. The positive side of it is that after some practice you'll be able to write in the code almost as fast as normal writing (in this method you don't need to 'calculate' each glyph).. but, writing fast implies that you write in the code a lot and this makes the code vulnerable for attacks by cryptanalysis as there will be well over critical mass of cipher text available.

As mentioned in the basic principles of handwriting encryption, it is a good idea to have a code for social interaction, because you never know -- one day someone may see your encrypted notes and be curious. So instead of telling that this is a secret, basically telling someone off which isn't pleasant in every relation, you can begin to explain enthusiastically how good the thing is having a simple public code for demonstration in such occasions. That substitution code is just that, because it is so easy to create. Sure that in most cases the person whom you begin to explain an encryption will lose interest in the subject right after few sentences and will change the subject, but if not, then you have the reason to be cautious with that person. Simply offer this book to read then, to get all the details of the method.

There are literally thousands of ways to create cryptic symbols in a clear system, where every little added modification can mean yet another real symbol (TS), thus you can create many repeating sets of alphabets, numbers and punctuation marks. The more sets of CS for replacing TS you create the more diverse the cipher text will look. But if they will be in a system which is easy to memorize the code will also provide no great security.

You can read again in the beginning of the book a bit of history (chapter 2) to refresh your mind about the problems arising from using simple substitution methods. I went to such great lengths to develop a secure substitution method because it is easy to write in substitution code if your memory is great enough. In the end I had to discard my attempts of developing a safe substitution code.

Here are some examples of the hieroglyphs you can create for replacing actual symbols (TS) and the way of systematization so that it wouldn't be needed to remember each one of the glyphs separately. Just few of them:



As you can see some of the hieroglyphs in the system may look quite similar (connected with the red line). If you write the glyphs fast then the details may become less distinct and be confusing. When you create your glyphs you must decide either you prefer to pay attention to details and have more secure code or leave out all the confusing options being able to write your code with an ease, after sufficient practice.

In the above example the systematization may look quite obvious to anyone but if you apply the system of glyphs to the system of symbols in your matrix then the differences in these systems will give seemingly random output of the pattern. Thus above symbols won't read like A-B-C-D-.. if you won't deliberately equalize the two systems – of course you won't do it if you're already smart enough to bother yourself with creating your own code.

7.1a Information carriers.

To confuse others properly it is not enough to create just all kind of glyphs but the glyphs themselves must be made flexible. You may use the same glyphs (base glyphs) over and over again modifying the meaning of them through info carrier points, but for others they will all look like different glyphs. The information carriers (modifiers) are attached in a seamless way that it wouldn't give out the base glyph. Only you will know which characters are modifiers and which constitute an integral part of another base glyph.

If you remember, in the history chapter I told you about the thousands of hieroglyphs I had to memorize before I came up with the latest method.. previously I used a large amount of complex hieroglyphs relying on the systematization of them. Most modifiers were attached in a clearly distinguishable way or separately. Then I discovered that using a small amount of base glyphs with a large amount of modifiers, attached seamlessly, is way more practical, because the modifiers are much easier to systematize while providing greater flexibility in writing the script. This method improves overall security of the script as well. The only down side of the method is that you need to 'calculate' the glyphs before writing them – if you write a base glyph first and attach the modifiers later then it will be obvious to others. You must have the full picture of the

whole glyph in your mind before beginning to write it, which slows down the speed of encryption.. but that's the price you pay for having a highly secure handwriting encryption method. In any case, after some training it will be not so hard to do. It will also be flexing your brain a bit, which in the end is only good for you. With this method you cannot fall into habits by repeating the glyphs but you must be continuously creating new glyphs on the go, which means that the exercise of your brain will also be continuous.

With the technique of simplified glyphs, while attaching to them modifiers, you will choose a small amount of base glyphs (10, 20, 50, .. you choose) and decide the characters used as modifiers for these glyphs, which in combination will be representing substituted symbols or sets of them. While getting used to them, with the help of switches, you can gradually add into your code (into your symbols' vocabulary) more and more complex glyphs with new modifiers (representing substituted words for example, while modifiers will specify them grammatically depending on context). Many of the glyphs may look very similar to others, as if written a bit differently only by a mistake, but for you they will carry different information, a completely different meaning, because only you will know where to look for differences in values within a glyph and where the differences aren't changing the meaning.

Here's an example of some of the info carrier points you can include in a hieroglyph. This is not a fixed rule – hieroglyphs may contain different points of meaningful data. I will call them fixed points here, for this example, as in actuality this may constitute a single hieroglyph:



While one point can have tens of different characters of representation of a meaning, then possible amount of the hieroglyphs will be calculated in the power of the points used. Let say you decide to use only 10 different characters for each point (the rest of the possible characters can be used as noise – amount depends on your imagination), then in the picture above the first block (on the left) will have 10^{12} possible combinations.. it makes 1'000'000'000'000 different hieroglyphs which can carry a meaning and countless other possible glyphs that can be used as noise. The right side block on the same picture can in this case have 1'000 different meanings and while used as an additional modifier to the previous hieroglyph you will have a thousand ways to change the meaning of already written glyph be it a symbol, a set of symbols, a word, a process (e.g. to separate/join the blocks), a function (e.g. capital letters), an idea (e.g. highlight), a switch (e.g. numbers/letters/punctuation), a correction (e.g. insert backwards, delete or replace), a pattern addition (the reading order of the glyphs), another order (e.g. the reading order of characters within the glyph), a command (e.g. to memorize some part of the script for later use), or just a noise predefined by matrix or by glyphs – only you will know which of them.

Keep in mind that this was a calculation for a single base glyph with modifiers (the

second 'slim glyph' counted as a modifier). With another base glyph you will have another million or billion meanings to hide, depending on how many info carrier points you use. Thus a two glyph combination can have let say modestly a billion multiplied by million possible meanings (with each additional glyph only multiplying the amount of possible meanings of the whole set) without affecting your ability to write and read the script because you know the rules you created... but definitely it will create head aches in a cryptanalysis department if they dare to try to attack your script.

You add to it all the external noise (extensions to hieroglyphs which carry no data, used just for design) and I suppose even a mathematician won't be able to calculate what it takes to crack a code using the confusion/substitution method only, without even applying other techniques. It's impossible to precisely calculate the amount of true possibilities because the glyphs can change their meaning depending on context (previously encoded information), thus exactly the same looking glyphs can have different meanings in different places. There's no limit to human imagination -- not sure anyone can quantify it, putting the possibilities into exact numbers.

Sounds like too much for your brain? Don't get disappointed yet, it's only difficult to attack the code, to try to crack it, but for you it won't be difficult at all if you don't mess up something yourself.. you will only need to use a few hundred symbols (TS - letters, numbers etc.) positioned in a matrix and to create several rules of access using in the matrix always the same symbols. In different representations of the encoded symbols (in different hieroglyphs) you can hide the meaning using values as digits, numbers. If you like the mathematical way of substitution then all of the possible combinations will only be used for calculating (by your own rules) the position of an actual symbol (TS) in the matrix you hold in your mind. The matrix is static for as long as you use the same encryption method, but you can use the same matrix for other methods as well - it only needs to be created/learned once and used as long as not compromised.

Basically, using the same matrix of symbols (TS) you can represent a single symbol from it in hundreds and hundreds of encrypted ways, making it impossible to derive a meaning from a script even containing millions of words, if all the precautions and rules are used properly of course.

I must give here an important warning. All the imagination in creating different glyphs and a complex method of encoding will be useless if you get lazy and will be repeating in the same script parts of the code regularly in habitual manner, simply copying previously encrypted words. During an encryption you must verify your script against previously used ways of representation, not just copy a word but to use another combination of symbols. Each time the same word must be encoded in a different way, not just repositioning the noise in it. If you use the same word with different noise attached then it will be possible to deduce the actual symbols from them and eventually it will lead to deduction of all the parts of the script which carry no information. Subsequently it will be possible to convert the rest of the script into meaningful symbols and blocks of data for computers to analyze, maybe even successfully cracking the script (or some part of it) if the script is long enough for applying cryptanalysis.

For making the code even stronger you should add to the glyphs some additional features (separate signs) which either carry some information or are there just for

confusing. It doesn't really matter that theoretically the method is already strong enough. You should add as many different techniques as reasonably possible, to confuse others ever more, because we are humans and can easily fall into habits, repeating ourselves over and over again, regardless we are not always aware of it. These repetitive errors can help a sophisticated computer algorithm to find out certain patterns, and maybe, while compared with your repetitive errors in your plain text writing, the code can be cracked in some parts. Thus the additional features of encryption attached to the glyphs will somewhat compensate for our weaknesses as human beings.

These additional signs like dots, lines, curves, circles, etc., whatever you like, may be added to different locations – above, below, left, right, inside, and in several locations at time. Depending on location, relative size and combination of several of those in a specific way they may change the whole meaning of the glyph or carry some other information like reordering the structure of the code or changing applied formula of encryption or a pattern of reading of symbols. If the visually easily spotted combination of the separate signs is present then you read the information hidden in the glyph differently, if not then you simply discard the noise, but for an analyst trying to crack the code each dot, line, curve, etc. must be entered into a computer program and each addition makes it harder to crack the code.

Simple words written in a complicated way – glyphs with additional signs, a bit designed in this case to look symmetric, may look like this:



To appreciate the work done in developing the method and to understand the necessity to invest your time in doing the same while developing your own code, I must repeat that using only substitution of symbols is not enough for a strong encryption simply because you will never use all the millions of ways of encoding – after a while you'll develop a habit of using only certain amount of combinations and will be repeating them in quite a same pattern whether you like it or not and whether you know it or not – even if you think that you encode each time differently (will try hard) you will still be a slave of your hidden habits. The substitution won't necessarily be visible but a good computer algorithm for cryptanalysis will guess the words (by length) regardless being incapacitated in understanding how the separate symbols are encoded – thus you need to add a diffusion pattern to the simple substitution (on top of separate additional features of the glyphs mentioned above).

Theoretically it's possible to use the substitution method only and still have a strong code. But then you need to use a large amount of noise which means you must create a large amount of hieroglyphs which mean nothing and insert them into the script almost as often as the symbols which carry information (I'm talking here about the simpler encoding method I have developed, with fixed meaning of glyphs, although with many replacements for each actual symbol). The longer the script the more noise you should use compared to the amount of information, so you must in the beginning of encryption decide the length of the script and the approximate amount of noise in it. This

makes the method inconvenient and the script becomes uselessly large while hiding little information in a large volume. It's only suitable for writing up temporary information in places where you can easily destroy the code carrier later. Adding diffusion technique makes the script much more compact for the same level of security the script provides.

7.1b Noise and distraction elements.

Noise is a very important part, so important that it will repeat in different context throughout the book. In principle you don't need to have a very complex code if you use loads of noise in the cipher text, but in that case it will be more like steganography. A huge amount of nothing valuable within a code is simply impractical -- imagine having to write instead of one page of real message some ten/twenty pages where the only one page of message is hidden.

Another story is with distraction elements. These are so called sub-elements which aren't entire meaningless hieroglyphs but only additional (usually smaller) characters within the cipher text which are there for confusing/distracting a cryptanalyst, not permitting to find out the set of cryptic symbols that carry values of encryption.

If there was no noise included in a cipher text then the mathematical probability to figure out a system of encryption would be much greater. The hieroglyphs provide good enough security even without noise, beginning with and above medium level encryption method, but still, with ever increasing computational power in the hands of analysts the inclusion of noise becomes a very important tool of encryption.

There's not only one way to insert noise. The noise elements can be:

- cryptic symbols specifying parts of hieroglyphs as not valid values of encryption, but which in other cases do have encryption value;
- cryptic symbols specifying whole hieroglyphs as not including valid values of encryption, which in other cases do carry information;
- cryptic symbols specifying parts of cipher text (entire blocks) as not valid, which by default are valid;
- cryptic symbols not used as values of encryption, thus automatically discarded as not valid encryption elements;
- context of elements of cipher text which determine specific parts of hieroglyphs as not valid encryption values (for example using correction tools from matrix);
- designated locations in the matrix that are never used for storing real symbols or commands but only for creating diverse noise within cipher text;
- context of information within the cipher text which determine specific parts of the code as not valid;
- context of the entire cipher text which determine it as a distraction.

The rules, symbols and locations of noise must be determined during creation of the encryption method with no less precision than the elements which have value of encryption, which doesn't mean the noise is very limited in the way it can be applied. The noise is only limited by the rules of encryption you create, thus the rules of applying noise can give an enormous freedom in designing your script not only for hiding information but also to make the whole thing look beautiful. In the end that's also the noise within the script that makes it incredibly flexible, if the different ways to encode values aren't

sufficient in some point of the script.

7.2 Diffusion / transposition.

Diffusion (also called transposition) is changing the order of symbols in a code. In fact the previous chapter already partly included diffusion, but as it was dealing with representation (substitution) of symbols, words or processes with a single hieroglyph it was still just the confusion technique by its definition.

Again, diffusion is particularly important tool while using the simpler encryption method, with fixed meanings of glyphs. When the base glyph method is used, with seamlessly attached modifiers, then diffusion isn't that important tool in the meaning of using it in many sophisticated ways. In latter case diffusion can be used much less, but is still an important additional tool for the safety of the encryption.

For the TS-TS method the key is a pattern of substitution and/or the pattern of diffusion of the symbols. For the TS-CS method the key of substitution will be the matrix of symbols (TS) with the way the code symbols (CS) are systematized and related to the matrix. In the TS-CS method the diffusion keys can be (re)ordering patterns in the matrix, (re)ordering patterns in the system of code symbols, and the (re)ordering patterns in the writing/reading of the symbols in the script. The many ways of encoding (applying different keys) in TS-CS method opens the possibility to use the same method, created once, between different parties, giving each of them their own keys, so other parties won't be able to read the messages not meant to them. You don't need to create a completely new writing method for each occasion – you only need to teach other parties how to use the method you have created, then give them their own keys and send them encrypted messages in letters/e-mails openly with no worries.

Diffusion tools are hidden messages (using specific glyphs or signs to give commands of application) about changing the order of the hieroglyphs, about replacing them or marking as errors, about highlighting them, repeating (not visually, with a hidden command), copying for later use and many other different messages (orders, commands). The specific commands can be applied to separate hieroglyphs, to encoded words, to sections and blocks of hieroglyphs and also to the whole script. Creation of the tools of diffusion is limited only by your imagination. The hieroglyphs for applying specific diffusion tools don't represent substituted symbols but carry commands for execution of specific tasks.

The methods of diffusion can be either fixed for whole the text (like a pattern of reading) or for some part of it, or be modified with the help of hidden switches during the encryption. The tools of diffusion are also the keys of encryption and must be created once and for ever during the creation of your method of encryption. The commands for application of particular diffusion methods can be placed into your matrix and also be hidden in specific base glyphs, modifiers, external symbols, or in any combination of them. You can leave into your matrix of symbols some unused spaces for any new symbols which you may need to add one day and for new diffusion methods' application commands.

If your method of encryption is well thought and calculated then you don't always need to use visible switches to apply specific diffusion tools – the commands can be

executed then there's a certain context in use of substitution. Omitting visible switches (when possible and reasonable) will make the script shorter for the same amount of encrypted data without compromising its security.

Never discard already created and used diffusion patterns. Never change the application commands nor context of those patterns. Never use the empty (reserve) spaces of your matrix as noise because later you may need the spaces for new commands to apply new patterns of diffusion, or even for the (new) symbols you decide to include. The noise locations within the matrix must be marked (in your mind) as such, for not confusing with vacant spots.

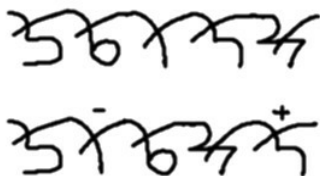
If you discard an old diffusion formula as too simple when you get used to it, replacing with a better one, you may one day be in trouble while trying to decrypt a message in your own old script. The old commands of diffusion patterns can easily be modified by simply adding a sign (dot, line) to a specific location in a glyph, so they can be reused as new diffusion commands.

With diffusion too, as with substitution, it is theoretically possible to create millions of ways to represent the same word be it just one letter – “I” – because the switches (commands) of the diffusion themselves can be encoded as hieroglyphs or attached to glyphs seamlessly in millions of ways and placed in any location inside the script – you can write the whole page repeating the same word “I” over and over again while representing visually random script. It's achieved not necessarily with a single glyph for each word, but with many combinations of different glyphs you can easily end up repeating the same word over and over again.

While creating your personalized diffusion patterns you must decide about the use of them in different situations – some simple patterns may be used for encrypting a text, for counting and for dates, time and so on, while other more complex ones for logins, codes, telephone numbers etc., for each of them a different diffusion pattern with the related switch. You may also use different systems in your matrix of symbols with the related switches to change the systems (placement of actual symbols within your matrix), or include extra sets of symbols in the matrix for different occasions without the need to use switches.

The method is in principle well secure without diffusion if you don't use the fixed substitution of symbols but the advanced method of ‘calculating’ the glyphs.. and will never repeat the same encryption of words. If you won't use diffusion you must be almost perfect in creating each time different hieroglyphs for the same set of symbols. It's safer to add another layer of security – diffusion – and to relax a bit in creation of the glyphs.

The diffusion patterns must be complex enough for providing sufficient security, yet not too complex that would make writing and reading the script too impractical for use.



Above there's an overly simplistic example of using switches for diffusion. Both the upper line and bottom line represent the same order of glyphs. Of course you won't be using +/- signs for such diffusion but the signs that won't give any hints about diffusion patterns you use.

7.3 Other techniques.

Writing method (visual representation) of the script remains basically the same for all the methods of hiding information in the script. Visually it will be impossible to determine which particular method you're using. Depending on your preference you can hide information in the script giving different values to different attached parts in it, while the rest is discarded as noise, and decide the technique of 'calculating' the actual symbols (TS) in the matrix.

7.3a Data points' method.

One way of encoding is to use for encryption only data points (info carrier points) within a glyph, without giving values to base glyphs and creating a set of digits to calculate position of a symbol in a matrix of symbols (TS). In each glyph the data points themselves may be written/read in different orders using some of the data points (or specific characters, cryptic symbols) as control switches between reading patterns, thus seemingly different glyphs may have the same meaning. For encoding different types of data (text, codes, etc.) in different ways the switches of reading patterns can also be used for switching between different matrixes. It doesn't require creating completely new ones but simply ordering the symbols you already use in the matrixes differently.

7.3b Mathematical method.

Mathematical encryption of particular data will not be practical for a handwriting code if used in high complexity, but applying a mathematical formula which is simple won't provide much more additional security than well applied diffusion which in principle is itself a simple formula. I'm not going to explain advanced mathematical methods (advanced in the meaning of processing in mind) as most of people will never use them, but in principle the method of writing of the script allows the inclusion of all kind of hidden rules for mathematical encryption. The problem is that it slows down the processing time during both encryption and decryption (it's a good exercise for brain though). Also, as you must do all the calculations in mind it is more prone to errors. The level of security provided just by confusion and diffusion in combination with noise is absolutely sufficient for everyday use.

7.3c Art method.

You can also convert your script into an art, so a cryptanalyst will never be sure if your picture contains a hidden message or it's just a picture, what brings up the question

is there any point to spend time in trying to decrypt it. If a cryptanalyst will see a kind of 'picture' (inserted in a text) with many lines and dots etc., it will psychologically depress him (or her) with the feeling of wasting the time – maybe the picture itself is a message while the lines, curves and dots in it aren't carriers of information but all just noise meant for confusing. But if the picture really contains a code with a hidden message then who knows what in the picture can be converted into digits to begin decryption with?

Using this method of encryption you can paint pictures with your own rules of encoding. For example you can decide (during creation of your encryption method) that one hidden symbol in the picture will discard all the other characters leaving only dots and lines as information carriers, thus you only need to convert them from 0-s and 1-s into amount of levels in your personal matrix of symbols (TS) to hide a message and to derive the hidden message. You don't need to change your personal matrix for each occasion – you will create it once and for ever. All the different methods of encoding will be decided around it using different switches for changing the patterns of access of the information in the matrix. That makes the method of encryption uniquely adaptive and flexible for all kind of ways of representation, for almost any occasion of necessity to secure written information from unwanted access.

7.3d Hiding method.

There's a possibility to go even further and to hide the real encoding symbols (CS) in the script by following certain rules – the code symbol can be taken apart placing different parts of it into background noise. While you're capable to discard the noise easily you can reassemble the separate parts of the symbol in your mind into a single glyph while reading the script, and to decode it. I haven't paid particular attention to this method, simply figured out the possibility to do so. If you're creative enough you can work out if it's a practical method. In conjunction with other methods, with the help of switches, it may be applied for deeper confusion and diffusion, or for hiding a message in an art piece. The hiding method allows you to design almost any type of hieroglyphs just for a visual pleasure. With the set of few predefined additional elements for pointing out which glyphs and which parts of them carry information and can be combined into real glyphs which have a meaning, you can hide a short message in a beautifully designed meaningless script. Cryptanalysts will hate you for that.

7.4 Writing styles.

You can write the script in any direction, as you like, but certain rules of encoding must be modified to change from horizontal to vertical style. Personally I prefer encoding in horizontal style, thus almost all my methods have been around horizontal style, but I do realize that it could be a good idea to include rules of encryption for every direction of writing within the same method, because you never know where you might need to write down some data. Extending the possibilities of your method is never a too bad an idea.

Here's an example of vertical writing -- it's quite obvious that I haven't practiced too much writing in vertical style, so the design isn't coming out very exciting..



Style can be not only directional but also calligraphic. For that purpose is better not to give different values for the modified types of lines. The values are better to keep the same regardless of width of lines, different shadows, colors etc. If you hold onto the sizes of curves and angles, breaks in lines and the direction of some of them (in specific conditions) as values, then you'll have a greater flexibility in design being able to exploit the noise to the maximum. If you give different values to colors and shadows then you may lose some data when making copies of your calligraphy artwork.



A writing style can be made so barely distinguishable from background noise, that it can nicely be embedded into a picture above a safe in your house. So whenever you happen to forget the code of the safe you can decrypt it from the picture, while others won't be aware of the existence of the code there.

7.5 Using switches.

Switches can be anything, even the style of the script can be used as a switch – for example if you write in one style then you apply one set of techniques of encryption and in another style another set of techniques, or other patterns of diffusion. Some stylistic elements, like colors and shadows, are not good for using as switches, but the general style itself can be connected to a particular method of encryption. Most switches must be decided during creation of your method, like the ones which are attached to glyphs and meant to switch between data 'calculation' methods within a glyph, and the separate ones which are meant to switch between diffusion patterns or carry other commands.

You can add switches to already existing ones in later times, but you must be careful because if you happen to use newly designed switches similar to previously used

noise then you may confuse yourself. It is wise to have a reserve set of symbols for any occasion, or to have one predefined reserve symbol – so called modifier of switches, which you never use as noise. Also is wise to have in your matrix several empty (reserve) spaces which you never use as noise.

The more sophisticated ways you use switches the more secure will be your method. For example the separate switches (which are not attached to glyphs) you can divide into several groups: each group will have specific location of application: above, below and so on. If a switch is used in a wrong location it can be noise (or for design), for highlighting (for keeping track on particular data) or meant for marking an error, but not applied as a predefined switch. Some switches may be in a group where they only obtain a value or a meaning when used together with another predefined sign or in specific conditions, while without those conditions are design elements or noise for confusing others.

You may also give different values to similarly looking switches pointing slightly in different directions, or are of different size related to the size of the glyphs – each difference in representation will confuse cryptanalysts deeper and make an attack on your script ever more difficult.

Never give different values to the different sizes of glyphs, only to the relative sizes of parts of the glyphs which have meaning when used together with a base glyph, but separately have no value.

This is an obvious rule I guess. Regardless that in ideal conditions it is not advised, it may happen that you will need to write an additional glyph above already written script, or to squeeze it somewhere between. You may also need to write some complex hieroglyphs a bit bigger than the rest of the script as it may not be easy to do in small script with all the details if you are already writing the script quite small.

The difference between commands and switches is in the general application. For example hidden commands can specify reading patterns of cryptic symbols, there can be copy/delete commands, error correction commands, and so on, whatever commands you create for your code. Switches can be used to correct things. When a set of commands is in one block of a matrix and you have specified it wrong by one or few steps only, then instead of writing a cancelling (deleting) glyph after the erroneous one and then writing the correct command, you can apply a switch attached right to the erroneous command, correcting it for how many steps necessary with the particular switch. Switches can also be used for specifying language if in the cipher text there are several languages used, instead of in each section to write the command of the language to use. And so on depending on the rules you create for the application of switches.

A handwritten signature or scribble in black ink, consisting of several stylized, interconnected loops and lines.

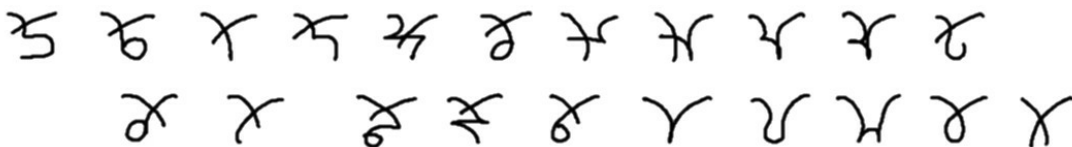
8 Encryption symbols.

The encoding/encryption symbols (cryptic symbols - CS), are the complex symbols, base glyphs with separate attached parts to them (characters), sometimes with additional signs which are although not attached but integral parts of the glyphs, all together representing particular meanings. Each part of them when used separately will have a different meaning, thus are different encoding symbols. Added to the complex symbols modifiers will change the meaning of the symbols but modifiers on their own are not always encoding symbols – when used in a wrong place the modifiers won't represent or change anything, thus can be used as noise or design elements without affecting the meaning of encoding symbols. All of the integral parts of encoding symbols (glyphs) must be decided when the system is created. Some specific base glyphs, characters and modifiers can be made idle for use in later times when other meanings (for example words which you often use) are needed to be represented with single hieroglyphs. These idle glyphs, characters and modifiers must not be used as noise but kept in reserve for such occasion or they may create confusion while reading older scripts.

8.1 Base hieroglyphs.

Base hieroglyphs must be created in such a way that they will never confuse each other when additional features are attached to them, be it characters with value or noise. The more base glyphs you're able to create and remember the more sophisticated will be your method, but at the same time there's no point to create too many base glyphs as then you will lose in flexibility to use additional features: if your base glyphs are in big number and complexity then attaching characters to them may create quite similar representations of several hieroglyphs, especially when you write them fast and don't pay attention to details like relative size of different parts of the glyphs.

There are thousands and maybe even hundreds of thousands of ways to create sets of base glyphs. Here are just few examples of possible base hieroglyphs:

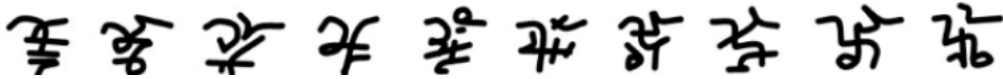


The base glyphs (base cryptic symbols - BCS) must have the carrying capacity of attached cryptic symbols (ACS) and space available for separate cryptic symbols (SCS), while the BCS-ACS relation don't need to be obvious to others, better not.

The base glyphs don't need to be that simple as in the above samples. You can create way more complex base glyphs that will look like they already have some attached modifiers on them, but only you will know that these are actually just the base glyphs.

Thus anything that differs from your base glyphs can be used during encryption for noise and general design of the cipher text.

Here are some examples of more complex base glyphs:



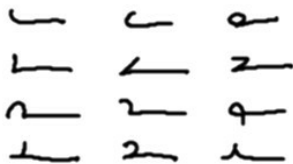
Notice that regardless being complex they have the necessary capacity to carry extensions and additional characters (attached to the base glyphs), and of course the space for separate cryptic symbols around the glyphs.

8.2 Characters.

Previously, in the section 'information carriers' (chapter 7.1a), I gave examples of data points in hieroglyphs. Here I'll show some possible characters to attach and to represent information in the data points. The characters can be used as constituent parts of the glyphs, as modifiers, as switches to change the reading pattern of other characters, or even as commands to change the whole meaning of the set (like command to switch from matrix of symbols to matrix of words, if you have made one). There are simpler ways of use as well – some characters can be defined as replacement of some actual symbols where the symbols in the words are obvious (to make the script shorter, for kind of abbreviation), or for other assisting tools. While deciding some characters for use as the simpler tools you must think if by a frequency analysis they could give a hint on something – better is not to use them as punctuation marks or sort of things.

You can create a fixed set of characters to be used in different data points (info carrier points) while having everywhere the same value (so you won't get confused yourself and make less errors), be it a symbol, meaning, noise or switch, or you can give to the characters changing values. Personally I use different values given to the same characters in different data points and even switching their meaning in the same locations when certain conditions are met, without applying visible switches – it's fun, good brain training and gives to the cipher text extra security.

Some examples of characters:



On top of creating more different characters there are many ways to make these characters even more numerous, like making them longer (by adding extra curve or angle), crossing the line through right before the character with different types of lines, turning around (in above example turning down) and making them more precise (adding

more values by subtle differences).

Here's an example of complex base glyphs extended with characters:



As you see I didn't use all of the locations available for extensions with characters and I didn't extend the lower parts of the base glyphs nor the high above -- these spaces are (in this example) for different types of attached characters -- you'll see how it works in the example below.

Here are the same glyphs with more attached modifiers, different sets of cryptic symbols/characters specialized for lower and upper parts of the base glyphs:



You can see that none of the lower and upper attached cryptic symbols extend from the internal attached characters, only from the base glyphs.

You can apply any modifications to the meaning of the characters with no need to learn each modification to each character -- if you create a system to all your characters and learn them like a kind of 'alphabet', then for example crossing a line through with another line can mean take one step back (in the 'alphabet' of the characters), or adding a dot next to it can mean turn the character down (meaning: read the character as turned down), or any other modification of your choice.. a dot above or below the character can also give a different meaning to it, or to cancel each other if used twice.. there are countless ways to confuse others -- nobody else will know which signs in which places will mean something and which are there for making the script just to look better, while the characters firmly keep their values.

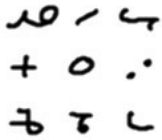
To make it harder to guess the different parts of the glyphs (BCS-ACS), in case they're difficult to picture in mind in advance, you write them first down on a separate piece of paper (which you destroy later) and then continue with the cipher text without breaking the line in the attachment point if possible -- the glyphs should not show where the ACS were added to the BCS, because you wouldn't want the BCS to be easy to determine.

8.3 Additional signs.

These are the separate parts of the glyphs which are meant to change the whole meaning of the set, be it a hieroglyph, a block of hieroglyphs, or even the whole script, or simply to point out something for you for easy following. As previously described, some of

the signs can also modify the modifiers, the characters, if used in a correct location decided during creation of your method. Depending on the method of encryption the separate cryptic symbols can be a part of a data set to specify a real text symbol (TS), or have a value for a specific dimension in the matrix of symbols (TS), or be context dependant with different applications.

An example of additional signs (SCS):



Most practical use of them is as switches but some additional signs can be used for giving additional information without changing the value of the glyphs or characters, for example to highlight parts of a script.

Here are the same hieroglyphs used in the above examples with some separate cryptic symbols added:



All simple and clear, isn't it? When you use simpler base glyphs then all the attachments will be easier to read, but at the same time you don't need to add so many different cryptic symbols to each base glyph -- if it gets too crowded you can use simple rules to continue with the same set of values (for accessing data in the matrix) on the following glyphs, or use simpler assisting glyphs to complete the set. In the end of the book, on the pages 'to print/copy' (chapter 28) you'll find a compact version of the example, step by step different encryption symbols added to the complex base glyphs. That's the only page there where you need a color printer (or color copy machine) for clarity.

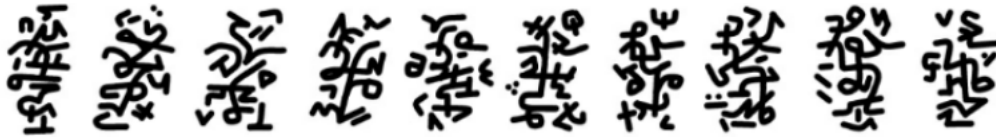
You can use the same signs in different ways. For example in context dependant application, if the same additional separate sign is written together with a hieroglyph representing a symbol (TS), then it can change the symbol (e.g. one up or one down in the alphabet or in the matrix), if used together with a glyph representing a command of diffusion pattern, then it will change the pattern, or if used together with a glyph which carry no meaning the same sign will simply highlight the part of the text giving it a value (like meaning "bold text", or pointing out new line or chapter). In another place the same sign can have no encoding value and be there just for your easy finding of the part of the script later.

Depending on context the additional signs can carry any values decided by you during the creation of your encoding method. You can also have one particular 'fun' sign which render whole the block in the script 'useless', so you can design any hieroglyphs of

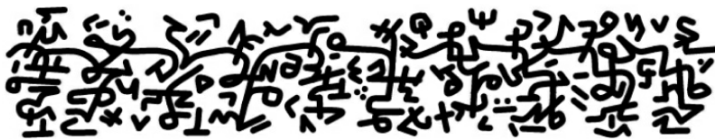
your choice in the entire block (not too different from the rest of the script of course, not to be obvious) and it won't affect the rest of the script, but the daring cryptanalysts will have fun on the script for entire life with no success..

..by the way, you don't even need to have such a sign, because if you write some hieroglyphs in the block in the way which can clearly (for you) have no meaning by your encryption method, then that's sufficient for marking the entire block useless (if that's the rule you created for the case of using unknown cryptic symbols), and a single dot in these meaningless hieroglyphs can point out that only these glyphs are to be discarded while the rest is still a part of the meaningful script (by the clear rules of the method).

So, here are the completed hieroglyphs as they would look..



And here's how the connected cipher text would look..



Almost perfectly 'square' hieroglyphs.. which will hardly ever happen naturally. To accomplish that you can add ACS and SCS to fill all the gaps -- these versions of writing ACS/SCS which carry no encryption value, just as noise and design elements. By filling the gaps you will make the code safer and will be better repelling cryptanalysts, while the cipher text will be looking better avoiding bizarre shapes.

Writing of the hieroglyphs in a 'perfectly square' mode isn't the best by the design and cannot really be called a calligraphic cryptography, but that's the safest mode -- the more noise you add on the go, after the code symbols are completed, the better. But that way the code also becomes more difficult to read. You must come to a compromise between an easy to read cipher text, a beautiful cipher text and an extremely safe cipher text. In my opinion you can easily go for the calligraphic, beautiful cipher text when the rules of accessing data in the matrix of real symbols are creatively complex. These hieroglyphs above were just an example to which level of complexity you can go if you really want to. The choice is further yours. In any case, who will be enjoying the beauty of the cipher text but you.. the cryptanalysts won't give a damn about it, I suppose. But I may be wrong. In that case you shouldn't do their job any more pleasing, rather let them hate your code and make more mistakes in their search for the encrypted message.



9 Lists of symbols.

This section is about the lists of symbols which you will need to include while you create your matrix. I dedicate to the lists some attention because I've been going through several systems – some too simple and some too complex, so I know what you would actually need for the encryption purposes without being forced to begin the creation of a new matrix after a while.

9.1 Classic text symbols (TS).

9.1a Alphabets.

Alphabets may in some cases seem difficult to include in their entirety because many languages have a large amount of modified letters (ã, ā, ä, à, á, ã, â, â, ç, ...). Writing them all separately into a matrix is pointless because the encryption glyphs provide the flexibility to attach seamlessly several modifiers. The additional features of letters – the additional characters written above, below, beside and attached – are better to include in the matrix separately (˜, ^, ˇ, ´, ¸, ...). You can then create rules for combining them into single glyphs representing modified real symbols ($a\tilde{} = \tilde{a}$). Often there's no need to write the combinations out, only in these occasions when a word could otherwise be understood in several ways. But if there are just few 'special' letters in a language, like in Spanish (ç, ñ), then it's more practical to include the letters entirely. Myself I use just some simple extra characters as I see no point to include in the matrix all the possible characters of all the languages I may speak one day – I have no plans to encrypt in any other language than in the five which I speak well.

The languages which use hieroglyphs of their own must be included in a matrix the way they're spoken, with separate modifiers for different tonalities and accents when necessary. As far as I know for all the languages which are written with hieroglyphs have also been created writing systems based on Greek and/or Latin – Japanese Rōmaji writing is one such system – thus there's no need to become a linguist to be able to create your encryption matrix.

If you speak several languages then in some cases the alphabets can be merged to include all necessary representations of letters in a compact way.

Definitely you'll need to include Greek alphabet with different written representations of some letters, because these are internationally used symbols for mathematical formulas and some of them are in common use.

9.1b Numerals.

Numbers must be in several sets because they are the easiest to crack if it is somehow known where you use them and where they're stored in encrypted way.. phone numbers should be encrypted differently from your bank codes. On top of simple sets of

numbers it's a good idea to have at least one set of complex numbers.. complex not in the meaning of mathematical formulas but in the meaning of encoding. It's quite enough if you have a way to represent pairs of numbers with single hieroglyphs, and with a modifier be able to use the same pairs of numbers in opposite direction for example.

For some time I used special modifiers to create from a double set of numbers triple sets, and these modifiers I could use for letters too. The thing is there's no need to have even more complex modification tools as they can rarely be used. But if you do need to use lots of numbers in encryption here's an example: take the double number 01 and with just a few modifiers you can do from it 010, 10, 101, 001, 100, 011 and 110. If you create clear rules for the modifiers it's ok, but if the modifiers confuse themselves (in the priority of application for example) then you may make mistakes during encryption.

9.1c Mathematical symbols.

There's no need to include all the mathematical symbols in existence.. and it would be one pointless undertaking even for a professional mathematician – why should anyone encrypt complex mathematical formulas? At the same time leaving out a symbol which you will never use you may think, but is still quite often inserted in a practical daily context, will have you writing it down as a word during encryption. I could give you a simple set of symbols here but there are two main reasons why I spare you from this information in this book: first of all it would disclose the symbols I use in my matrix and for second, if you take my set of selected symbols it will make an attack on your own code easier as well.. it's for the safety of everyone who will create his/her method based on this book that the selection of symbols (TS) for the matrix is made personally.

9.1d Other symbols.

There are many common symbols which are quite often used in texts, thus you must include those too, for the cases when you would like to encrypt citations for example. Remember, you're creating a method of encryption for whole your life, for any occasion. Although you can also encrypt symbols using words (the way you pronounce the symbols) it's easier to include common symbols in matrix once and forever, for any occasion.

If you use complex passwords in your computer works, passwords which include not only letters and numbers, then it will be smart to include in your matrix all the symbols available on your computer keyboard, so whatever password you'll create you'll be able to write it down on a paper in encrypted way. There are many services which don't allow to use only simple symbols (only letters and numbers) in a password so you will need to include all the keyboard symbols in your matrix just in case. If your computer's keyboard doesn't have all the symbols written on it then type into a text file one by one all the symbols holding down 'Ctrl' key, 'Alt' key and 'AltGr' key consequently. Some keyboards have different options, including keeping down several keys at time. You must try all the options and then you see which symbols you can create on your keyboard, to include them in your matrix. Same with a tablet -- all symbols available on tablets must be included in the matrix, for case of using in passwords and thus encrypting the codes. That's easier to do than choosing the right math symbols because there aren't too many of them on common keyboards of the devices, be it tablet or computer.

One option is to leave several blocks of your matrix empty and each time you use a new 'rare' symbol you add it there. But in this case you may forget a symbol after a while because the added symbols accumulate randomly. It's safer to create a system and to include all common symbols in blocks leaving for each block one or few empty spaces, then you can add new symbols in an orderly way relating them to specific blocks of symbols.

For my latest matrix I looked out on internet the best existing collections of symbols and selected from them all I felt to be necessary, put them in the order I could easily remember and learned them. I never wrote them down in a system which I would use in my matrix of real symbols – I simply learned just few of them at time, the ones which fit into the system I had in my mind, then placed new symbols which fit together in another section of the matrix, and so on until I could remember them all. It's possible to put into an order and to memorize them all at once, but my experience shows that the 'perfect' system I create too quickly isn't perfect the next day after completion.

9.1e Highlighting and more.

Including highlighting symbols in your matrix may look useless, but you never know when you may need it for not to be forced to do the highlighting in an obvious visual way. I personally included in my matrix all common options of text modification, just in case. If you systematize all the options and keep them in specifically designated areas of matrix then it doesn't take much effort to memorize them. It's good to have these tools ready for use there as it gives you a great flexibility in encryption of all kind of data without giving anyone even a hint what the script may be about.

9.2 Cryptic symbols (CS).

The symbols of encryption are those which make up the main core of the system you create. Substituting common symbols (TS) with other similar symbols is a child's play – it won't offer any protection for your data. Even a doctor's handwriting may protect the content better than a simple substitution.

While the cryptic symbols are in a separate list from matrix, all the commands that are meant to change the pattern of reading of the code need to be included in the matrix even when you have fixed some glyphs and/or characters (and also separate signs) to the commands of application of the patterns directly. And of course you will need to include in the matrix the rules which help you to disguise your errors. These encryption commands and assisting tools can be (but not absolutely necessarily) related to specific encryption symbols which can be similar to glyphs, modifiers and/or additional signs, which make the encoding/decoding easier, but if you include the encryption symbols in matrix as well then you can represent any of these rules (commands of application) in many different ways as hieroglyphs, not just as specific encryption symbols. This increases security of the method.

Generally, almost anything that you represent with a cryptic symbol directly is better to include in the matrix with it's own location, then you can represent everything in many different ways. That may become very helpful in a situation when a specific

plaintext needs to be encrypted using the same fixed cryptic symbol over and over again, which you wouldn't want to do for the security of the code.

9.2a Diffusion symbols.

These are the symbols which give commands to apply all kind of rules to change the way you write and read the script. 'Classic' commands are to swap already written (or the next) hieroglyphs, to repeat, to join (when there's visual break in script but a word actually continues), to separate (instead of visible space), and so on. You can create as many diffusion symbols as you feel practical to use, all of them related to the specific diffusion commands, but for greater flexibility and security of your encryption method you better use one location for each diffusion command in the matrix as well.

9.2b Correction symbols.

Mistakes do happen. You may need to correct an error if the error isn't obvious and possible to neglect (for example a forgotten letter in an obvious word, omission of which wouldn't affect the meaning of a sentence) and may confuse you (or receiver of the message) during decryption. Rather than correcting an error in an obviously visual way it is safer to have a set of correction tools at hand. For example you discover during encryption that you wrote a password wrong forgetting a letter or digit in it or writing with a wrong symbol, then it will be good to have the correction symbols meaning 'replace back', 'insert back' and others. When these correction commands are used together with a number from one set it can mean how many steps back they replace or insert, or when used together with a digit from another set of numbers it can mean how many symbols to correct, and from yet another set of numbers taken digit can actually mean the digit you wanted to insert or to replace with. As an option, the difference in meaning of numbers during correction can also be established with the help of modifiers.

9.2c Assisting symbols.

These are the tools which make the script even more flexible so no cryptanalyst will ever be able to find a system in your script.

To mark a letter as capital letter or whole the word as consisting of capital letters you can use specific symbols (CS), and the same symbols applied for capital letters can mark them small. It will make the work of cryptanalyst much harder to find out which letters are actually capital letters meaning names or beginnings of sentences – by statistical analysis it can be done if you have habitual patterns in using other diffusion tools, thus hiding capital letters is yet another help in diffusion. And there are more tools available.

Theoretically it's already extremely hard to deduce any meaning from the properly diffused hieroglyphs, but as we are humans having certain habits then adding yet another assisting tool, no matter how 'pointless' or little professional it may look, makes figuring out some patterns from your script harder still. Every additional tool for flexibility of the code adds to the security of it.

Definitely you will need to add tools for marking symbols as shifted up and made

smaller for encrypting such common symbols as square meters or some mathematical formulas. If you have made them visually spotted in the script then you may give out the meaning of the script and some used symbols in it.

Other possible assisting tools can be for marking specific locations in the script for easy following, tools for repeating (to bring attention to an important part and encode it in different ways for avoiding errors – thus you know that you don't read it as written twice but just as reconfirmed section of the script) and more. They are not visually obvious (to others) on their own because they will look similar to the rest of the symbols. Also you can include tools to assist in the design of the script to make it look better, more aesthetically pleasing. Basically the tools are specific hidden symbols for creating meaningless hieroglyphs inserted into the meaningful script.

All the assisting symbols are good to include in the matrix not only for greater flexibility and security of the code but also for easier memorizing. These symbols may be in large quantities and you may forget some of them if not used for a while. Your matrix, which has an established number of symbols included in each block, will help you to repeat each and every one of them with no worries that maybe you have forgotten few of the assisting symbols. Keeping all the commands and tools in the matrix in a clear system avoids the situation where you've forgotten that you've forgotten.

9.2d Modifiers.

These are all the assisting symbols for changing the meaning of already used tools or symbols when necessary, either for deeper confusion, for diffusion, or correction. The modifiers have already been discussed above. One of the possible applications of them may be predefined editing, when you change the meaning of a previously written wrong command without the need to use deleting tools to cancel the wrong command and write a new one. The predefined modifier can simply specify the correct command by placing it in the right position next to the glyph.. even a single dot in the right location can specify the right command which the modifier is meant to change. It may sound too complicated but you don't have to remember all the rules of editing/correction separately – remember, you create a system in your matrix, so using the correcting dot may simply change the command to the next or previous command in your matrix, it's as simple as that.

You must include the modifiers in the list of symbols in the matrix because you may have no space around a written glyph (to keep the design of it nice and the script smooth) to add yet another modifier if few are used, or if the glyph itself is large with all the characters attached.. in such case you simply add another glyph which represent the modifier in your matrix.

9.2e Switches.

You should create a set of switches which are ordered the way you can remember, also creating corresponding set of switches in your matrix placing them in the same order. You learn them once and for ever and create a few reserve switches with corresponding reserve places in the matrix. Switches are also modifiers but they're rather meant to swap between previously specified commands (for patterns of diffusion for instance) or symbols

(letters, numbers) depending on context, instead of applying any specific rules on their own.

To clarify better the distinction between modifiers and switches (in my method, and thus in my definition of them) I give you an example. Let say you have ten different diffusion patterns between hieroglyphs -- ten different ways of reading the hieroglyphs in different order. And you have also ten different reading patterns (orders of priority for application) of characters within the hieroglyphs. Each of the patterns will have a specific modifier related to them. Then, during encryption you will see that one of the modifiers you want to use isn't fitting into the hieroglyph due to the lack of another free attachment point or free space. That will force you to write the following hieroglyph specifically for applying that modifier. Then you see that the modifier of the same value from the other set would fit perfectly to the glyph at hand, but to specify it correctly you will also use a specific switch meant to swap between the two sets of modifiers, so both of the sets can be used in both cases. The specific switch itself can be just a dot nearby or a crossed line on the modifier itself, or the switching can be accomplished by turning the modifier upside down, in case if that turned modifier won't be similar to another cryptic symbol.

The rules of application of switches must be created together with modifiers, insuring that there won't be any confusion between them.

9.2f Other tools.

The additional tools which you may want to use but which haven't been pointed out as absolutely necessary depend on what you are planning to use the encryption for. In any case you better leave a block or two in your matrix empty and you won't be in trouble when you'll decide to advance your method.

If you want to keep a diary or write a book in encrypted way then you may include several 'memory tools' for repeating a word or parts of the text with a single symbol. The tools then need to consist from several symbols -- one for marking the beginning of the text to memorize, other the end (and a number you assign to it if you memorize several sections), and yet another symbol for pasting the section into the script (from the specified memory number) when needed. If you encrypt a long text it's quite a useful tool, but it cannot be used if you need to take copies of separate parts of the script in later times -- you will not know later what was the word or section of the script you pasted there. When you use special memory commands (save, add, apply) in combination with specific spaces in the matrix related to the letters of alphabet and numbers, then it won't be too hard to remember 'saved' words during encryption and decryption, after they're written out once -- the letters of alphabet under which you stored specific words make easier to remember them. And if you use in your matrix combinations of symbols, like syllables, it will be easier still to 'save' words for temporary use in one cipher text. While some long words may be used in specific texts quite repeatedly, using the memory option for encryption saves up encryption time and the length of the resulting cipher text, which also makes the cipher text more secure. Common sense tells that the words worth saving must be at least five letters long or more, because the memory commands must also be encrypted, thus gaining just one-two symbols in length isn't worth the trouble. The more you use the word in text the more sense it makes to save it for later use.

Some common words which you use almost in any text are also quite long, but a comprehensive list of them could be a challenge to create, for use in the encryption matrix

with permanent replacement symbols. Long ago I created such lists, but then later decided it's not worth it even when the same substitution symbols for words can be used in different languages. The reason for abandoning the lists of permanent words was that many of those words change in grammar depending on context, requiring prefixes or suffixes or both, which makes the trouble of using them pointless – it's easier to encrypt each repeating word differently, with less chance for errors. If just one symbol in a word gets encrypted wrong it will still be possible to understand it in context, but if a single symbol representing a word gets written wrong it may change the entire meaning. For example if the words 'before' and 'after' are single symbols in the matrix, then depending on representation in the cipher text that may be a concern for an error. Anyway, if you like it that way, then single symbol words are fun, and when necessary for avoiding a confusion in specific cases, if something important depends on it, then you can avoid the substitution symbol for that particular case. By writing the word out (while having a symbol for it but not using) you will be stressing the importance of it in the context, which is also a good way to go on important issues. You should always remember, during creation of your own encryption method, that you're not creating it for computers where possibilities of errors are negligible. You will always have occasional errors during encryption, no matter how good you're at it after lots of practice. It's human to make errors, and that in mind I created the method which allows the errors to be present, yet continuing the cipher text to be deciphered with minimal loss of data by those knowing the key and the matrix of symbols.

Another tool you may need is to mark a letter as encircled, to create the copyright sign for instance, but you may also have these encircled symbols, like registered sign and others, included in the matrix. It all depends on your likes and preferences.

If you need to encrypt quite often some financial relations then you will need memory tools to create several monetary signs and to change the signs regularly, to represent the same monetary symbols in many ways without being forced to use too complex hieroglyphs when accessing and combining different parts from the matrix into a monetary symbol. Simply call the symbol by word (e.g. “euro” or “yen”) attach a code symbol meaning “create monetary symbol”, and add one of the idle memory locations in your matrix.. thus for the entire encryption piece you will use this memory location for this purpose. If you use only a few monetary signs then you can include them in the matrix permanently as well.. from the keyboard of a computer the €, £, \$ and some others should be included anyway, to be able to encrypt complex passwords.

You can also create specific “memory glyphs” (as base glyphs) while the modifiers will be giving them different memory assignments. But even these options are better to include in the matrix as well, just in case.

Almost anything can be encrypted. You only need to assign necessary meanings to specific symbols and to create the rules of modification when needed. If you do it correctly then you'll be able to encrypt large amounts of data in much shorter versions of script, simply because you can represent whole words and meanings (like 'I must pay', 'he owes me', 'personal debt', etc.) with single symbols. Instead of writing long numbers repeating many zeros you can create symbols which represent 3, 4, 5, 6 zeros, all doubled in matrix and possible to write many different ways, not to be understood by others.

This encryption method gives you an enormous flexibility – it can be used as a kind of shorthand for some information which isn't even possible to represent easily in 'normal' ways. You can create tools for your specific needs, the tools which aren't even in

existence in normal writing. For instance if there's an idea which to express in normal ways takes quite a few words, but you use it often and would like to include in the encryption, then you can create for the whole idea a single hieroglyph.. and modifiers will help to specify it if needed.

Personally I have used up almost all the places in my matrix, leaving just a few squares open just in case, because after 30 years of work on all kind of methods I know what I need (not too much), and how to go around the problem if I should run out of the last few squares in my matrix. If nothing else helps and I do need to create extra tools and symbols one day, maybe after learning yet another language, then there's possible to create a separate matrix for this specific language, and to create switches (by some unused combinations of symbols/signs) to change between the matrixes. Also it's possible to create specific glyphs for that purpose, from the millions of possibilities out there.

By the way I'm using a multi-dimensional matrix – it makes storage and handling of the information easier in many ways and at the same time makes the representation of the data in cipher text more diverse, while increasing the strength of encryption exponentially with each extra dimension.

茶 茶 茶 茶 茶 茶 茶 茶

10 How complex code is good enough?

Perhaps you have realized how strong can be the encryption method, how difficult it will be to crack the code even if you are a bit careless during encryption. It must be made so strong as principle because as humans we make mistakes which weaken the code. So if theoretically it is impossible to crack the more advanced versions of the code, and even if someone will have access to all of your encrypted writings, including diaries with encrypted private entries, then all the cumulative unwilling giveaways of context of the cipher text and a bit careless encoding in some places should leave the script still impossible to crack.

You may think that creating a simple code is good enough for your needs, without paying much attention to all kind of tricks to make it a well secured advanced code. But think about it how often you would actually want to use your method: only occasionally (codes), for one thing only (a diary), or you want to be able to use your encryption method at any time during entire life, whenever needed? Sometimes we can make really bad mistakes in life, so it may happen that you do leave around a plaintext of some notes together with the encrypted text. If your encoding method is too simple then one such mistake will compromise all your encrypted data in the past, and if not noticed then all your future encryptions as well. Is it really worth spending the time then to create your personal method? Perhaps it will be safer to rely on your memory alone than on a weak encryption, in case of passwords and other such type of data.

Taking the time to learn the art of handwriting cryptography well, with all the precautions and tricks, is a good investment to avoid any trouble in the future. If you create your own method which have not just a single pattern of encoding but at least a hundred ways of representation of the same symbol or word (which is still a relatively simple code) you may feel quite safe that someone having access to the plaintext data together with the cipher text will not derive by it the rest of your private information.

The most advanced encryption method which I'm using (of course) has a diffused encoding of symbols (TS), in the meaning that I don't always calculate a hieroglyph to represent something (actual symbol, word, command, etc.) from my matrix, but I often calculate the 'something' diffused between several glyphs. This method permits me to write the script more diversely and smoothly, applying design elements almost whenever I like.. which also makes the code stronger and any mistakes made during encryption easy to hide (with the help of the design elements). But that method is better to learn (create) after practicing with a bit easier methods (explained in this book), because for writing in my latest encryption method you will need to calculate several glyphs at time, in advance, before beginning to write them.



11 Systematization and memorizing.

To be able to systematize the glyphs, characters (attached to glyphs) and additional symbols (separate) you begin with writing them down.. first simple ones, then more and more complex glyphs. After you have practiced for a while you'll begin to see the possibilities to systematize them. You must create at least three systems – one for base glyphs, one for characters and one for separate signs. There's also possible to create a system of glyph extensions, with any purpose and values given to them. You can see the extensions in the script examples in this book, below some glyphs.

The extensions are not the same as attached cryptic symbols (ACS), although they may look quite similar. The ACS are in a different system, much smaller than base glyphs and mostly used as modifiers, while extensions to base glyphs are much larger, attached below the base glyphs and in some cases can be the size of the base glyph or even larger. I don't discuss the glyph extensions in detail, you can give them any values and use for whatever operations of encryption you like. Also the ACS can be below the glyphs but to be limited with the options of attachments below the glyphs there's no point. Below the glyphs there's enough space to design very elaborate extensions, as diverse as the base glyphs themselves, while on top of base glyphs the use of extensions isn't advised in such an extent, for not making the script to look horrible. You will see some writing options of the extensions in the medium level demo code later in the book, although there the extensions are used differently, not as true extensions.. but you'll get the general idea of the design. Some secrets of my encryption technique must remain secret, thus the use of extensions is somewhat scarce in examples -- you'll create your own unique tools as well, with a bit of creative work. I only admit that in the old very complex substitution method (mentioned in chapter 2 where I talked about the history of the method) which I discarded, I used the attachment points and the extensions as other sets of alphabets and numbers. And also in case of separate cryptic symbols similarly, having large amount of different sets of additional substitution symbols. While many additional cryptic symbols already had some simple, very basic commands of encryption, not being used only as the substitution elements, the simple substitution method allowed to write in a single complex hieroglyph words up to nine letters long, some even longer but some shorter, depending on design of particular glyphs and the repetition of letters within words. The BCS, extensions, ACS and SCS all had their different alphabets and sets of numbers and so on, but I didn't call them as such, they were just all kind of different substitution symbols for me, combined into complex forms as hieroglyphs.

Thus, if you decide also to use extensions as encryption values, not only as design and noise elements, then you must create the fourth system dealing specifically with extensions. You must practice them as well, separately, writing them down in all possible forms and shapes, to discover the systems you can put them into, to decide which of them you're going to use for encryption, and consequently leaving all the other possibilities of writing them as design and noise elements. I can assure you that if you begin dealing with extensions you will discover entire new world of writing parallel to base glyphs, so maybe it will be wise to leave the extensions for later use as an option, after the BCS, ACS and SCS

are clear and decided. You shall not use extensions as noise and design elements if you consider using them later as encryption elements. Or, you must make clear distinction between the encryption methods – one with the use of extensions as encryption values and the other without, where all the extensions are just creating noise. This distinction can be determined with specific cryptic symbols within cipher text.

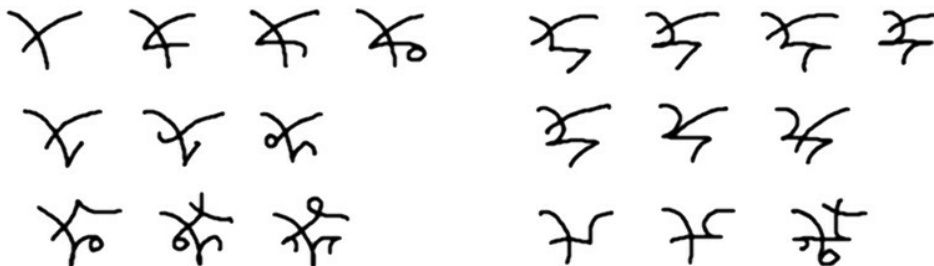
The fifth system of cryptic elements you can create could be the system of connection modes between hieroglyphs, but that will be fairly simple one and will come naturally, thus you don't need to create it right in the beginning. Rather let that system of extra possibilities to be idle until you have another idea of encryption but will run out of already systematized elements in your method.

The sixth, seventh and more systems can be created for specific base glyphs which are very different from the main base glyphs. One example of such glyphs is in the medium level demo code, later in the book.. the elongated tall glyphs, used in the example as correction glyphs. There can be such design of base glyphs that would look great in the cipher text and you would like to use them for something, but they won't fit into any category, because some designs may not allow to easily attach the characters. Then you create a separate system for those cryptic symbols, and yet another system of ACS specific for those kind of base glyphs.

The systematization of base glyphs and characters must be done in relation to each other in the way that some simple base glyphs together with attached characters won't be similar to some more complex base glyphs. You simply write the glyphs and attached characters down whatever design you like and eliminate from one of the systems - base glyphs or attachments - those elements which can be confusing. You can then continue to practice with ever more complex hieroglyphs to see the possibilities of characters.

The ACS can be broken down into different systems for each location within a single glyph, but that would rather confuse you because there are so many different ways to write base glyphs. Still, if you like it you can do it.

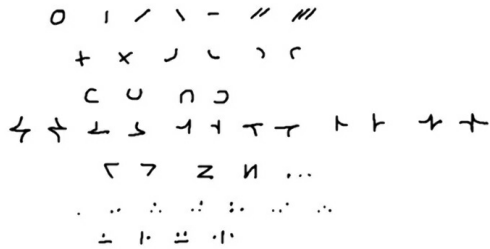
Here's an example how you can begin training yourself, writing simple glyphs in ever complex way. Some parts of them can later be used as attached characters..



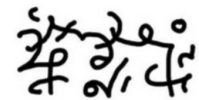
For creating and finding a system for all the additional (separate) signs you need to write them down too as they come to your mind and after some time you'll see the possible systems you can put them into, for easier memorizing. As it is not advised to leave your systems around for someone to see then create the signs in little blocks writing on

paper, learn them as a system, and destroy the notes right after memorizing. After a while you'll be able to do it in your mind, assigning to the groups of signs related values or commands in a clear order, and with a bit of creativity no-one will ever be able to guess what the signs are meant for.

Here's an example of additional separate signs, written down randomly as they come to mind, to select from them the preferred designs and to put them into small groups, into subsystems within the system of SCS ..



When it looks to you that all the systems are well done try to recall all of the symbols – each and every single one of them. If it's hard to remember then you can modify the systems. You can do it over and over again until you end up with the systems which meet your needs both aesthetically and in writing precision. The writing precision is the requirement for details in your script – the more precisely you're able to write your glyphs the more values you can give by all the subtle differences in representation of the glyphs. It will be necessary to decide the precision right in the beginning, before you create your method, thus before creating a system it will be a good idea just to practice the writing of differently designed glyphs without giving them any meanings. The practice of writing of the glyphs works even better if you give to the created glyphs temporary substitution values as letters, numbers and pairs of them. After using the glyphs for a while for writing simple substitution codes create new sets of glyphs and practice with encoding again. You will need this ability to change the meaning of glyphs depending on context because modifiers will be changing the meaning of the glyphs – do not try to fix some glyphs to particular symbols (TS) permanently, it won't be a strong code. Some commands can be fixed to specific glyphs, characters and/or additional signs, and some modifiers as well can be fixed, but definitely not the alphabet, numbers and not even punctuation marks. The less fixed cryptic symbols the stronger the code. Meaning of a strong code is coming from combination of its elements and context. Nevertheless, for training and developing your style of writing you can use temporary substitution.



12 Sharing and personalization.

Obviously you can only share encoded messages with someone who knows the encryption method. All the parties with whom you'd like to share messages should decide together about the level of complexity of the shared code. The longer the messages you need to share the more complex must be the method. One of you must take up the task of creating the shared code.

No-one of the parties should take his/her personal code as basis for the shared code. It is possible to have different keys for sharing while using your own method, but for the security is better not to go that way -- the systems of glyphs, characters and separate signs must all be created from scratch, also the matrix and the rules of access. This is quite a complex undertaking for everybody to be happy with the result, thus the encryption method is better suitable for personal use. It's especially true regarding to the work input and risks involved – you can never be sure that one of you isn't careful enough in handling the code. If the code gets compromised you'll need to create a new code again.

If you do need to share something with handwriting cryptography then it's better to create quite a simple set of glyphs, with clear characters and modifiers, and flexible matrix. Glyphs must be simple because several people will have different handwritings with different attention to details in different parts of the script – you need to make sure that all of you will understand the code correctly. Flexible matrix means that the symbols and commands positioned there are in such systematic order which is easy to reorder without the need to create it completely again. If the script gets compromised then you reposition the blocks in the matrix, change the values of some glyphs, characters and signs, and change some diffusion patterns. This way you won't need to create again the whole new system of encryption, only let other parties know the new values, at least temporarily until a new code with deeper modifications has been completed.

As using simpler code provides less security it's advised to change the code after some time of use, depending how much data you share. The more encrypted data you send through internet to other parties the more often you may need to change the code (or at least some values in it), so as no critical mass of information will be out there for cracking the code by cryptanalysis.

You can personalize the code which you share with several parties for use with only one of them, or only one smaller group of them. You only need to create couple of modifiers and switches which will be particular for that person (group), applying specific diffusion patterns. By applying a new set of switches with corresponding encryption keys, the other parties won't be able to read the code unless they use sophisticated computer programs to crack it, but in that case your code is compromised anyway by human factor. In less critical cases you can differentiate the code between several parties using exactly the same method with each, only having different matrixes of real symbols particular to the groups of people you deal with.

As you should never keep your method written up somewhere permanently (you create the key parts of your code in your mind or destroy the keys after memorizing) there's no easy way to share the system with other parties – they need to take some time to

understand the shared code you've created and to learn it. The process of teaching and learning by itself may compromise all the work and effort you've put into securing the script. For sharing is better to use simpler methods of encryption which are easy to change at any time when there's a suspicion that it has been hacked or the critical mass of data is beginning to accumulate.

An important part in sharing your encryption method with others is to hold onto basic precaution rules for sending encrypted messages, which would make attempts to decrypt them by unwanted parties unsuccessful. Clearly that wouldn't be a good idea if someone sent you an encrypted message by e-mail where to meet and you write a non-encrypted e-mail to others saying where you all will meet. you should use either other ways of communication or to have a fair time delay before you forward the message, so the two messages - encrypted and non-encrypted - wouldn't be clearly connected. And definitely you should always use different words for saying the same thing openly what was said in the encrypted message, as much as technically possible.

Another unique feature of the handwriting encryption method with hieroglyphs, uncommon in other cryptographic methods, is the possibility to completely change the meaning of already written cipher text, if that becomes necessary. But that's possible to apply only to relatively short and simple encrypted messages, not because it's impossible with longer cipher text but because it is difficult. For instance you write down some data and someone demands you to reveal its meaning. Then, if you're trained enough in the method, you will be able to make up in your mind the simplest matrix with letters and numbers only, and to show that the encrypted message means your name or whatever else, proving it on the spot. The encrypted script itself (by the way it looks) won't reveal the complexity or simplicity of encryption.. all you need to do is to apply your newly made up matrix to a small number of data points in the cipher text, while "revealing" that all of the rest is just a noise.. and if few of the symbols won't fit in then that could have just been an "error of encryption". That's unlikely that you will need to go to such a length as to prove a different message instead of a real message, but if that's a possibility then you can have a simple matrix ready and to encrypt into every longer cipher text a bit of non relevant information, doing it in parallel with different cryptic symbols.

12.1 Sharing with one person.

Obviously, first you must make sure that the person you want to share a code with is capable of understanding the method of encryption. You can arrange a short training session with the most basic level of the advanced encryption method (chapter 17.1). It takes five to ten minutes to create. But you must verify the capacity of both to understand the handwriting of each-other. After you have successfully encoded-decoded test messages to each other you can agree in which level of precision to detail you're going to create the code, more complex one.

If the person you want to share the code with is just a temporary 'contact person' for exchanging confidential messages, then it's perhaps too laborious task to create anything more complex than the most basic method. Especially when the messages to exchange are short and not of a significant strategic value, like dangerous to your life,

freedom or property. If they are, then creating a strong code for an exchange of information is worth the input of time and labor, rather than relying on a weak handwriting code or even worse -- a weak computer encryption.

Remember, the most basic code can be changed out any time for another most basic code. If the messages are short, like only specific date and place to confirm while all the other details have been discussed in person, then that's the best way to go -- there's less chance to make an error with the most basic code. Another story is with a long time relation, be it with a friend or a business partner. Especially when you're far apart. Then you must consider creating a more secure code than the most basic version, but it's a good idea to test your confidential correspondence with the most basic level first.

After all the differences in handwriting have been settled then one can take up the task of creating the more secure code. Also both of you can create a code and after the exchange of the keys to the codes in a personal meeting the two different codes can be used for different occasions.

12.2 Sharing with several people.

Again, first you must make clear how long time and for how big data sets in length (in total, over time, using the same code) you're planning to use the encryption method. And of course also for how serious issues in sense of confidentiality -- this determines the level of complexity of the code you will create for sharing with several people.

Mostly the procedure is the same as while sharing with one person only, with a little distinction -- you may want to add to your code different sets of cryptic symbols for each party, which means that the party who accidentally gets the wrong encrypted message will discard the 'wrong' cryptic symbols as noise, and as the result won't be able to decrypt the whole message.

This method of encryption allows to send messages to all of the involved parties in exactly the same cipher text while each of the parties can receive specific messages meant only to them, using a set of cryptic symbols others won't be able to read. All the other parties won't see anything wrong with the entire message, decrypting only the general message meant for reading by all involved parties.

12.3 Temporary sharing.

When sharing the code between several people there may happen situations when you need to share the code with one of the parties only temporarily, while excluding the temporary party later from the capacity to understand the secure communication.

For not creating entirely new code for these occasions you must trust the person in general, simply having no need to disclose every secure exchange of information. For that purpose you must have for the same matrix a different layout of symbols (TS) as the new key. The cryptic symbols and the rules remain the same, so all the permanent parties will have an access to the code, while decrypting the messages in a usual way, simply with the different positioning of symbols in the matrix.

This technique of only changing matrixes can be used for any arrangement of

involved parties of secure communication while all of them know the general set of keys. For the parties not involved in a given communication it will be hard to get the encrypted messages because the excluded parties won't have the specific layout of symbols in the matrix and they won't have the specific message neither. Thus quite a large number of trusted people can use the same code with no need to create yet another code for a different message to different people -- only one part of the key can be changed, and that can also be only a temporary key for a person temporarily involved.

12.4 Urgent sharing.

For an urgent sharing of a new code when an old code has been compromised, when there's no possibilities to deliver the entire key to the new code confidentially in person, then there can be made different modifications to the compromised code. The simple modifications can be made to the positions of real symbols within the matrix, moving around entire blocks, and within the glyphs, moving them few steps back or forth. The specific modifications are possible to explain in words over a secondary channel, not by which the cipher text is delivered, even by phone. If you're wise enough and preview such possibility then you can agree in advance with all the involved parties which changes to the code will be made in case of an urgency or emergency, but never use these modifications until there's a real urgent need for it. Then you spare yourselves from possible misunderstandings by verbal explanation of the modifications over a phone.

After a code has been compromised it is safer to create a new code, of course, but in certain conditions, in case of an urgency, an old code can be used for delivering short messages using the modified keys. The modifications to the positions of the glyphs and to the symbols in the matrix accumulate into a quite a significant change in the code and in case of an urgency it may temporarily help you out.

You can have several most basic codes ready for such cases, numbered or given code names to them, but that implies that the keys of these codes must be somewhere ready to use, not only in your memory. It will be safer to use urgent modifications and deliver them verbally than relying on a key that has been lying around somewhere, in places of all involved parties.

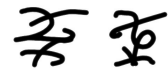
12.5 Key delivery.

Whenever you have the chance to meet the other person to hand over the keys to the code directly, that's the safest way, if the person memorizes the keys to the code immediately and eliminates all traces of the keys. If that's impossible, if you must send the keys using help of other people, for diminishing the risk of the keys being disclosed to unwanted parties you can send them in separate parts through different people.

For delivery you can divide the keys into so many separate parts as necessary. The different keys themselves can be divided -- the matrix can be divided and each part of the matrix can be sent to the other person by a different route. You can also send false parts of a matrix, which won't be used. In case of the most basic version of the code the vertical set of glyph parts can be sent separately from the horizontal set of glyph parts, and the

additional symbols (modifiers) can be sent separately. You can also send the parts of the key by regular mail service using special envelopes. Or even better: send some parts of the key by regular mail and some parts by a person you trust.

All these precautions are symbolic, because if someone really wants to know what's going on then your house can be bugged with hidden cameras -- the technology is in such a level now that even if the handwriting cryptography is needed in certain cases, no encryption will ever provide total security in exchange of information. Your cryptography method can remain secure for ever -- it will not need to be cracked to figure out your activities and most probable intentions. There are ways around all the encryptions to access the decrypted information directly, when the stakes are sufficiently high or the curiosity is sufficiently big.



13 Securing the method for recovery.

To secure your encryption method against problems from possible gaps in your memory, especially after not using your code for a while, you must have all parts of the code organized in a clear order. The code can be safer if all of the cryptic symbols are in random order and memorized as such, but what the code is worth if you can't remember your own cryptic symbols after a while. Definitely the system of cryptic symbols cannot be made completely in a logical order, that if few of the symbols are known then all the others could be determined. Rather the elements of the method must be a compromise between total randomness and total order. The cryptic symbols must be created in small logical blocks while the blocks themselves should better be not related by the design of the symbols. Each block of the symbols must have enough logically ordered symbols in them so that in case of forgetting one of the symbols you can think of another symbol within the block, and after recalling it you can determine the design of the forgotten symbol.

Same with the real symbols in the matrix -- these must be ordered in small blocks where all the elements of the block are in certain order which would be easy to recall if one symbol is forgotten.

In case of the rules of access of the real symbols in the matrix by the cryptic symbols there's no real need to use certain order, because you will be using the rules any time you encrypt and decrypt -- the rules are difficult to forget. With the design of cryptic symbols and with the location of real symbols in the matrix it is possible that you won't be using one particular symbol for so long time that it may get forgotten, thus certain order must be established. If there are some rules of access that you rarely use then you better memorize the exact number of rules you have created and repeat the rules in mind recalling each and every rule, from time to time.

Repeating every element of the code, be it all of the cryptic symbols, all of the real symbols in the matrix, or all of the rules of access, must be done regularly, if you wish to never encounter gaps in memory. For that all the elements of the code is better to create in blocks of certain size which you could easily remember. The repeating process of all of the elements is much easier when you know exactly how many elements there is in each block of your system, then there's no worries that you will forget to repeat one of the elements. Repeating itself must be done not too often that it will become boring and you will abandon the cryptography. Rather challenge yourself -- after a good memorizing do not think of any of the elements of the code for a while, even for few months, and then try to recall every single element without errors. You'll be surprised how good your memory actually is in recalling the things you yourself have created.

13.1 Recovery of encryption method.

Against the loss of certain elements of the encryption method which you have created it is a good idea to have written down some irrelevant data, but which is logically continuous, using the encryption elements which are rather rarely in use. Then you can check the sample record and to recover the forgotten element of the code at any time.

After that had happened couple of times then you won't even need to look at the sample script anymore -- you can then think of it in your mind and the rarely used encryption element will be in your mind quite fast. If some of the elements has had taken very long time and much effort to recover then write an additional sample code using that hard to remember element, then the next time you will recall it faster, if forgotten again. These things do happen, because it's unlikely that you'll be using every single element of your encryption method whole the time. So, thinking of the possible troubles you can protect the integrity of your code with the samples of cipher text.

13.2 Recovery of encrypted data.

To make sure that your encrypted data is always recoverable you must create the method which isn't cryptographically dependant over long distances in the cipher text. Just a few cryptic symbols in length of dependance is enough for a secure code against hacking and yet secure against loss of data. If your symbols of encryption modify each other over many elements in distance then just one error may mean the loss of entire word, while in shorter interdependence within the cipher text will mean a loss of just a few symbols, in case of an error of encryption. The method which I'm presenting in this book does not require the code to be severely interdependent over long distances to be secure. The method was developed keeping in mind that people do make errors, thus the errors must not affect the recovery of data. To secure yourself against the loss of unrecoverable data -- like a digit of a code or of a phone number -- you better write it down twice, encrypted in different ways. Do not be over-confident while encrypting the data that in case of loss may cause you troubles. Rather take the trouble to encrypt the important part few times differently to be sure to recover the code/number when there's an error in one section of the cipher text.



14 For developers.

There are several methods of encoding which I have realized as a possibility but haven't paid much attention to:

An easy one.

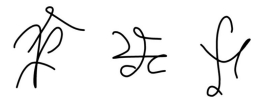
Develop a method of encryption which involves taking encoded symbols into separate parts and hiding them in a script or in an artwork. While hiding the symbols in an art do it the way that the parts of symbols wouldn't be easily spotted as some kind of hidden message.

A complex one.

Add to the script a second layer of information by controlling already used symbols in the script. You can do it by combining parts of the symbols within the script or by changing their values, and pointing them out for the second layer encoding. The second layer must not change the message in the first layer. It's possible to accomplish by utilizing some of the noise tools as the values of the second layer, without visual damage to the design of the script. You can also come up with an original solution which I haven't realized yet.

A challenge.

Develop a method of encryption where the hieroglyphs contain encoded verification against an error. It means that if you encrypt an access code or telephone number and it doesn't work after decryption, then you will be able to determine which symbol was encoded with an error in it, and consequently will be able to deduce the possible set of symbols one of which will be true.



15 Application.

The method of encryption I developed can be applied to any language, without exemptions, thus anyone can make use of it. I guess even aliens could use this method to transfer information as long as they produce sound, and even without sound. The method is based on representing one kind of information with another type of information, thus even music can be written using this encryption method. But the main application, in my opinion, is keeping your boring moments in life filled with an interesting and useful activity and keeping your brain active for the health of it.

There are countless situations in life when people get stressed because something is happening just too slowly, forcing them to wait. I filled many of those moments thinking about the cryptography – either developing a system in mind or repeating the symbols not to forget them. In some occasions, when I was tired of working on a new system, I challenged myself to recall a discarded system which I have created long ago, and very successfully, being surprised how well I have remembered things I thought to have long forgotten. I wasn't able to do it in the beginning but after years of training my capacity to memorize has grown quite a bit.

My life has never been boring and for me the encryption was never a means for really hiding some information.. I don't keep a diary and there's no really a point in encrypting an address book – the data like phone numbers is possible to find out by other means if someone really wants to. After I did it once, for training purposes, I have never again encrypted my contacts. I am a developer.. I had interest in creating ever more sophisticated methods of encryption rather than really using them. It doesn't mean I won't, because now I have my own method which I'm happy with, but previously there was no point to encrypt something for a long time storage because after creating yet another code I didn't use the old one anymore and purposefully tried to forget the old system which I didn't use, not to confuse with the new one.

Creating your own writing method is an excellent memory training and on top of that it trains you in multitasking. For example if you stay in a queue in a shop exercising cryptography in your mind you cannot just lose your attention falling deep into thoughts – you must pay attention to everything around you, not to let have something stolen from you. You can clean your house and train your brain the same time – what an excellent application in my opinion. You can achieve the level when you walk around in a city for your daily stuff and develop cryptography in mind at the same time. I wouldn't advise driving while doing so, unless you drive a car in a computer game. In any case, after some time of training you'll be able to have complex tasks going on in your brain without affecting your attention to surroundings. I would even claim that you'll be reacting more adequately to the situations around you when your brain is actively working on something than if you hang around on 'autopilot'.

When it comes to memorizing and recalling it's quite clear from recent studies that human memory is specific to states. It means if you learn the glyphs in a quiet environment in home and try to recall them in other places, often in quite complex situations, you'll present to your brain an extra challenge. With changed states (different surroundings, environment) between learning and recalling of learned information you'll

be creating and strengthening new paths in your brain (especially for finding connections in seemingly unrelated information), giving your brain extra capabilities which will help you in many ways.

Memory training helps you in keeping the plasticity of your brain and in reducing the subconscious stress about wasted time while forced to do nothing in certain social circumstances. Thus creating your method of cryptography is not only the means for hiding private information, but also a great training of your brain for being more capable and less stressed in different situations in your life. The more complex processes you'll be able to handle in your brain the longer your brain stays healthy and less stressed you feel in your life. By the way, it also helps to forget things you don't like to think of. When you push your brain to the limits while thinking about something very complex you'll forget about the things that bother you. Forgetting unpleasant situations of life works even better if you don't have to pay attention to your surroundings, when you're alone, trying to fall asleep for example -- if it happens that you can't fall asleep and some annoying thoughts are bothering you then just concentrate on your personal cryptography method and the next morning you'll wake up happier and smarter.

Generally speaking, by creating your personal encoding method you'll learn to do necessary active thinking while busy with other things, without the need to sit down to do some thinking. You'll gain in your active life time.



16 Example of glyph parts.

There are countless ways to create glyphs with different parts, so here I'll give you just an idea, not a rule for following. It's always better if each and every code is created unique, not copied from the book or anybody else.

An allegoric comparison can be made with music -- if someone gives you an instrument and you're free to play then you don't need to play the same melodies that others do, you can compose your own, but you're still bound by the principles of music and by the possibilities of the instrument. But if you look around then you're not bound to the instrument neither.. you can find another type of instrument. And if you're talented enough you can even build your own original instrument. But no matter what instrument you have you're still bound by the rules of the music. This is like the boundary of the cryptography, the rules you must follow in order for the cryptography to accomplish the task of encoding. But you can use whatever suitable instruments.. whatever hieroglyphs or other symbols that cannot be even called hieroglyphs, as long as you're able to encode data with the tools you've chosen or created your own.

Here are some examples of different parts of glyphs..



In the samples above the base glyphs are black and data points are in several colors. The gray part in the end of the glyphs is an example of connecting glyphs in different ways. The different connections between glyphs can carry specific information or may be just a design element, a noise.

Here's an example of a cipher text how it would actually look, prior to showing you the possible different parts:



The different ways to connect the glyphs are here hard to notice to those who don't know where ends the base glyph and where begins the connection part between glyphs, which can actually be a separate cryptic symbol. Without showing you the separate parts in different colors it would be impossible to determine (for those who don't have the key) where are the base glyphs and where are the attached parts.

An example of separate parts in writing the glyphs together..



Now lets have another look at this. In case of different parts of the similar looking glyphs chosen as base glyphs, and therefore different ways of encoding information, the separate parts of the similar cipher text may be as this..



Notice, the attached parts weren't possible to distinguish without knowing the base glyphs, and that's just one of the easiest examples I have given you. During design of the encryption elements in your code you must consider all the reasonably applicable ways to confuse possible adversaries, but not going to such length as to confuse yourself. There are more tools and ways to make the code sufficiently strong, not only the style of writing and visible/invisible attached parts of the glyphs.



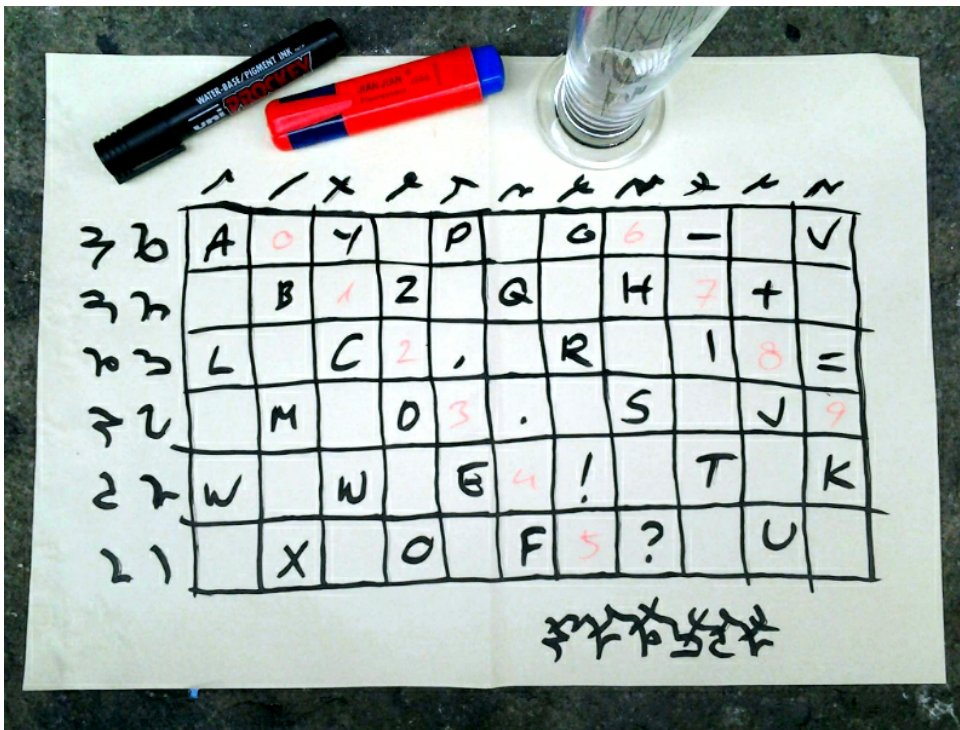
17 Examples of code creation.

The examples are created as simple as possible for easy and quick understanding. While using these examples as a template for your own method you should alter the code significantly, with original keys and symbols of encryption.

17.1 Basic code.

Here I show you the most basic code, created during a street performance in five minutes, to demonstrate how easy it actually is to create a simple but relatively secure code (if little used), when you're familiar with the principles of the code creation.

Here's the matrix of symbols for the example of the most basic code..

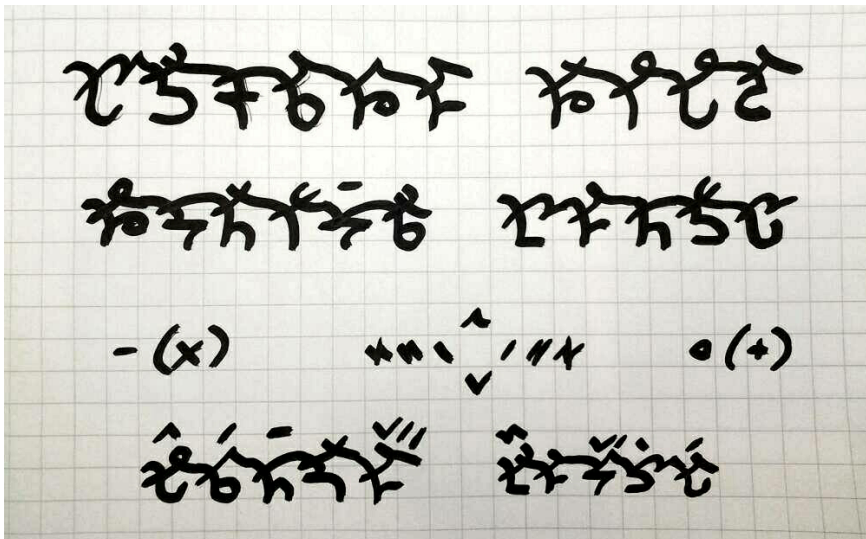


The encoding method is basic, but still advanced method of encryption. There's possible to make the method even simpler, without a matrix of symbols, with a simple substitution of symbols and lots of noise included. That would also be secure if the level of noise is great enough, if noise is similar enough to parts of the script which carry information, and if the code is very little used -- for example encrypting certain codes only, without long text messages. But making the whole thing into a simple substitution method that wouldn't be an advanced method of encryption anymore.

Thus the most basic advanced encryption method is the example given here. By adding to the matrix more parts of the glyphs, and inside the matrix repeating several letters/numbers in random order, it will make the code more secure while the level of complexity will remain the same, still basic.

No matter how many different sets of the parts of cryptic symbols you add to the sides of the matrix (realistically taking), and thus increasing the number of possible combinations to access information in the matrix, it won't make the code so much stronger as being impossible to crack it if you encrypt very long messages. Adding more different parts of the glyphs will only buy some time for you -- it will provide some extra security only for as long as there isn't critical mass of the cipher text out there to begin an attempt to crack the code. You can use the overriding rules -- like cancelling previous symbols or modifying them with some assisting cryptic symbols -- or other tricks, but it won't be as effective as using other techniques of encryption which I will explain later.

Use the matrix of symbols above to decode the script:



This kind of short text (above) would take about a minute or two to write. Some additional symbols are also added, to make the script in the last line a bit more complex. The dash (x) means that the glyph is discarded, the circle (+) means the symbol is repeated, forward slash means that in the matrix the actual symbol is one step to the right, double forward slash means two steps to the right, and so on respectively and in every direction correspondingly. The example is purposefully made very easy to understand by using clear and logically arranged additional signs. In your own code the additional signs and also the glyphs themselves (of course) shouldn't give any hints to the way they're used in the encryption process. I remind you that in every code there should be added, during encryption, some symbols which look similar, but aren't listed, including additional signs which mean nothing, for noise, for making the resulting cipher text safer.

Stepping back a bit about uselessness of the most basic method. Even though it's weak, this method wouldn't be in the book if it had no value at all. If you're very far from intention of becoming an expert in my method, you can still learn the principles of this

very basic method for your needs, and be happy with that. In some cases this method can be used safely, temporarily, to exchange few short messages, and then to discard the code, never using it again. If there's a necessity to use a code again then a new code must be created (at least it's advised for security reasons), with different dimensions of the matrix, different order of text symbols and better also a different set of the code symbols, not to confuse yourself and the receiver of the messages.

It's especially advised to create a new code for new encrypted messages when the encrypted message sent previously has been decoded and the message has been discussed with third parties, or the message is related to a known event. Such information, in presence of a copy of the cipher text, may lead to deciphering of the code by unwanted parties, and therefore also to deciphering of the following messages sent using the same code.

When all parties willing to communicate some confidential data are familiar with the principles of the most basic advanced encryption method, then it's entirely sufficient if one of them creates the encryption matrix of the code and gives a copy to others on a personal meeting, without discussing the code verbally (the conversations can be recorded by unwanted parties, thus discussing a code verbally isn't wise). Then the encoded short messages can be exchanged over the internet or by regular mail without worries that third parties will guess the meaning. But again, even the actual symbols inside the matrix must be placed in a random order and some of them repeated several times, then no computer algorithm, however advanced, will ever be able to guess the meaning, because in short exchanges there's not enough data to discover a pattern in the code and to make any sense of the symbols.

When dealing with newly created codes, due to lack of practice, as yet another safety precaution is to write the message once, and then write it again without visible writing connections within the glyphs -- the points where some parts are attached later should merge into the lines without visible attachment points or line breaks, not to give away separate parts of the glyphs, where possible of course. In the example of the most basic code there aren't any additional attached parts, but if you do add some, even as a design element or noise, then the cipher text should look smooth. You should not give an easy way to discover separate parts of the glyphs for an analysis, for an attempt to crack the code. And of course first verify the ready cipher text for any possible errors -- only the final product must be written again with a smooth look.

17.2 Medium level code.

It's quite a nuisance for myself to be a perfectionist, in almost everything. But with cryptography it's good, because it will never be perfect and thus it will never get boring -- there's always a possibility to make it better. And it can be quite addictive.. sometimes I can spend whole night lying in the dark awake while rearranging symbols in a matrix and redesigning cryptic symbols, all in my mind.. for a better design, better grouping in matrix, and so on.

17.2a Matrix of symbols.

The matrix of symbols for the medium level demo code I have chosen three-dimensional: 9x9x3, but it also has switches and modifiers added, so the matrix itself, without an explanation, won't tell you the whole story.

The matrix of the code I created without making any notes.. half of a quiet evening and half a night of work, from the beginning till completion. I kept it in my mind for several weeks, prior to writing it down. You see, after years of practice it's way easier and faster to rearrange symbols in mind, when necessary, rather than doing it on paper, and by the habit of safety precaution I never use a computer to create a new method, even a demo version.

I remember how in the very beginning I was using paper notes a lot, for designing symbols and matrixes, and to some extent up to just several years ago. I have been using notes quite extensively for over ten years. (You don't need to use paper notes for such a long time -- I have been developing new and better methods while you already have the information you need to create your own best method, with the help of this book). At times it was creating loads of paper filled with all kind of hieroglyphs in every corner, because multiple deletions and rewritings made some spots unreadable, thus I needed to draw lines and arrows for interchanging parts and pointing where some symbol on the side of the paper should be in the matrix. And when all got too messy then I had to rewrite the whole 'ready' matrix in a clean way, to continue whole the process of editing. I never kept any notes on my cryptography for more than few days, memorizing all I could and destroying the notes. Most paper consuming work was done when I still used sets of different whole hieroglyphs.. tens of sets and thousands of glyphs. That was before, now I know better.

After the initial outline of glyphs is designed and tested on paper, doing the further editing in mind is way more easy to make sense of -- by working in mind the matrix and symbols are always clean and clear. But of course you need a bit of practice before getting used to seeing the whole picture in mind, and remembering every change you have made several steps back. For the purpose of not losing some data you also need to remember couple of old "completed" matrixes, temporarily, together with the latest one, then you can review an old matrix (all in mind) for any symbols that may have been forgotten in the latest matrix and the list of cryptic symbols. It isn't too difficult because you don't need to remember entire old matrixes but only those parts that have been rearranged.

When the matrix of symbols for the medium level demo version was ready, I wasn't still sure if it's better to publish only the code symbols (CS) and make the readers to guess the meaning (to guess the arrangement of real text symbols (TS) in the matrix). After giving a bit of thought from the point of view of someone who hasn't practiced with the method for some time, I realized that it would be too tough an exercise and hardly anyone would be interested in doing that. Indeed it may look terribly complex but in fact all the symbols in the matrix are in clear order and easy to remember, once you understand the order they're placed into.

I suppose that to guess even a simple matrix with all the symbols without having ever seen the matrix it could be tough even for the analysts in the NSA. In any case, designing a complex matrix, when the cryptic symbols are complex and well designed as well, is a better option. Then it will be impossible to crack the code for sure.

Here's the simple 9x9x3 matrix of real symbols for the medium level demo version of the advanced handwriting code. It's written in two dimensions. The top matrix on the picture is in normal use, the symbols in the matrix below (the second one on the picture) are used after applying a modifier. The horizontal line of the matrix includes the third (x3) dimension..

a a A	b b B	c	j	κ	l	s	t	u
d	e	f	m	n	o	v	w	x
g	h	i	p	q	r	y	z	ϕ ϕ² ϕ³
1	2	3	4	5	6	7	8	9 ² ³
- + = ÷ * # / % °	' ' ` , j ^ . : ..	~ -	⊂ ⊃ ⊆ ⊇ ⊈ ⊉	⊊ ⊋ ⊌ ⊍ ⊎ ⊏	⊐ ⊑ ⊒ ⊓ ⊔ ⊕ ⊖	⊗ ⊘ ⊙ ⊚ ⊛ ⊜ ⊝	⊞ ⊟ ⊠ ⊡ ⊢ ⊣ ⊤	⊥ ⊦ ⊧ ⊨ ⊩ ⊪ ⊫ ⊬
⊮	? ⊮	⊮	a?					
⊮	! ⊮	⊮	a()					
H	'	⊮	a a A	e	i	o	u	y
Γ γ	Z ζ	I ι	M μ	O o	Σ σ ς	Φ φ ϕ	Ω ω	Ϸ ϸ Ϲ
B β β	E ε ε	Θ θ ϑ	Λ λ	Ξ ξ	P p	Υ υ Ϛ	Ψ ψ	Ϻ ϻ ϼ
A α	Δ δ	H η	K κ κ	N ν	Π π ϖ	T τ	X χ	Ϙ ϙ Ϛ
	≡	*	\ % ∞	' ' >	√	:	≈	
) }]	" >	>	i ï					⊂ ⊃ ⊆ ⊇ ⊈ ⊉

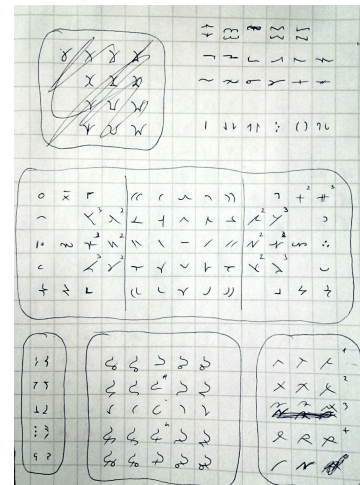
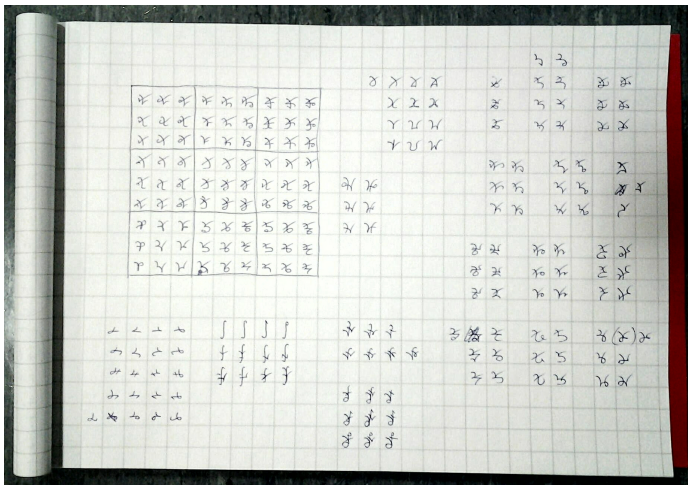
As you can see, many places in the matrix are vacant, possible to use for more symbols and functions when needed. Vacant for use are the squares in the last three lines of the second matrix (the matrix after applying a modifier). The other empty squares are mostly left unwritten not to make the matrix too messy for understanding -- the symbols not written out are understood by logically applying the order of alphabet, by repeating

the first symbols which are present etc. Only necessary symbols have been written out, for clarity.

17.2b Encryption symbols.

These are the symbols from which the hieroglyphs are built. Separately they don't mean any letters, numbers, functions or else. They are the position markers, modifiers, etc., to access the real symbols (letters, numbers, punctuation marks and so on) in the matrix of the demo code.

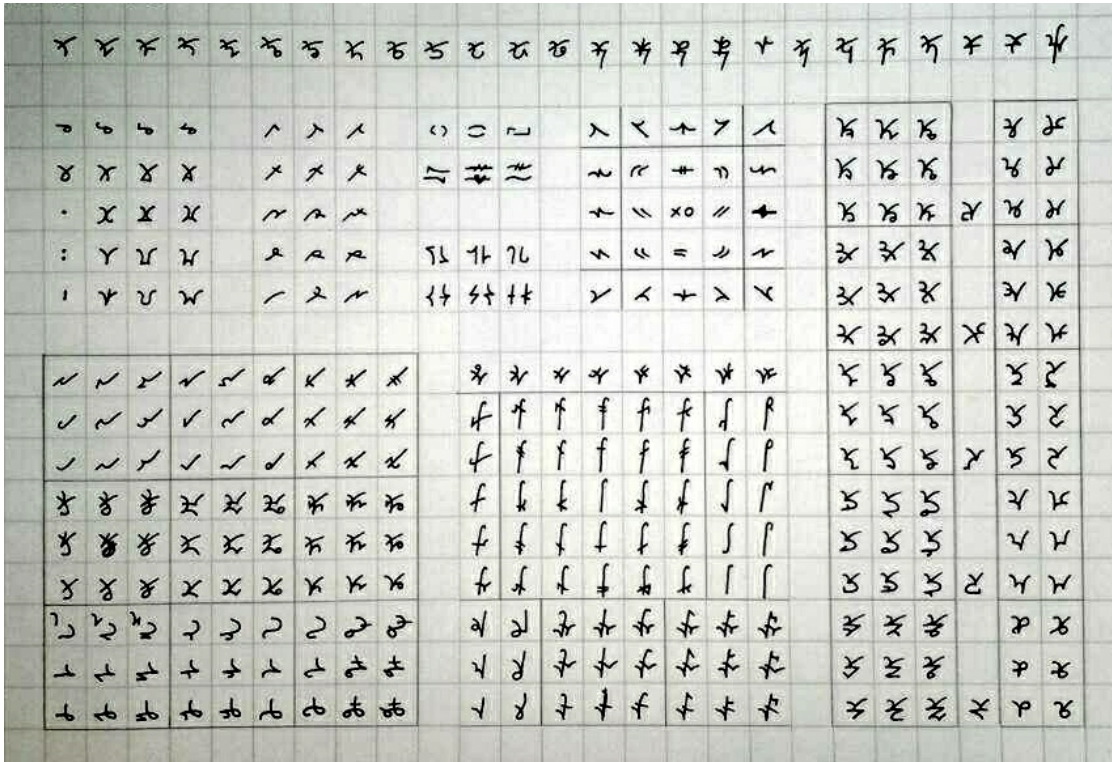
Here are the pictures of the work in progress, on the encryption symbols..



These hieroglyphic symbols aren't picked by just a random writing. I wrote them down in the way that has proven to be the most convenient and practical for the advanced handwriting cryptography. There are countless trials and errors behind before arriving to this type of cryptic symbols as the best, in my opinion anyhow. But this is definitely not the only possible design of cryptic symbols.

I did the research of different writing systems more than once. First when I began taking my hobby seriously, when I was still a kid. I had to go to libraries on these old times. Later I tested several times with different writing methods, till I found that the current way of writing suites for the demands of cryptography the best. A few years ago I decided to make another research, downloading from internet hundreds of pictures of different writing styles, from wikipedia and dedicated websites for languages, but then my computer crashed and I lost whole the work. For over a year I couldn't afford to buy another computer, so my attempt to find a better way of writing was over this time. And maybe it saved my time actually, who knows. Just for this book I did yet another research, to give you some examples which you find on the pages 'to print/copy' (chapter 28). These are not suitable for encryption, obviously, the samples simply can make creation of new cryptic symbols easier when you get stuck around similar looking glyphs thinking that this is it, nowhere to go from there. There are many ways to go in developing your own writing, but some refreshment of mind may be needed at times.

The cryptic symbols to build access hieroglyphs for the matrix of the demo version of the medium level code, after completion..



As a reminder, you should never use any existing classical symbols in your handwriting encryption method. Even if you never use those existing symbols as a simple substitution, only in different combinations and in different levels of writing (base, on top, below) these will be too easy to recognize, to enter into a computer program and let the algorithm to figure out your encoded messages. You wouldn't like to make the life of cryptanalysts too easy, would you? All the symbols must be created different from those existing in the lists of computer symbols. If few happen to be similar not a problem, you cannot verify every single symbol against computer recognition when scanned in. But the trend must be your original way of doing things, then it will be hard to crack the code – it will demand enormous labour from cryptanalysts in making sense of your code, without having the keys, one of which is the list of cryptic symbols.

17.2c The creation process.

So here's how the medium level demo version of advanced handwriting cryptography got created..

Step 1: First I decided the dimensions of the matrix of real symbols, based on how many symbols (TS) I was planning to use. (The dimensions of matrixes may vary largely depending on how many alphabets of different languages you will include, how many sets of numbers you include for your encryption needs of financial transactions, etc.). Then I

placed the real symbols into the matrix. For me it's easy as I have my own method in mind, so I just needed to change the dimensions of matrix, to select symbols leaving out all that's not necessary for the demonstration and put the selected symbols in an order which could be easily remembered. While you don't have the 30 years of experience with it, and sure you can't do it all in your mind never having worked on it before, better use a paper when you create your code. I can't stress it enough that never use computers/tablets for the creation work.. and while working out your method on paper you must memorize the symbols and destroy the paper. This is one part of the 'key' that nobody else, apart of people you share the code with, should have ever access to.

Step 2: As the second step I created a set of code symbols, covering each square of the 9x9 matrix, leaving the third dimension for playing around with modified symbols (using attachments and separate modifiers). That was done in few hours -- just wrote on paper hundreds of symbols that first came to my mind and selected those I liked. Each cryptic symbol's meaning is easy to change with additional signs. By default any of the symbols on their own should never be attached to any real symbol (TS), even though it is possible to set default meanings. By default (without a modifier) a symbol could only be related to a previous modifier used with a previous hieroglyph, or using other related definitions without a fixed or frequently repeating meaning.

After that was done I decided that this doesn't bring the demo code up to a medium level, thus the next day I wrote down many more symbols and selected all those I preferred to use for the demo version. Also I decided that the cryptic symbols won't be covering all the squares of the matrix in the exact order, then you can't use them for simple substitution of real symbols by a habit. If the cryptic symbols are not defined by some rules before they will mean anything they can be used as noise to distract cryptanalysts from track. The selection of additional cryptic symbols took few more days, few hours each day -- mostly the selection time for the best design. In the step 2 the exact rules to access real symbols weren't decided yet -- the rules I was planning to use were in my mind in general, which helped me to decide the amount of the additional cryptic symbols.

After I had chosen the necessary amount of cryptic symbols for applying the rules, I proceeded with defining the rules. For you, without prior practice, it is better to go straight to the step 3, creating the rules of the code first and only then create the necessary amount of cryptic symbols for applying the rules. But there's nothing wrong with it if you create a large amount of additional symbols first and after defining the rules discard those symbols you like the least. You can also use several symbols for the same meaning, which only makes the code safer still.

The list of cryptic symbols is the second part of the 'key' (after the matrix) that must be secret. All the other symbols which are not in the list are used as noise, thus obviously the symbols which carry information must not be known to others.

If you like it that way then the new base glyphs which you haven't included in the list of symbols and will design on the go, can be used as attachment points to extensions and separate symbols, which will define previous or following base glyphs. Also, this option can take place if a specific additional symbol is used, or specific placement and/or combination of additional symbols (up/below or both).

Step 3: After the matrix with all the real symbols had been decided, main cryptic symbols (base glyphs) designed and sets of additional cryptic symbols created (extensions, separate symbols), the next step was creating/applying different rules to access information in the matrix by the cryptic symbols in different combinations between them. And of course to conclude everything with creating an example of writing in the code, to show the final result.

There are so many possibilities to apply different formulas to access data in the matrix by the cryptic symbols that I only needed to choose the ones I liked to use for the demonstration. That's what I was doing, choosing carefully simpler rules first, not to mess up anything.

Remember that the rules can be added/changed on later times by applying different (new) modifiers, but you can never discard any previously created rules, you can only increase the complexity. At least you cannot discard any older rules if there are somewhere in your notes some older encrypted texts with the rules used. If you're still in a practicing mode without permanent encrypted notes with important information then you can change whole the system. The issue becomes then that you may drive yourself into frustration because the code will never be perfect after some practice.. you'll be constantly finding new ways to improve the method, thus your code will never be completed. It may happen that after yet another set of changes you cannot decode some of your older notes even for a practice, to find any errors and improve your skills. As a solution to this dilemma of stable code versus improved code is to take both options. Use one code as it was completed in some point and start using it without any further changes. The other code then, which you will be developing further till you're sufficiently satisfied, or just for your own amusement never ending the development, must include some symbols (which can be as simple as a dot in the beginning) which always make sure you apply the right code and don't confuse yourself between the two, both during encryption and attempting the decryption in later times.

This is a medium level encryption method for this reason that you can easily see the order in the method. It may already be quite complex and impossible to crack if the matrix and the list of cryptic symbols aren't known to the attacker and the encrypted texts won't amount to a critical mass needed for applying an analysis. In a high level method there are actually less cryptic symbols (base glyphs) and instead a bigger amount of different complex encoding rules, thus the ways of accessing information in the matrix are also much more numerous, increasing exponentially with every set of new rules which can be applied together with other rules or separately.

A high level encryption method is way more hard to guess by those trying to crack the code. If well designed and used with all the necessary precautions then the high level code will be impossible to crack with the most sophisticated algorithms on the most powerful supercomputers, because it all comes down to the input of data, and that must be done manually for the entire length of each similarly looking script around. No matter what the skills and high expertise of a team of professional cryptanalysts they will never be able to guess all the variations of human imagination in creating the rules of a code, even when the separate parts of glyphs in handwriting can be identified. Particularly great time will be having cryptanalysts while trying to crack the scripts which mean nothing, just exchanged between friends for practicing design, while only few of the

scripts will have an encoded information in specific parts marked with specific combination of additional symbols or specific base glyph.

Digital cryptography and most classical handwriting encryption methods require the encryption rules to follow each other, which makes cracking the codes with the help of computers easy -- the attempt of cracking I mean, leaving the bulk of work for computers to do. In this method of encryption the switches of rules can jump in different order of application all over the place within the glyphs and between several glyphs while for you it makes clear sense of what is going on due to the visual input of the many parts of the glyphs simultaneously. A computer cannot read the information like human brain does, with parallel input and processing, giving instant result (an answer) when the necessary requirements are met. For a computer algorithm, with unknown and unpredictably changing order of application of data carriers within the glyphs, without knowing where is the information and where there's a design with no information, it creates an astronomical number of variations to calculate, which no array of computers can handle. That's the power of handwriting encryption using hieroglyphic method.

In this book I am not giving you an example of the creation process of a high level code (I'm not talking here about the highest level method which I'm using myself and is briefly described later), because the medium level code has all the necessary explanation needed to apply the creation process for a high level encryption code. As I said, it must include less fixed cryptic symbols, more changing symbols for a design and noise, and lots of rules to access data in the matrix. In the following chapter you will see that the description of the rules is quite long even for the simplest and minimal set of rules in the medium level.

The rules are actually simple on their own, creating complexity and security of the code in many possible ways of use and unknown to others application. In the mind they're an instant understanding of a process when you decide each combination of symbols (CS) for what they will mean, but it takes a lot of work to describe a rule to someone who can't see inside your mind, how you visualise the hieroglyphs and how you manipulate the data. To create an example of a higher level code will be pointless, because if the medium level code example isn't enough for you to get an understanding how the process works, then nothing will help you. Anyway you will need to create your own unique code which is an interesting and rewarding process, instead of studying in detail yet another example which you can't use.

To achieve an unbreakable personal code it must be done by a good input of personal imagination, not by compiling it by taking examples from others and creating a code on ready templates with some modifications. The method and general security requirements remain the same, but almost all the details of the code must be unique enough not to be comparable with some published examples, which would help in an attempt to crack the code.

For the demo version I prepared cryptic symbols with a clear system of additional signs, to show in which direction the modification should be applied, to access the real symbols in the matrix. With the cryptic symbols there's no hint about what they mean (for those who don't have the list), but the additional signs are too easy to guess. For yourself you must create them in a disguised way, not like I did it in the demo.

Once I had quickly completed the rules of the code in my mind, soon after I had to discard the rules and to begin creating new ones, because I realized that those rules that were easy to imagine in mind would have taken few months to describe on paper, to explain step by step for others. I had to create the encryption rules that allowed relatively easy description. On the list of cryptic symbols I added encircled numbers to the sets of the symbols, also marked some columns and rows in some sets with different letters, for description purpose. Even complex rules by themselves are easy to create -- you decide the rules for yourself and simply use them in the encryption process. But to explain all the rules to others is much more work than creating them.

So here's the list of cryptic symbols again, with added location markers for the purpose of description..

The image shows a large grid of handwritten cryptic symbols. The grid is organized into rows and columns. At the top, there are circled numbers 1 through 9. Below these, there are letters A, B, and C. The symbols are arranged in a grid that is roughly 10 columns wide and 15 rows high. The symbols themselves are various stylized characters, some resembling letters or numbers but often distorted or combined. The grid is used to illustrate the complexity of the encryption rules and the need for location markers to describe them.

[for an exercise of attention to details find two corrected symbols comparing this list to the same list of cryptic symbols without the added location markers. During verification of the cryptic symbols I found two errors and corrected them, and took another photo of the same list with the location markers added.]

Note: A large version of the C-list you'll find in the end part of the book on the pages to print/copy (chapter 28).

Also the matrix of real symbols is easier to describe with additional position markers. I added on the top left the letter R (meaning real symbols), thus in description the position R2'ae'il would mean position of the reverse question mark $\dot{\iota}$..

The image shows two handwritten matrices on grid paper, labeled 1 and 2. Each matrix is a 9x9 grid with additional markers on the top and left. Part 1 (top) has columns labeled 'a', 'b', 'c' and rows labeled 'd', 'e', 'f'. The symbols include letters (a-z), numbers (0-9), and mathematical symbols like +, =, %, etc. Part 2 (bottom) has columns labeled 'a', 'b', 'c' and rows labeled 'd', 'e', 'f'. The symbols include Greek letters (Gamma, Zeta, Iota, Mu, Omicron, Sigma, Phi, Omega, Lambda, Beta, Epsilon, Theta, Lambda, Xi, Rho, Upsilon, Psi, Chi, Sigma, Alpha, Delta, Eta, Kappa, Nu, Pi, Tau, X, Gamma, Epsilon) and other symbols like infinity, percent, and arrows. The matrices are used for advanced handwriting cryptography.

A short description of the matrix of real symbols with dimensions 9x9x3:
 a, b, c and d, e, f are the 9x9 dimensions, each marker letter covering 3 rows or columns; j, k, l are the third dimension (same for each block as shown on the bottom right); the first, fourth and seventh column in part 'a' of the matrix are actually first, second and third columns in vertical dimension (continuing similarly till the ninth column of vertical dimension); part 'a' is covering 9 columns because it includes 3rd dimension as well; the encircled 1 and encircled 2 are part 1 and 2 of the same matrix (marked on the right), symbols in part 2 are applied after using a modifier in position R1'ae'ij -- the half circle arrow with lines up and below; the lines above and below symbols mark commands in the matrix, which change symbols (these are internal modifiers, the modifiers which change the meaning of symbols within the matrix, which can't be done with the modifiers as cryptic symbols, in this example).

Note: A large version of the R-matrix you'll find in the end part of the book on the pages to print/copy (chapter 28).

To clarify the system in the matrix: R-matrix 1 block a-d is filled with alphabet letters from a to i, while the third dimension specifies the same alphabet in capital letters (third dimension third option) or the "same as previous" | a | (meaning if the alphabet was in miniscule writing the the specific real symbol is also in miniscule, if the previous real symbol was a capital letter the this particular one is also in capital writing (the second option of third dimension). This explanation should simplify the picture of how the matrix is written in three dimensions plus the second matrix (R-matrix 2). If you ever get confused about the rules and specifying of a position in the matrix, then refer back to this explanation and you'll be able to get a clear picture again.

I imagine it will be difficult to picture the whole method at first, so for easier following, particularly the rules in chapter 17.2d while studying the method, you will find the same C-list and R-matrix in larger versions in the end of the book (chapter 28), for printing them out if the book is in digital version or for copying from a hardcover book.

When I was writing the position letters on the sides of the matrix I was planning to use three consequent blocks of three columns as the third dimension (marked as m, n, o), but then I changed my mind, using j, k, l instead as the third dimension. That way it's easier to understand the whole system. If I was using m, n, o as the third dimension then the alphabet would have been shuffled between three dimensions of the matrix, providing better security against cracking the code, but that's not the purpose in the demo version -- I'm trying to make the system as clear as possible for learning the method.

With the parts (1) and (2) of the R-matrix taken as a separate dimension it could legitimately be called a four dimensional matrix with dimensions $9 \times 9 \times 3 \times 2$, but for avoiding confusion in understanding extra dimensions I don't describe the rules in four dimensions, calling in a modifier. It's just a matter of convenience in understanding the rules as the fourth dimension can be hard to picture in mind for some readers.

About utility of creating additional dimensions of a matrix: if a specific modifier of a base glyph or modifier of another cryptic symbol is barely ever used, changing only once in a while just a few symbols in the entire matrix and in one possible step only, then creating an extra dimension for the matrix is useless. But if you use a modifier extensively, for example for almost every letter of the alphabet and every number, then creating another dimension may be quite useful for memorizing the data, for adding different ways to access data in the matrix (by that making the code more secure as well) and for creating extra places for the symbols and functions that you may need to include one day for specific needs of encryption not previsioned at the time when you completed your method.

In the example of the medium level code the symbols are positioned in the matrix this way because then they are in the most convenient order and easy to understand, which is far not the most secure way. But here comes the point when you can ask yourself is there any real reason to shuffle the symbols (TS) in the matrix to make their position hard to remember and thus the code prone to errors.

In the most basic advanced encryption method there's a true need to shuffle the symbols in the matrix and to repeat several times the symbols which are more common in the language you encrypt, and to make the matrix much larger than the total number of

symbols would require, to leave many empty spaces for noise. Because in the most basic method there aren't too many ways to encode the data, so you must employ a seeming mess in the matrix to provide a bit more security, even if you use the method for very little cipher text without coming near to a critical mass for analysis. In theory even a relatively short message encrypted with the most basic method can be cracked, when some rare long words are used, by discovering an order in a matrix which will confirm that the deciphered message is not just one possible solution among many other possibilities, but the only solution. If the symbols in the matrix are shuffled randomly then a possible (cracked) solution can not be confirmed as true.

With the medium level advanced encryption method there are already so many ways to access data in the matrix that in an attempt to crack an encrypted message a possible solution cannot be confirmed to be true by third parties, even if the solution seems plausible. Thus to shuffle the symbols around in the matrix will rather confuse yourself more than it will provide extra security. As I have mentioned it previously, to increase safety, you can use different sets of letters and numbers within the same matrix while encoding related data like logins and passwords, names and phone numbers. The different sets of letters and numbers can be kept in an order that is easy to memorize, just in different order from one another. But even that isn't always necessary and you can do it well with a single set of alphabet and numbers within the matrix, because for someone in the capacity to crack your encryption method it's easier to get the passwords and phone numbers by different routes of access, not by trying to crack your cipher text.

17.2d Description of rules.

The following may scare you off from the entire method as it may seem too complicated for everyday use, but that's just the first impression. Similar thought may come to mind to someone after first seeing a musical partition, which isn't making any sense until it is well studied. After learning all the rules just a simple glimpse at the notes will give you the whole picture, logic and harmony behind it, having an instant understanding and allowing to play a musical piece while reading the notes first time. But of course after some practice. So, take one rule at the time, make sure you understand it exactly, not by a vague perception that maybe that's the case but unsure, and when all the rules are clearly understood for what they exactly mean then reading a script written in hieroglyphs will be like a music for you -- after some practice it will give you an instant message in the cipher text, while remaining an unbreakable code for others not familiar with the rules of the code. It's not worth making a code simpler for a lifetime use, as then it won't be safe enough after using just for some short period of time.

In this chapter I describe the ways to encode information for the medium level demo code of advanced handwriting cryptography -- try to picture the rules in your mind looking at the cryptic symbols in their combinations.. write few of them down on paper for each rule, then it all becomes clear and easy. You don't need to memorize anything -- when an understanding comes by logic then the rules will stick to your mind by themselves, without any effort. And for a long term memorizing there's no need either, because that's just an example for assisting you in creation of your own code later.

17.2d - 01 Basic rules of the code.

o Rule 01a:

Only a combination of the symbols present in the list of cryptic symbols can create an access route to real symbols, commands and modifiers in the matrix of real symbols. New base glyphs, created later on the go, can be used as attachment base for the cryptic symbols present in the list, but the new base glyphs on their own will not provide any access information.

o Rule 01b:

No separate cryptic symbol can ever correspond to a real symbol directly as a simple substitution. Cryptic symbols represent general locations, areas and directions in the matrix of real symbols, in different combinations capable of representing specific locations in the matrix.

o Rule 01c:

All symbols which are not included in the list of cryptic symbols mean nothing for the encryption process, thus can be added (created) as noise and design elements during encryption, and they will be skipped during decryption.

o Rule 01d .

The writing/reading order of cryptic symbols within one whole hieroglyph with all the attachments and separate additional signs are specified by default. The numbers with the letters in brackets show the sections of cryptic symbols in the list of cryptic symbols, to which the subrule of reading order is applied. The writing order may differ, depending on design of the glyph and convenience of writing, also due to the corrections made later (when possible), but reading order is strict by the subrules (by default).

Reading order is as follows:

0. - (Read the subrule 9.)

1. - Base glyph (1, 2b, 7, 8e, 9C); The long vertical correction symbols (9B) are written/read from top to down, all attachments by the subrules;

2. - Attachment below the base glyph (8f);

3. - Attached modifier on the middle left-side of some type of base glyph, in the beginning of a continuing line (8d);

4. - Attached modifier on the upper left-side of some type of base glyph, crossing through the line (first four of 9A);

5. - Attached modifier on the upper right-side of some type of base glyph, crossing through the line (last four of 9A);

6. - Separate sign below the glyph (2a, 4, 6);

7. - Separate sign above the glyph (2a, 4, 6);

8. - Attached modifier in the end of the glyph, at the connection point with a following glyph (3);

9. - Separate sign between glyphs, or after glyph if it's the last glyph in the block (5) (block - all connected hieroglyphs). If the separate sign is before a single hieroglyph or before the first glyph in a block of glyphs then the separate sign of the section (5) is read very first, before subrule '1.', and in this case subrule 9 is subrule 0.

^ - If some part of a whole hieroglyph as described in the subrules of the rule 01d is not present then the subrule is skipped. The same applies to the base glyph if the used glyph in cipher text is not present in the list of cryptic symbols and the newly designed glyph is used instead as an attachment base for additional cryptic symbols. The latter applies only if the base glyph is used for an assistance when the set of symbols in the previous glyph isn't complete to specify a location in the matrix of real symbols.

o Rule 01e:

If a cryptic base glyph has different modifiers written on it, which are representing the same dimension in the matrix, or directly followed by another glyph representing the same dimension as a previous modifier belonging to the set (set = one combination of cryptic symbols locating one precise point in the matrix), while the previous set isn't completed, then the latest cryptic symbol is counted, cancelling previous cryptic symbols with the position information in the same dimension.

(This rule is particularly convenient when you make an error during writing and notice it straight away -- then you don't need to erase the symbol or to try to correct it by writing over.. you can leave the error untouched and write another symbol representing the same dimension with the correct location. When needed then several positions in the same dimension can be applied within the same set, in which case they are pointed out by additional cryptic symbols as all valid -- these additional symbols aren't specified in this example of encoding, but they can be created.)

o Rule 01f:

If a combination of cryptic symbols is specifying a real symbol in the matrix, but in the combination there are additional features not presented in the list of cryptic symbols, then the code is valid, discarding additional nonexistent features as noise.

o Rule 01g:

If a base glyph is not in the list of cryptic symbols but still has additional features and modifiers attached which correspond to the cryptic symbols in the list, then all of them are discarded as meaningless noise. An exception is made when the glyph is created to assist in completing a set of rules started with the previous glyph, to pinpoint precise location in the matrix, or to correct an error.

o Rule 01h:

No real symbols used within the cipher text (mixed into the code) can ever represent true real symbols and are discarded as a distraction and noise. All real symbols are counted only in an encrypted way. The same rule goes for spaces, line breaks etc. -- these are used only as design elements and are not representing true locations of them. (For example if a question mark is written as real symbol '?' within the cipher text, after a block of hieroglyphs, it doesn't represent a question in the encrypted message, and even not an end of a word -- words too can be and must be taken apart to separate blocks)

17.2d - 02 Base cryptic symbols (BCS).

[BCS: sections 1, 2b, 7, 8e, 9B, 9C in the list of cryptic symbols.]

o Rule 02a:

Base glyphs in the upper line (1) of the list of cryptic symbols (further: C-list) have no specified rule or designated encryption value, allowing for attached cryptic symbols (ACS) and separate cryptic symbols (SCS) to specify location in the matrix of real symbols (further: R-matrix). (In comparison to newly created base glyphs the glyphs in the list will always mark additional symbols as valid, regardless of assisting function or separate use.)

o Rule 02b:

Glyphs in C-list section 2b are multipurpose numerical designators. They will never represent real numbers as simple substitution, but help in correcting errors when a location in the matrix must be changed by a number of steps back or forth. Also other uses can be designated, without single application but only in combination with other glyphs.

o Rule 02c:

Glyphs in C-list section 7A specify 3x3 blocks of 9x9 matrix (R-matrix blocks a-d, a-e, a-f; b-d, b-e, ...), which in reverse version writing specify the use of R-matrix 2. (Symbols in C-list between the sections 7A and 7B are the examples of reverse writing of the symbols in the section 7A, not written out for all symbols but logically easily determined).

o Rule 02d:

Glyphs in C-list section 7B are used for specifying 3rd dimension of the matrix (R-matrix columns j, k and l, same on all sections), in reverse version writing (the second column of C-list section 7B) specifying the use of R-matrix 2. The glyphs in C-list section 7B are read from bottom up in each block, all blocks corresponding to R-matrix 3rd dimension j, k and l.

o Rule 02e:

Glyphs in C-list section 8e specify location within separate 3x3 blocks of R-matrix (m-g, n-g, o-g, m-h, n-h, corresponding to real symbols of alphabet from a to e). The C-list section 8 columns a, b and c specify location in the third dimension of the R-matrix (corresponding to j, k and l).

o Rule 02f:

Glyphs in C-list sections 9B and 9C are correction glyphs, meant for overriding any of the mistakenly specified locations within previously written combination of cryptic symbols.

17.2d - 03 Attached cryptic symbols (ACS).

[ACS: sections 3, 8d, 8f, 9A in the list of cryptic symbols.]

o Rule 03a:

The attached cryptic symbols of the C-list section 3 specify the location in 3rd dimension of the R-matrix, in different variations of writing.

o Rule 03b:

The ACS in the section 8d and 8f of C-list specify location within separate 3x3 blocks of R-matrix. The columns 8a, 8b and 8c specify location in the third dimension of the R-matrix. (Similarly to the Rule 02e about base glyphs).

(Notice that the ACS 8f, every single one of them, can be written in many different ways as hinted in the list. The short line crossing the six symbols of the ACS 8f are just one example of modifying them -- the crossing line can be extended with different endings. Also the first three of the ACS 8f have another version of writing shown, which applies to all of them -- the different writing is convenient when the ACS is used with different types of base glyphs, for better design options. The ACS 8f can also be written in a reverse version, flipped horizontally. In different combinations -- with/without line, flipped/not flipped, short/long version -- every ACS 8f can be written $1 \times 2 \times 2 \times 2 = 8$ different ways, and that's only on this example.. there are way more ways to play around, by adding a dot, or few dots besides the ACS, and so on. The power of handwriting again.)

o Rule 03c:

The ACS in the section 9A of C-list specify the use of R-matrix 1 or 2 (as after applying the modifier R1'ae'ij, in step of one real symbol only), in different variations of writing, including unspecified extensions to the ACS 9A. These are the exceptional attachments which can be elongated to produce different design elements while not being discarded as non valid. When the symbols are written (with whatever design) on the left-side of a base glyph they specify the use of R-matrix 1, if on the right-side then the R-matrix 2.

17.2d - 04 Separate cryptic symbols (SCS).

[SCS: sections 2a, 4, 5, 6 in the list of cryptic symbols.]

o Rule 04a:

The SCS in the section 2a of C-list are assisting multipurpose numerical or functional operators, for specifying locations in matrix or correcting errors in other cryptic symbols, or having other uses, for example for pointing out specific areas of a cipher text without changing the code itself.

(In the list there are just four examples, while there are literally hundreds of ways to write these small separate symbols - instead of circle can be a triangle or a square and these can be filled, not hollow, the small line can be with inverted extensions, many different kinds, these can be in several locations extending from the circle (triangle, square) and so on. Also they can point in different directions while having the same design, with change in value of encoding, or without a change in value depending in context. I'm not using them in the example, but I still wrote them down to give you an idea what can be done if such external modifiers will be needed for some purpose.)

o Rule 04b:

The SCS in the sections 4 and 5 of the C-list specify and modify reading patterns of cryptic symbols within hieroglyphs (the subrules of the rule 01d), the reading patterns of whole hieroglyphs within blocks and the reading pattern of blocks of hieroglyphs. Symbols in section 4 are used above base glyphs and symbols in section 5 are used on the sides of base glyphs. If used in wrong locations the symbols lose their encryption value.

(Again, this is just an example of possible use of additional cryptic symbols and not in complete use here for not wasting time on all of the possible rules in detail. But I wrote the rule out in general description because these can be very useful tools when you have forgotten a word or letter during encryption, write it out later and need to swap positions to have a correct sentence or word. And you can change the subrules of reading pattern when necessary.)

o Rule 04c:

The SCS in the middle part of the section 6 of C-list specify 3x3 blocks of 9x9 matrix (a-d, a-e, ..), when added to BCS-s in line 1 of C-list, or to the glyphs in the blocks 8e of the same list. They also specify the use of matrix 1 or 2 of the R-matrix, depending on single/double writing, or x/o for the middle symbol.

o Rule 04d:

If the SCS in the middle part of the C-list section 6 are used together with base glyphs from section 7, the SCS of section 6 (middle part) will correct (not override) the base glyphs. Correcting here means modifying the value of a base glyph -- thus the final value is dependent on the value of the base glyph, but not giving an independent value which would override the base glyph. In that case the SCS do not specify the use of matrix 1 or 2, which is 1 by default and if needed then specified by another cryptic symbol.

(This example is made for showing different use of the same cryptic symbols, depending on context.)

o Rule 04e:

The SCS in the peripheral parts of the section 6 of C-list specify locations within separate 3x3 blocks of 9x9 matrix of real symbols (g-m, g-n, ..), and correct the locations if used after other cryptic symbols that specify location within the 3x3 blocks of 9x9 matrix, similarly to the rules 04c/d, but within 3x3 blocks.

(I should have used the middle part symbols for specifying location within 3x3 blocks, but that's how it came out in the end. Initially I was planning to use the outside SCS of the section 6 for other purpose, that's why they aren't written out clearly in the same order as the middle part, in a separate section. Now few of the symbols are overlapping, but that's only good for safety of the code. You see, there's always ways to improve a method, both for clarity and for complexity while increasing safety -- no code comes out perfect from the first time, even after years of practice.. or perhaps it's correct to say that I've been just too confident, so I completed the symbols as fast as I could and continued right on with the rules. This is a demo code after all, so making mistakes and correcting them later is only good for showing the process in the book, thus I see no reason to make it all over and perfect.)

A note: The C-list section 6 middle symbols are written out in the exact order of application, while the outer symbols will force you into a little bit of a guesswork (for training) as I'm not explaining here in detail how they're ordered. Imagine that you found a list of cryptic symbols of someone else, without explanation attached, and try to figure out the order in it -- be for a short time in the skin of a spy and cryptanalyst -- it will only help you during creation of your own code. For a hint -- look at the design of the symbols.. all of them are designed in a clear logical following, not at random.

Another note about the CS (particularly SCS) in the C-list: For a compact overview of the list not all of the symbols are written out in every way they're used during encryption. Some symbols are written down in a way that it is easy to determine the other options, which are logically following from the rules. This way of writing won't be confusing the creator of the list, because the creator knows what's the purpose of the symbols, and which variations of the symbols are considered valid, discarding the rest of possibilities. It only makes the list easier to read.. and easier to memorize the list prior to eliminating any traces of the important encryption key.

17.2d - 05 Creating additional rules.

o Rule 05a:

Previously described rules (01 to 04) may be modified (become different, specific for certain area of the cipher text). The modification of the rules does not mean that the old rules will be discarded, but that the new rules can be very similar to the old ones, simply applied for encryption differently. The modified rules must be compatible with the previously used rules, not cancelling or contradicting any old rule, if the new rules are used with the old rules together while all of the rules remain valid. If a new rule happens to contradict an old rule then the new rule overrides the old one, while the old rule still remains in use in regular cases, without the use of the (new) modifiers in specific locations and combinations which would apply the new rules.

The SCS of the sections 4 and 5 of the C-list can be used in specific combinations (which are not meant for use to define certain reading patterns of the cryptic symbols) to locate the use of the new rules.

o Rule 05b:

For specifying the new rules, unused combinations of SCS from the C-list can be utilized. Also completely new BCS, ACS and SCS can be created for specifying the new rules, if these symbols haven't been previously used as noise during encryption. The new rules are an option for encoding different data in different ways, and when different level of encryption hardness is required. The new rules are created whenever needed. Once created and used for storing data, the new rules become a permanent set, incorporated into the encryption method. Since the new rules have been used during encryption they cannot be modified -- the cryptic symbols used for specifying the new rules must be added to the C-list and memorized.

o Rule 05c:

When the new rules of encryption are decided, if specific combination of SCS from sections 4 and 5 of the C-list are used or completely new cryptic symbols (which haven't been used as noise) decided, then the exit combination of the cryptic symbols must also be decided, which will end the new encoding rules within cipher text.

As an option, it can also be defined if the new rules, after the new specific modifier is present, is applicable to any content and in any context, or only in certain cases, without which the new modifier is discarded as noise and the encoding/decoding will continue with regular encryption rules.

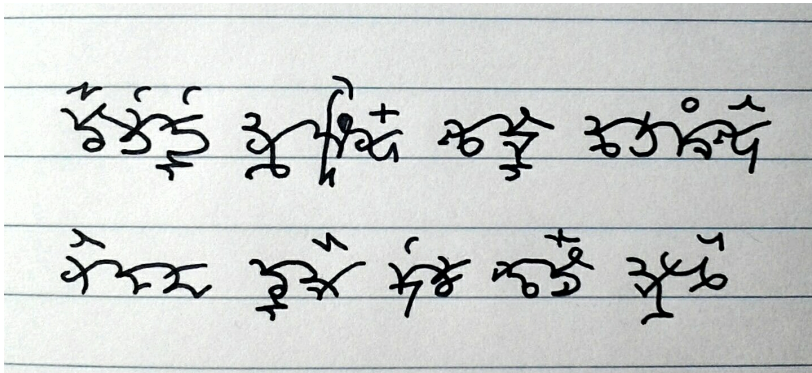
An exception from defining the ending modifier to the new rules can be made to those new rules which by default are applicable for a fixed length of a cipher text, like for one word, one sentence, a set of numbers within encrypted message, or a block of glyphs, and so on.

A note: the error correction tools on C-list section 9 can be used as noise and design elements, to avoid repeating similar combinations of symbols (CS) too often -- you can write down something deliberately wrong and correct it right after, doing which increases security of the cipher text against attacks. You can also use them after newly designed hieroglyphs that carry no information, thus the correction tools will be non valid as well.

17.2e Writing sample.

During practicing it's always a good idea to verify test scripts much later, after giving yourself a time to forget how you encoded them, then a possibility for a repetitive error is smaller.

Find if there's an error in the sample -- since writing it in about 10-12 minutes and taking a picture I haven't actually verified.. no need for it I think.. and if you do find an error or even few errors in the script then you will still be able to decrypt the message, which will be a perfect example of the usefulness of the encryption method.



If you have paid a good attention and actually compared the rules of encoding with the cryptic symbols in the C-list and real symbols in the R-matrix, then you have noticed that the use of some cryptic symbols in C-list haven't been described in every detail. In some cases only the general purpose of them has been given. With the encoding rules which I did describe in detail all of the real symbols and function commands in the R-matrix can be located.

When I was creating the demo version in my mind and writing down the symbols, all of the cryptic symbols were meant to be fully operational. When I began writing down the rules, and later decided I will need to change the rules for easier and shorter description, apart of changes I also left out from use some sets of symbols. Let them be there in the C-list for you to play around with, for training purpose. As I'm not planning to write new examples of cipher text with every single rule which I had in my mind during creation of the demo code, I see no reason to describe all of the rules in detail neither. For decryption of the short message within the sample above you have all the necessary rules described, and it's even more than you need to understand the entire creation process.

Before you begin to create your own code from scratch, if you're a busy person and not sure yet if you want to spend much time on creating your very own original code, then

here's an easy way to test it out in few steps:

1 - Draw an empty matrix with the same dimensions as the R-matrix in the example;

2 - Write into matrix only necessary real text symbols making your own arrangement of the symbols. For testing you only need alphabet, numbers and few punctuation marks like comma, period, !, ?, space and enter (for marking a new line), that's it. It's easy and fast to do. You can repeat all of the symbols in several locations to better fill the matrix, still leaving some empty spaces;

^ .. If you're a VERY busy (or very lazy) person then you can skip the steps 1 & 2 and to use the very same matrix of the real symbols given in the example, for sparing yourself from losing valuable time and from hard physical labor;

3 - Use the same cryptic symbols given in the C-list;

4 - Change the order of the code symbols in the list -- simply swap some blocks of the symbols. For not messing around with writing them all again, print out or copy the C-list, cut with scissors the list of symbols into separate blocks and reposition the blocks at your will.

^ .. You can also skip the rearrangement of the cryptic symbols, going to the next (most important) step.

5 - Create your own rules for practicing the method, to see if the method is convenient for your practical needs. You don't need to write the rules down, and it's even better NOT to do it.. simply imagine in your mind the process of accessing information by the CS in the R-matrix by your own rules.

^ .. That's actually the only important part -- creation of your own rules -- to try out if you like the method in general. Writing cryptic symbols is also good to try, but using just a different arrangement of the CS given in the example isn't a very creative process, while creating you own rules is.

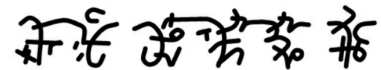
With that quick self test of creating original rules of encryption you'll have a practical hands on experience before the entire creation work from zero. It will definitely be beneficial for the creation process as it won't be your first time anymore. Having a previous experience is always a good thing when creating something permanent, then perhaps you won't even need to discard your first full attempt later, to begin creating a new and better personal code.

A code which you'll be using for encrypting sensitive data you must create from scratch. Every part of your own code, which you're planning to use during entire lifetime, must be created on your own without copying any part of it from here. Writing technique of the cryptic symbols can be used, but none of the ready glyphs that are in the examples. It is mandatory for safety reasons.

If you're planning to share a code with someone else for exchanging encrypted messages, which will be a different code from your own very private code of course, there are more benefits compared to regular non encrypted letters, apart of just hiding a message from third parties. With this encryption technique you can write in a single stream of a cipher text several messages meant for different people. By combining in a single stream of a cipher text different sets of cryptic symbols, which don't overlap (or if they do, then are marked/discarded by additional attached symbols specific to the set),

each person can get only the message meant to him/her, discarding as noise the symbols which don't belong to the correct set of the method at hand. It provides with the possibility to transfer written encrypted messages to different people with added comments particular to each one, with specifics known only to them, in the same written letter. With this is also ensured that the letters pertinent to an issue are always delivered to all required recipients, because in loss of a letter anyone can copy his/her own letter and hand it over to another one. On top of that it also ensures that the full message is received.. the message to each part remains intact as long as everyone receives the full message.

How's going by the way? Did you decode the message in the medium level cipher text? Now imagine for someone having no matrix, no list and no rules available, to attempt cracking the code.. that example should be enough to create confidence in the encryption method.

A sample of handwritten encrypted text, consisting of five characters in a stylized, cursive script. The characters are interconnected and appear to be a mix of letters and symbols, representing a message that has been encrypted using the method described in the text.

18 Exercises.

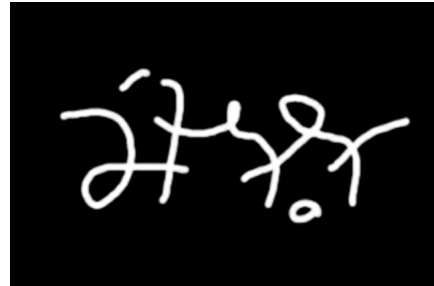
For improving your skills in the cryptography method with hieroglyphs there are only two specific types of exercises needed: to improve attention to detail within the glyphs and to improve the skill of drawing the glyphs with all the attention to detail. All the other skills necessary for the encoding method are not so specific to the method -- many other fields of life train you in general skills.

Obviously, those who have learned the languages which use hieroglyphic writing are much better off in this method of encryption, but at the same time this previous skill of writing different kind of glyphs - with completely different rules - may even be an obstacle. Exercises are good for all, with any background in languages.

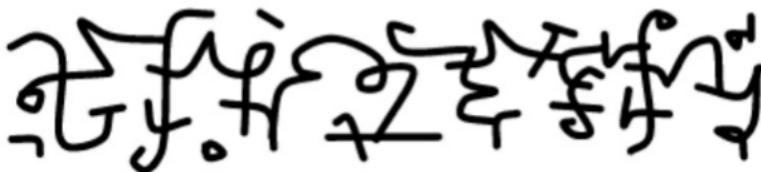
18.1 Attention exercises.

With your friends you can help each other with some visual attention exercises. You can write on paper whatever script, initially without any meaning, copy it, then change just a few symbols a little bit in one of the scripts -- for example extend a line or a curve or add a line with an angle, and then copy the modified script. Then ask your friend to spot the difference on the two copies of the script.

An example 1, part 1 - original script:



An example 2, part 1 - original script:



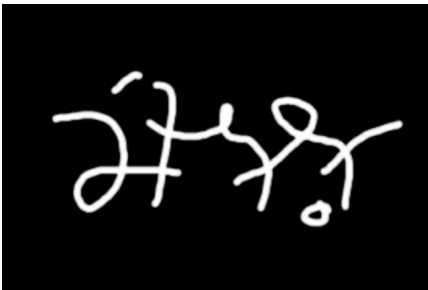
The original and modified scripts must both be copied because then one can't figure out the changes made by the applied extra ink. The exercise should first be made with bigger differences in scripts and gradually with ever subtle differences, up to

reasonable degree of course, for a practical script writing purpose. While making changes try to stick to the logic of hieroglyphic writing described in the book -- not all the differences between hieroglyphs can count as differences in encryption -- you will never be able to write the same glyphs absolutely identical, making rather hugely differently looking glyphs at first glance, but with exactly the same value, thus only the relevant changes to hieroglyphs must be made right from the beginning, even for the visual attention exercises.

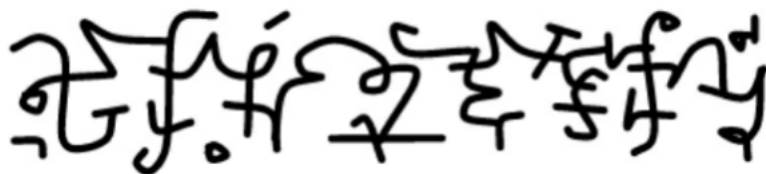
The script comparison samples should be shown separately, not together, to have a better effect on training memory and attention to detail. If one cannot spot the difference then both copies can be shown together to find the difference. Then new examples can be created repeating the process, showing first the different scripts separately. After some practice everyone should be able to see the whole picture and be able to spot the difference (in short scripts) without looking at them both at the same time.

These are two similar looking but far not identical scripts (part one of the exercise is on the previous page). Find the difference -- just one. This little difference in writing will make a big difference in deciphering the code:

An example 1, part 2 - modified script:



An example 2, part 2 - modified script:



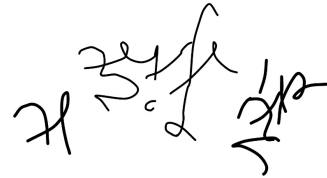
In the modified script of the second example there are four differences significant for the encryption method. Try first find the differences by visual memory, and only after the failure to find all the four differences compare the two scripts side by side.

18.2 Drawing exercises.

Even if you're good at drawing, having a lot of practice in precise hand movement, drawing the hieroglyphs which have a meaning in every line, dot, curve angle and so on,

is quite different compared to an approximate representation of things where few lines more or less won't make any difference. Also the Chinese and Japanese hieroglyphs do not compare in precision to drawing hieroglyphs for cryptography -- it requires much more attention to detail than just general look of something represented in few almost careless lines. You must kind of learn to draw again, no matter what your experience.

That's probably how you'll begin to draw glyphs in my style of writing..

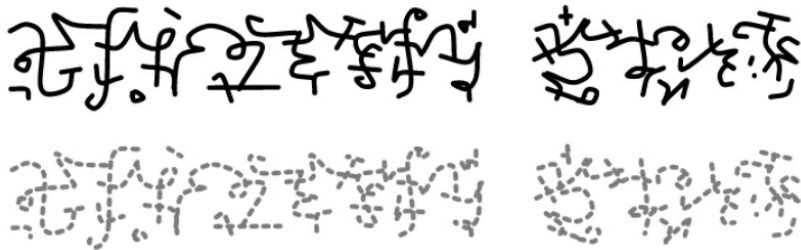


Here I give you some examples of glyphs to draw lines over them with ever greater detail (you don't need to practice on this page -- there are pages to print/copy in the end of the book -- chapter 28, with the exercises as well) ..

The simplest glyphs, the beginning..



More complex and connected glyphs (as in these glyphs there are more details - which are important - and harder to understand, I include the sample script) ..



After several of the simplest exercises is easier to begin creating your own cryptic symbols of whatever design possible. As I mentioned, in the end of the book there are many glyphs to practice, on the pages to print/copy.. you can enlarge a copy if necessary in the beginning, and later to make another copy much smaller for the same exercise.



19 Commercial uses of the cryptography method.

Apart of securing some data from access by third parties there are other uses of the encoding in hieroglyphic writing as well. I'm describing some other uses of the method in other chapters briefly, but to the commercial use of the method I dedicate entire chapter. Not only because it needs more detailed explanation but also because while implemented in commerce the encryption method will become popular faster than with simple promotion.

19.1 Cryptic messages project.

Some time ago I sent to many marketing and design companies around the world a letter with a project involving my cryptography method. I was trying to find interest in a project which would help to promote different types of products using messages which consumers would need to decrypt, to get a prize for the right answer. Obviously I was keeping in mind the most basic version of the advanced encryption method (described in the chapter 17.1), otherwise it would be an impossible task to find a correct answer.

Here's an excerpt from the letter. From the original letter are excluded some parts like the examples of the script, links and contact information:

/ ... /

The project is meant for promotion of different consumer products in a unique way, by using encrypted messages on the products, to let buyers to guess the meaning of the messages. It brings an extra interest to the products and a possibility to win prizes for those who happen to guess the encrypted message(s) correctly. The messages will be in handwriting cryptography, which is aesthetically beautiful and provides additional design value to the products. The messages can be put on wine glasses, cups, plates, pens, postcards, designer furniture, lamps, hand bags, T-shirts and other types of clothing, food packages and many other types of products. The cryptic messages can also be used for a lottery type of games and on public promotional ads, to bring an extra curiosity. The handwriting method is developed by me and is completely original.

For the purpose of promotion of products I will create a simplified method of encryption for each product. The matrix of symbols used (alphabet, numbers and their position in the matrix) will be for public to see on the website of the related product. The key for each different message will be secret (way to access the alphabet and numbers by the cryptic message). Between those who guess the message right will be shared a prize (or several prizes in the order of the right guesses made -- first right answer gets a better prize). The cryptic message will be made public and prizes will be given after certain amount of products with the specific message is sold. If the product is sold out but nobody has correctly guessed the secret message, then more products will be put on market with the same secret message, the prize for guessing it right will be higher, and some hints about the secret

message will be published on the product's webpage. If a product won't sell enough then the prize for guessing the cryptic message correctly will be made higher, while the price of the product don't need to be lowered. When the correct guess has been made, then also the key of the code will be made public, together with the message, so anyone can verify its validity. After enough products is sold and the right answer arrived, then the price for the rest of the products, with already publicly available meaning of the message, will be lowered, which is an additional marketing tool. A new cryptic message (using a new key) will be put on following products, which will be sold for higher price till the message and the key of the code will be made public again. Depending on product there can be different conditions for participating in the guessing -- for some products anyone can make just one guess, for other products a ticket of purchase will be required and a certain number of guesses can be made. The encryption method allows almost endless flexibility to accomodate the cryptic messages to specific products.

/.../

Regardless that I sent the project to literally hundreds of well established world class companies I received only few responses, all negative (meaning no interest). One company responded that they don't want to read the whole project (which I sent as an attachment and a link to the page on my website where the project was published), because as they claimed, they may be already working on a similar project and don't wish to have any issues with copyright later. Well, I responded to that assertion that the project is unique as there doesn't exist any encryption method similar to mine, at least not what I know of, and I have made a proper research to be confident in the non-existing competition in that particular method at the given time. But I got no more response. One company responded to my project proposal with a question as in which department of the company I would be interested to work. I responded that I would prefer to work as freelance, on contractual terms, seeing my part in creation of codes for promoting specific products. I got no follow up responses.

19.1a Guessing of messages.

Guessing game is only possible while using the simplest, straight layout of the alphabet in the matrix, and in easier versions even made public, otherwise it would be an impossible task to find a correct answer. Only the rules and the list of cryptic symbols must be secret for a guessing game. Depending on the length of a message there can also be the list of cryptic symbols made public (if the message is very short), keeping only the rules secret as the key to guess. Or, if the message to guess is quite long then the layout of the matrix (the dimensions etc.) can also be secret, only publishing the real text symbols (the alphabet, numbers) which are used in the matrix, in a single string of symbols.

If during a specified time no correct answer has arrived then one by one can be published essential parts of the keys.. the dimensions of the matrix, the placement of symbols in the matrix, the symbols used in the message (one or few at a time), the list of cryptic symbols in a single string, then the ordered list of the cryptic symbols, then few of the rules.. and if nobody guesses the message correctly then the prize can go to the person who sends in the right answer first, after the full key is published.

There are many ways to organize the guessing game, both for fun and for the

product marketing ends.

19.1b Creating a lottery.

A lottery with hieroglyphs can be very similar to the guessing game, but with mathematically precisely calculated values to the level of correctness of an answer. For determining precision there may be used locations of text symbols in the matrix -- the closer to the right location the guessed symbols the more points one gets. The easiest way to determine points is by the exact location -- the more symbols guessed the bigger the prize, while only full words of certain length are accepted for participation. A condition can be made that only the words existing in certain known and publicly available dictionaries can be used, then there won't be any issues about some highly technical word of some profession to be accepted or not. While doing the lottery online that's an easy task. This approach can have an advertisement benefit both for dictionaries by the lottery, and for the lottery by the dictionaries.

Another way to calculate the precision for a lottery is by the cryptic symbols, when participants must guess which parts of the cipher text carry information and which parts do not, being included as noise for (better) hiding the message in the script. Also both ways of calculating hits can be included, with different prizes for each, and the main prize for those who guess both: the exact word in the message and the cryptic symbols used within the cipher text which carry information. There are more options, but to make the lottery of a hidden hieroglyphic message even more complex wouldn't produce a very attractive lottery for a larger public in my opinion, but you never know -- I may underestimate the public interest -- the lottery using cryptographic method may take off wild once it has began.

In making the lottery more complex the exact dimensions of the matrix of real symbols can be guessed, the number of cryptic symbols in the list can be guessed, and the number of rules can be guessed. Perhaps there are more ways to add guessing values, but that isn't needed anymore.. the lottery would become too complex even to begin reading the rules of the lottery game itself.

When lottery is simple enough for almost anyone being able to understand the rules, then it may be a game on the spot, to win a prize locally -- a product, or a bonus to a product that has been purchased, giving the right for a free lottery ticket.

19.1c Promotional ads.

The promotion of products without a secret message, but still using the hieroglyphic writing, can be done when a company adopts already an existing and publicly available code, for example from past guessing games with the key of the code available on the company's website. Thus once popular guessing of the code can continue as a trademark encryption method for the product line. A company can also develop his own simple code without a previous guessing game, while in the commercials the spy theme can be utilized -- there are many popular spy thrillers which could lend a theme for a commercial, and of course original motives can be created. The spy theme will never go out of fashion as long as human culture continues.

19.2 Cryptic signatures.

A simple signature is easy to fake. There are secure digital signatures in use these days but all human legal transactions will never be digitally confirmed. Common handwriting signature will remain in use. Here the cryptic signature may come for help against misuses. A cryptic signature will be harder to fake and it will also provide a level of confidentiality, in cases where the name isn't written on the document but only a signature is required.

For using a cryptic signature there can be used the most basic version of the advanced encryption method (described in chapter 17.1), which takes only a few minutes to create when you have done it before. You can change the cryptic signature while it remains yours. You can create the key of the code in a small piece of hard paper, protect it with a plastic cover, and carry around like other documents. When an occasion demands to sign something and it's permitted to do it in encrypted way to hide your name from others who sign the same paper, then instead of your known signature you can create a cryptic signature on the spot, using the key you carry with you. Then in another document you can use a different cryptic signature so others can't relate the documents as both been signed by you. But then needed to prove the authenticity of your signature you will prove it with your key, which isn't secret but simply not available to anyone, like your passport.

The key can be copied and kept in a secure location as a backup in case of loss, and you don't need to write your name on the key. Anyone who finds the lost key will not know to whom it belongs, so a deliberate forgery of a signature becomes impossible -- the person who finds the key won't know the person to forge the signature, except stealing the key directly from the owner of the key. And even then it will be impossible if the owner of the key won't have the full key written down, keeping few extra elements in memory, for safety. At the same time, anyone who finds the key can use it (even if the key is incomplete, without the extra safety elements) for securing his/her own signature (adding personal safety elements), or for sending lightly secured messages to someone having the copy of the same key. The probability of accidentally having identical key with someone else, an unwanted party, is extremely low even without the additional safety elements which are created by each user of the same key and memorized. the code won't be unbreakable though, by other users of the same key, but that isn't needed for signature -- you only need the protect your signature against forgery.

A cryptic signature is definitely safer than a simple signature. You can create several different cryptic signatures in advance the design you like and to practice them for a bit. Then you won't have to invent a cryptic signature by the key on the spot, but use one of the available cryptic signatures which you use in different occasions, all of them valid.. or even a non valid signature if needed due to some social pressure, if you can't refuse a thing that others demand you to sign, so if later you'll get into a complicated situation you simply won't recognise the signature as yours.

For using your cryptic signature on an official document with your written name on it the key of the cryptic signature must be registered in some ways, for not claiming later that the signature isn't yours, and the authorities must be capable of verifying the signature with the key, I suppose, but that's not absolutely necessary because in these occasions you can use your regular signature.

19.3 Cryptic orders.

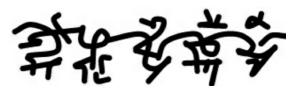
The use of cryptic orders can be accomplished in a similar way to the cryptic signatures, with the most basic advanced encryption method. It may be utile within companies and also between different companies which are in a regular partnership. Instead of cryptic signature the encrypted parts will be some items, or demands, or whatever needed, which are written on a document using the hieroglyphic method, while only the dedicated parties can read the messages. Then the document can pass through all the other departments and delivery persons without disclosing the sensitive parts of the document.

19.4 Other commercial uses.

In a company which has many workers there are often message boards to the workers in different departments, which all of the workers read when arriving to work. There's no end to human curiosity thus some people read the messages not meant to them. Not every company has the means and even economic benefit for setting up electronic devices of communication, because these need maintenance, repair and so on. Thus the message board will still be there for many years to come in many large companies. To specify some messages on the board for reading by the people in certain department only, a simplest encrypted message can be put up on the board, while all of the workers in the department have the key. If someone has a difficulty to read the message on the board then another one in the same department can help out. Shaking up the capacity of the brain to function properly will only be beneficial for a company in a life of a routine.

Encrypted messages in a company don't need to be in a completely secret writing. A steganographic way may also be used. In the end of the book on the pages to print/copy (chapter 28) I give an example of writing cryptic messages on top of a regular text. The example of different inserted codes are on the first five lines of the sample text -- you can practice with the following lines on a printed/copied page and then copy it after your cryptic symbols are written, then you will see how much of the added signs will be lost and how many remain visible after several copies.

Almost any field of human non-digital communication which needs some parts of the information exchange to remain confidential or secured may find the cryptographic writing method useful in some ways, while using a light encryption in basic level or strong encryption in medium level. The strong encryption in high level is mostly for personal use only, for having a code which one can use the entire lifetime without worries.



20 Writing music.

As I mentioned in the chapter 'Application' even music can be encrypted using this method. Many years ago I even made an attempt to encode music. I have studied some musical instruments as a child, in a music school, thus the idea came naturally at some point. But that attempt of encrypting music was just fun, not for a practical purpose. One must be a genius to write music without playing an instrument, thus encrypting notes has no point -- in most cases anyone can hear what you're trying to create while you practice, and after all, music isn't something you wish to hide normally.

For some musical instruments this method of writing may be useful, for easier and compact reading while playing, without encrypting in a secret way of course, simply encoding for fast reading. But for most cases all the necessary musical writing techniques have already been developed perhaps to the best possible way it can be done. People have dedicated to this their whole lives thus there's no need to discover a new way to do the same thing.

Handwritten musical notation in a stylized, cursive script. The characters are fluid and interconnected, resembling a shorthand or shorthand notation for musical notes and rests. There are approximately six main symbols arranged in a horizontal line.

21 Advanced games for training.

Already for some time I've had in my mind the idea to develop games, for creating more interest in the method and thus training in the capacity to read this kind of hieroglyphic writing. One example of the games is described in the cryptic messages project (chapter 19.1), a guessing game, but that's a very simple version. The encryption method allows to create way more sophisticated games, not only for a larger public but also for trained mathematicians and cryptanalysis professionals, up to the point till they begin turning themselves in for a psychiatric assessment and committing suicides.. just kidding.

It may indeed seem too tough, but again, it's like with music -- if you learn to play different kind of instruments from notes, then picking up another instrument which is quite similar will be much easier.

The problem here is that for me games are a waste of life, except if they demand physical input and thus help keeping you healthy, or those intellectual games that don't go over the same mental processes endlessly but teach you something new, practical for life -- but then I don't call them games, I call these activities training, to make the distinction. The distinction must also be made between those who practice and those who watch others practicing without ever having been engaged in the activities themselves in any serious manner, so called fans.

I haven't played any computer games since I was perhaps in my twenties, or maybe I stopped playing games even earlier, before I joined army in 18 years of age. After the end of the service it happened that I played minesweeper in some rare occasions -- I must admit wasting few hours of my life on it. Creating myself a game for this method of cryptography isn't really my calling, but as you see I did try to make a business with it, offering my services to marketing companies.. I suspect that from my letters was possible to read out the lack of enthusiasm by some syntactic rules known to people in the marketing business, and that's why I had so few responses regardless of a good idea.

I realize it will be beneficial for promoting the method to create some games so I'm open to this option somewhere in an unpredictable future when I'll get nothing better to do with my life, which actually I don't see coming. Life is very Interesting and there's much to discover. The book has waited to be written for so long time because I've been traveling extensively, been around the world, meditating in high mountains in several places, living in many countries for a long time, and all this demands time. As I love traveling then I hardly imagine spending time on creating games, only on writing the book for few weeks or months.

For now I leave the creation of games to others who would like to get involved in making this type of cryptography a popular subject in peoples lives. With creation of games the cryptography method can get a bit better traction in popular culture I suppose, because there are so many people for whom playing games is the meaning of life. Creating games on the subject that can benefit someone's life it's not a bad thing. With the cryptography games the attention of at least some people will be taken away from the games that are completely useless in acquiring practical skills.

I'll be happy to read about the encryption games from others who will come up

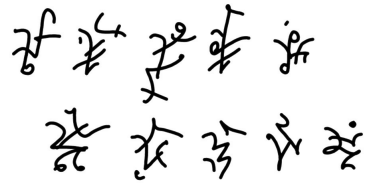
with something, either described in this book or something new, while using the basic principles of the method. That's my goal -- to promote the method to the point till it will be a compulsory subject in the education system of every civilised country on this planet and beyond.

ॐ मेरु ॐ

22 Cryptography in social life.

Clubs, organizations, religions, politics.. any social environments can make a great use of the cryptography method. It becomes especially useful because you don't need other tools but your brain to encode/decode messages, right on the spot (event, gathering) when necessary.

For example one could put under a slogan of a political party an encrypted smaller message to those closer circles, dedicated to the key of the code.. like saying in the hidden message "don't believe it, that's just for a public sell of our party".



23 Cryptography in gambling.

Anyone playing in casinos professionally knows that there are all sorts of people around in those places. Most of them are just casual losers or have dropped in from curiosity. Those who do play seriously are noticed and kept an eye on, to find out if they really are regularly winning and making some profit.

For certain casino games there are permitted to make notes, and if you're the one who do know the secret of the trade, it is better to keep your notes encrypted. The writing method is particularly convenient for certain casino games because it allows to make notes in just a few short symbols in place of a regular writing which would take for the same amount of information several times more symbols. But you need to develop your own writing of course. Your gambling notes don't need to be completely (deeply) encrypted because your play is visible to others anyway, but your long term strategy will not be known to any curious onlooker because others won't make any sense of your notes just by a quick look at them.



24 The highest level method.

The medium and high level encoding techniques of the advanced handwriting cryptography method aren't principally very different. Both of them are utilizing ready parts of hieroglyphs which in correct combinations are pointing to precise locations in the matrix. The difference in them comes mostly from the amount and level of complexity of the rules. The highest level encryption technique is more complex and more simple at the same time, depending on the point of view.

Lets have an overview first, beginning with the most basic level:

- ✓ In the most basic level (chapter 17.1) the combinations of the cryptic symbols are very straightforward, with no many possibilities of different combinations of BCS, while utilizing several additional cryptic symbols, ACS and SCS, for safety.
- ✓ In the medium level code (chapter 17.2) there are many more combinations of BCS with ACS, which can by themselves change value with the help of correction symbols and some SCS. The ACS/SCS themselves are also much more numerous.
- ✓ In the high level code (which I mentioned within the chapter 17.2) the principles are similar to the medium level code but with less BCS and many more possibilities of applying different rules to access information in the matrix, with the help of a larger number of ACS/SCS in many different combinations.

Now, the highest level code has no specified BCS, ACS or SCS at all, only specific rules to read information from the angles of lines, curves, dots, circles, triangles, crossing lines in different (relative) angles, relative positions/sizes of the elements and so on. There are countless possible combinations of them with consequently changing values which are also context dependant, location dependant, relative size dependant and more.. not all the specifics I can make known here because that's the method I use myself.

You can already get the idea that in the highest level code there's no list of cryptic symbols, only the rules of reading values from the elements of the glyphs and very complex matrix with many dimensions. All the hieroglyphs in the cipher text are created on the spot by the specific rules which allow to encrypt/decrypt values in the glyphs for accessing information in the matrix.

It may all look a bit similar to the principles of Chinese/Japanese hieroglyphs but with way more clear rules, because the rules of cryptography specify precise numerical values for accessing data in the matrix -- no varying exceptions which are present in any natural language. Of course you can include at will specific exceptions to your highest level code and with that to make the code even more secure, but there's no need for it. The code is in principle already extremely secure. You can use for noise any combinations of the elements which aren't included in the rules. As a reminder: a single specific symbol or specific combination of symbols designated for the purpose can discard all of the default rules which allows you to design for noise (and for aesthetic look) any hieroglyphs at will,

including the elements which in default conditions you're using for encoding with valid values. The options of including noise in the cipher text are limitless.

It will be impossible to crack a properly created highest level code because those who would attempt to crack the code will have no access to so many parts of the puzzle:

- number of dimensions in the matrix;
- size of the dimensions in the matrix;
- all of the real symbols used in the encryption method;
- which alphabet letters and punctuation marks are doubled/tripled/quadrupled/.. within the matrix;
- how many sets of numbers there are in the matrix;
- which alphabet letters are combined into syllables in the matrix and how;
- how numbers are combined into pairs within the matrix;
- how many positions in the matrix are utilized for noise;
- commands and switches used in the encryption method;
- position of real symbols/commands/modifiers/switches in the matrix;
- values of the elements in the glyphs (lines, line breaks, curves, dots etc.);
- relative sizes of valid elements in glyphs;
- valid combinations of elements in glyphs;
- values of valid combinations of the elements in glyphs;
- context of applying values in elements of glyphs;
- elements in glyphs with no values, the noise;
- reading patterns of values within glyphs;
- number of reading patterns within glyphs and when they change;
- up to how many glyphs takes to access one location in matrix;
- changes in the application order of different hieroglyphs;
- rules of access of information in the matrix;
- context of using symbols and commands in the matrix;
- which words are preset and encrypted with a single symbol (TS and/or CS);
- how the preset words are created within hieroglyphs or within the matrix;
- which prefixes and suffixes are preset and how are they used in the code;
- which grammatical rules are preset for easy creation of sentences;
- in which context the preset grammatical rules apply within cipher text;
- abbreviation tools for words within cipher text (TS/CS);
- abbreviation tools for sentences/propositions within cipher text (TS/CS);
- highlighting tools within cipher text;
- error correction tools within cipher text (TS and CS);
- copy/add/paste tools within cipher text (TS and CS)
- .. and more -- only known to me;
- .. and much much more, limited only by the imagination;

(The CS in context of the highest level method means specific combinations of the elementary parts -- the elements from which the glyphs are created on the spot -- lines, dots, etc., and not the ready glyph parts as in basic/medium/high level code).

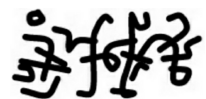
The possible number of different combinations of all of this is quite impossible to determine. Even if the number of possible combinations of the separate elements (lines,

angles, curves etc.) can approximately be determined by some level of human precision in writing, and by that also the number of possible combinations between the elements in a glyph, then possible number of rules can only be limited by human imagination. So the lifetime safety of your code is guaranteed because even the repeating hieroglyphs which you happen to create on the spot similar to each other, don't need to carry the same information -- the hieroglyphs can change access values depending on context (previously encrypted information or previously used elements in the cipher text). After all, you'll not be encrypting everything in life, only in case of necessity, thus there will never be the critical mass of a cipher text out there even to consider an attempt of cracking the code.

Regardless of all the mathematical impossibility to determine the used logic of encryption by third parties the code is still simple to use by those who know the key. In the sense of the cryptography method, while numbers are being used as values hidden in the glyphs to access information in the matrix, you can say that a hieroglyph is worth more than a thousand numbers.

Before going for the highest level code for your personal use you should begin practicing with the simpler versions, which can also be useful in some occasions. You should always keep in mind the Basic Principles of Handwriting Encryption (described in the chapter 6). Without paying attention to the basic principles whole your work may turn out useless - the keys to your code must remain secret.

One remark about whatever encryption method, not only the method described in this book: If there's a real need to crack a code of a common person then in many cases a simple, verified by historic accounts method -- fingers under the chair -- will produce all the necessary keys. Thus, don't overwork with this, have some common sense and consider your realistic needs, not an utopian scenario of aliens attacking and necessity to have an unbreakable code against an advanced civilisation, to save mankind.



25 Work on the book.

The book was first written in 2013 and published for free online, a promotional version. The significance of the encryption method became obvious almost instantly as just in few months the method took most of the first page on google search on keywords "handwriting cryptography", also in picture search. A good beginning, but I had no time to work on the extended version of the book till two and half years later.

The editing of the book in 2016 was done on the tablet which I succeeded to buy, but not before I succeeded to buy also a backup tablet, because one had crashed just a week after purchase and was in guarantee reparation for two months. The second tablet also crashed a month after purchase and was in guarantee fixing for a month. Thus I didn't even begin editing the book before I had tested and set up the tablets, waiting to find any issues again.

The main editing/updating of the book was done in a quiet and relaxed ambience, but that was only a temporary joy, in a place of a friend which was vacant for a month. In finalizing the book the old lifestyle of almost constant movement, which took time away from working on the book, was the reality I had to accept.

My life can be concluded with the saying that the world is my home. There are so many places I call home that now I already feel at home almost anywhere I go. I have lived for a longer period of time (for half a year and much more) in over ten countries. You don't need much money for such a journey, there are many people who can confirm that with their lives and there's nothing special in it these days. It can be special only for those who live that life and enjoy every minute of the life on the road.

Here's a map of my travels until the year 2012, general overview with some places where I've been (if marked all it would be 3 times more locations)..



In Russia, Europe and Canada there should be on the map not the dots but zig-zag and criss-cross lines. I have traveled by hitchhiking in Europe and Russia many times. Canada only once from Vancouver to Montreal, in four/five days -- didn't pay attention to the time so it's hard to recall exactly. The longest hitchhiking trip has been from Western Europe to Far East, till Vladivostok, Russia. It took three weeks one way and over two weeks back, occasionally on cargo trains. The trips have been so fast because regardless staying in some spots for a long time waiting to get on, I often got on trucks with two drivers, so sometimes I got several days of almost non-stop ride.

I had also planned to travel through whole the Africa but at the last moment my common sense got hold of me -- I don't want to become a free food package from Europe to wild animals of Africa with my poor means to travel. To Australia and New Zealand I simply couldn't afford to buy a ticket. Japan and also Tahiti were much cheaper to get to, buying a ticket well in advance. Australia is still in my plans, and Africa too, and several other areas in world, in Asia and Middle East perhaps. But not now, one day when I'll have the means to buy proper equipment for such a journey, and secure the means for a fast return just in case.

Some people have said to me that I should write a book about my travels. I don't feel unique about it -- I have met people whose travels are way more crazy than mine. Let them write their stories. I stick to my cryptography methods -- this is no doubt a unique thing, completely original and so far with no competition.



The photo above is taken in Montreal in 2009. I don't take much photos of myself, thus there aren't many to choose from. There are quite a few selected photos of the places where I've been, up on my blog.. youthextension.wordpress.com -- it should remain active in any circumstances because it's free (in financially tough times I had to abandon some of my paid websites and domains which referred to the sites), and perhaps some new photos will show up there as well one day, when relevant to context.

During my travels I worked on developing the encryption method quite often, especially when there was nothing else better to do. I prefer to spend my life as fully as possible -- learning, training, developing -- thus while out in the wild and daily meditation done then it wasn't an option for me just to sit around doing nothing. It would have been equal to wasting life. Then I worked on something useful, which may help my life is some

ways, and cryptography was one of these useful things.

I must admit that my friends haven't been very supportive with me about working on the book, those who knew it. Even though the book is a technical piece and on a unique subject, my friends still thought it's a waste of time and were very much insisting on it, up to some of them having arguments with me quite angrily proposing that I should better fix my life -- find a job and settle down. So, after meeting such 'resistance' several times, at some point I simply stopped talking about the idea of working on the book, which by this time was done in general but not up to the level to approach publishers in my opinion. Only occasionally I mentioned that it's still in my mind and needs to be done. Instead of writing I was running around trying to find a job, which in total makes perhaps over a year of wasted time.. could have written couple of books in this time, but who knew in advance. Some small jobs helped to pay my phone bills and some other urgent things.

Below is a photo of my work place in Norway, in a friend's garage, away from big cities. I enjoyed the place, and there the book on cryptography was first written. It took only few weeks to write the book when finally I got the opportunity in the summer of 2013. I was sitting in the office in a car repair company writing the book non-stop, living on coffee, sandwiches and occasionally cooked hot food..



I had to concentrate on this work entirely and it was done in no time when I knew I have the chance to work on it with no pressure. Every writer has his/her personal likes in writing a book. For me I need to take the work up and do it once, not leaving some parts for later time in life, who knows when, while finding another moment of free time. I finished the book in few weeks in 2013 and after a fast grammar check in the end I called it a day.

But even during that time I had some issues. Here's an excerpt from my post on a blog..

Saturday, September 21, 2013

/.../ sorry for not posting /.../ documentaries recently.. i'm working on my book on handwriting cryptography and mostly offline. each time i go online i must clean up the

computers i used - believe it or not my computers are heavily under attack.. currently someone is trying to insert trojans and, surprise-surprise, by google applications -- first got attacked heavily while using google chrome, removed it, today got the same trojans through google earth even without using it.. going to get rid of it too. i'll take some time off from posting /.../

I wouldn't have taken up the work if I didn't have several computers and tablets in my disposition at this time. I kept writing without going online with the main device more than needed for uploading my work for backup, and even then I preferred to first transfer the files with the USB key to another device. Not being paranoid.. I simply don't like to do the same creative work twice, thus the precautions were justified.

Then the book was unchanged for over two years, but I did create some encoding examples in 2015 which I decided to include in the book in later times, so I was waiting for the opportunity to come. I asked several friends for the possibility to lock myself up in a quiet place, a villa or a garage, to work on the book, but nobody believed it's a good idea, that almost nobody reads books these days and I'll be wasting my time trying to find a publisher. But I decided that I will improve the original book earlier or later, and in the end I found a friend who provided me with a place to stay for a month in 2016, where this book was taking its current shape. Although it wasn't a deal with anyone just to sit and write.. I did do some small jobs as an exchange for having a place to work on my own thing.

While I've been most of the last 20 years traveling, in every new place where I decided to stay for a while I had to find a temporary job, where possible getting the paperwork done for a legal stay. And I had to find a place to live with all the head ache coming with it. During my first years in exile it wasn't that hard to find a job, only needed to learn to speak the local language a bit, and then it was easy.. but as having no legal stay these were all temporary and little paid jobs. Many years later, when having legalized my stay almost everywhere I went, no matter how well and how many languages I spoke by this time, to find a job was often almost impossible. The world had changed and not for the better.. for the working class I mean.

In the beginning I even managed to kind of settle down, to rent a place, design my own home buying all the things a cozy home needs.. but after several places changed, giving almost everything away or leaving behind, taking with me just one backpack with necessary things, I stopped setting up my homes even when staying for a long time in some places. I couldn't afford to waste money on anything more than minimally needed as I was never sure I could call another new place a home the next day.

Often I began in a new place living in tent, or simply in sleeping bag on the street, and that lifestyle often took lots of extra time to find places to take shower and wash clothes, to find food while no money in pocket, to find from donations clothes and shoes when they were wearing off, to travel between places of camping and places to eat daily etc. -- writing a book wasn't an option in these conditions because even the battery of cellphone had to be charged in a random place, not to talk about working long hours on a computer or tablet, which I did succeed to buy with the money from occasional jobs after some devices broke down or got stolen. The irony is here that when in recent times I worked sometimes with a contract and rented an apartment I had less free money available for electronics than when I lived in a tent and only had occasional jobs from time to time -- the pay of workers had become so small compared to expenses.

There have been many electronic devices I lost in my life to destruction, hacking and stealing, well over ten of them in the last 15 years, since time when I became more active in politics than I was before. I was actively spreading truthful information about how the world really works. I did lots of research on current world affairs as it was more important than writing a book. In times when I lived in a house and had a connection to internet I was sharing alternative documentaries in many ways -- on my websites, blogs, by e-mails, on DVD-s, shared fliers with my website address in public places to spread the information unavailable on corporate media, and so on. I have dedicated to it a lot of time, especially since 2003. A lot. And I don't regret it. So no surprise several my blogs got blocked, documentary films and political videos often removed and related accounts on social sites deleted -- that's the so called freedom of information in our 'democratic' world.. up to hacking of my computers and worse.

In times when I found somewhat stable job and rented an apartment, in few occasions I noticed that the places were searched without my presence. One day when I was supposed to be at a meeting but decided to stay home, one man entered my apartment with his own key. I heard someone entering and confronted him at the door.. he mumbled something that he had lived in the place before and forgot something. I let him go not to begin a full investigation as I had enough issues to deal with -- I simply had no time for it as economical survival was priority and finding out who spied on me wouldn't change in my life anything. I was doing nothing illegal, I had nothing to hide and it was actually them who wasted their time on me. Since then all the secret searches had stopped in that place, or been made more carefully, who knows. Well, I also installed an alarm with detector of movement, so maybe that helped. Some years later, in another country, in one place which I rented on the last floor, the ceiling was drilled through like in a spy movie. Coming home I found some small white pieces on the floor which weren't there before going out. I tried to figure out what happened and laid down, and then I noticed a hole in the ceiling. Funny as it was, I made a remark about it on one of my social accounts, and I got left alone again.

At some point in my life the things had began developing so weird, as to noticing people following on the street, in several countries, which by my work experience in security and personal protection wasn't too difficult to understand. With specific actions I made these people to lose an open interest in me, but I did have to keep my casting agent informed, in case something bad had happened to me. At times we had a deal that if no news from me during three days then contacts in the news media will be informed. And I talked with the police as well, who claimed they can do nothing until something does happen. In one country I was even arrested on fake medical charges, as being dangerous to public (a contagious disease they claimed), and kept in a hospital under heavy medication. I got suspicious and refused to take medication until I could see the results of the tests. And then I got miraculously cured without any further medication regardless that I was told I've got only about two years to live if I stopped taking the medication. By the way, the side effects of the medication could have been getting permanently blind and deaf, for exchange of staying alive, and that was one reason I stopped the treatment preferring rather to die than to live that way. I explained to the treating doctor my preference to die very calmly and confidently, which probably prompted leaving me alone. In yet another country I was surrounded by anti-terrorist unit with machine guns simply for an attempt to take out my own money (salary of my last work) from my bank account. The list can go on, but that's the material for another book, so I don't mention particular

countries and places here. All that happened before I began writing the book on the cryptography method and barely anyone knew I had developed such writing. It was still just one of my hobbies, till I decided it's the time to put it all down on paper and to share with others.

Last few years my life has been more relaxed, apart of the economically tough times with no job or some occasional jobs with a very little pay. But it still happens that some things remind me about the all seeing eye, or 'big brother' watching. Once I had some monthly available international calling time left on my cellphone, which would have expired in few days by the conditions of the contract. I could have called for an hour to almost any country in world, so I decided to call my friends whom I haven't talked for a while, to find out how's the life going. When I had called perhaps to 10-15 friends in different countries I got an incoming call which looked like Canadian or US number. The man asked what is happening, and on my question whom I was talking to got the response that he's from Interpol and needs to know why am I calling to so many countries. Again, funny as it was, I said that I got many friends, on which he warned me to watch out and hanged up. As I later joked on a social site, at least I have the number now to call when my cellphone will be lost or gets stolen.

I have gone through several refugee camps in different parts of the world, living unstable life ready to be arrested and deported at any time. And it had happened many times. Since having been imprisoned in several countries as the result of my political life, for many weeks and months at time, it was another reason why I wasn't taking up the task to write a book little by little -- I was never sure I could finish the book before yet another issue, which would have restricted my work for a long time. So, writing a book in a compressed manner, in a short and calm period in a stable place, was the only reliable option for me, to get the job done.

Here's where I did big part of updating and editing of the book, in France, while the friend who permitted me to stay there was away overseas..



Maybe an irrelevant information, but to those of you who like to learn how a book is written, these pictures may give a better sense of it. As a poor man ever I also worked on the book in places where I had to go to get some free food provided by social services or by volunteers of different organizations. When I had a thought that needed to be written I

didn't waste my time till I got back to my temporary home. Often these places were very noisy, but I got around the problem by putting on earphones with some music on the loudest possible level and switched my mind off the outer world, working on the book.

Here's yet another place, at another friend in France, where I lived several times for many months, and also did some editing of the book. Although that friend didn't want me to stay there just writing the book. I was doing some of my official paper work and thus I used the time of waiting responses for research and writing..



My style of writing a book is to have a total freedom. Not that I need to work on it 12-15 hours a day, which I did for the general layout of the book in the beginning, but I need to have the freedom to jump onto writing at any point in time when an idea needs to be written down, not waiting for it half a day till the possibility arrives at last. Because in a busy life by that time when I can get to writing again some of the thoughts may have been lost and it's a waste of time to try to figure out what exactly I was thinking back then while the new thoughts need to be written down in the short moments available.

Same with my first book which I wrote almost 20 years ago for students in my school of martial arts. Back then I had a family, two small children, a private school of martial arts, several businesses, and was also in almost constant move. I managed to take me off all the daily hassle for two weeks leaving other people in charge of urgent things in business and advanced students in charge of training, locked me up in the office and concentrated on writing the book on philosophy of thinking. I published the book on my own expense just a few months later, after a quick editing in about 10 days. My first book wasn't written in English by the way, I could barely understand few words in English back then. It was a kind of manual for students of martial arts, not to repeat the philosophy to every new student -- it was easier to give the book first and then to explain further if something was unclear. The book was translated the same year by someone I hired for it, but I never had the chance to approach publishers for the translated English version -- my political problems had just began getting serious and I ended up in exile where I am to this day. The English version of my book on philosophy of thinking has been free to download on my different websites for over ten years.

During editing and updating of this book, on my cryptography method, two and

half years later after the first version was completed in 2013, I was working perhaps only 4-5 hours a day and few days a week, for one month, but I had the freedom to do it at any moment when a thought was ready to be expressed. It happened sometimes that I took out my tablet in the middle of a busy street and made notes for including later in the book. In the moments where I had some meetings and couldn't keep writing I still had the freedom to make notes at any time, while during most works which I had in my life it wouldn't be possible.. maybe once/twice, but not many times again and again.

Before finding the right place and time I couldn't just sit around writing a book while having no stable or sufficient income, but I used that time to develop my methods of cryptography. I also had fun with an 'artwork' -- the hieroglyphs written on the photos taken by Hubble telescope downloaded from public and free to use sources. The 'artwork' (chapter 27) is from the year 2011, when I was living in a refugee camp. It was done on an old laptop given to me as a gift in times when I was locked up in a hospital as a supposedly contagious person and danger for the society. After getting free from hospital I spent some time in the camp recovering from all the medication. Working on cryptography and 'art' was a way to forget the condition I was living in.

When the updated book was ready for publishing in about a month's time, in 2016, I began having ideas how to make the thing even better. One day, another day, and I came to realization that this was not the end yet -- there were many ways to improve the book. The important thing was that the book was good enough to approach publishers, so if faced with difficulties in continuing to write I could have stopped at it what had already been done. In any case I had to write on my blog..

2016-06-21, publishing must wait.

it seems that i've got too deeply into the writing mood. few days ago thought that the book on cryptography is ready but since then i began writing several more new chapters. some new chapters should have been included already in the layout of the book before editing/updating but somehow didn't cross my mind. more work to do to get the thing closer to a perfect handbook on the cryptography method. i'll keep working on the book while on the road as i must move on.. whether i like it or not.. pressures of life. one chapter in the book also tells about how and in which circumstances the book was written.. you must wait till it gets finalized and published, then you get the picture of my crazy life in some sense.

That's how some of the work was done, in the conditions of almost total freedom..



Instead of a bulky laptop i switched to tablets which lasted on battery for many hours and I also succeeded to buy an expensive large battery to recharge the tablets on the go. I could leave the battery to charge in some place for a day when needed. External bluetooth keyboard and bluetooth mouse were also expensive but absolutely necessary to work on the book.

Thus the book that you're reading now is not the result of just a few months of work, but about half a year worth of improving it little by little, whenever time for it. And of course the result of over 30 years of fun in developing the encryption method.

After writing the initial version of the book in 2013 I've been also promoting lectures/seminars on the topic. Unfortunately, as I lived in the countries where people consider lectures in English not as prominent as in their own language, I didn't have much success. Maybe also the topic is too difficult for a larger interest in public, especially not in the main language of communication. Most importantly, my financial means to promote the lectures/classes/seminars were almost inexistent -- I created fliers, printed and copied them whenever some little money available from occasional one-two day jobs, rarely longer, and shared them on the street. Sometimes I got a possibility to print/copy for free and I used every opportunity. I also tried to find free local online services for such information, which would be popular enough. On some popular online sites for private buy/sell/teach and such advertisements my ads were not allowed to be published and the money I paid was returned. The excuse was as "not compatible with community standards" or something like that. Also my tablet's virus protection and anti-spy apps began warning me when I installed the apps for those popular online sites. Thus I stopped attempting to promote my cryptography on these sites.

Here are some advertisements which I did while having temporary longer stops between the periods of almost constant traveling..

In Switzerland, 2014..



By the way, in Bern, Switzerland, I went to every private language school I could

find -- over ten of them -- and proposed to include my classes of handwriting cryptography in their lists of classes. None of them was interested, even if they looked curious in the beginning. I have no idea why people are scared of cryptography.. that's my impression.. people are scared.. curious, but afraid to get involved in any ways.

That's entirely another topic, how people are enslaved in their minds, related to the politics and fear-mongering. Sure you already got an idea about my position from my political activism, but here's something I wish to share: I have come to conclusion that generally (not individually) people deserve the life they get. Submission to authority as a nation or a large group of people isn't a one time decision, but the personality traits and actions (or non-actions) of them as a whole, in majority, thus they deserve the results.

Here's some of my street promotion in Italy, 2015..



These above are photos of the postcards which I painted, to promote the method. After few months of 'selling' I gave them all away for free because there was no point to carry around packs of them while the 'business' wasn't picking up. It's difficult to call it an art but I had to do something. At least I tried. People were curious but very seldom someone was willing to give money for a postcard where nothing could be understood what is written on them. In the back of the cards there was written the address of my website which clarified a lot to those willing to seek further information. By now the web domain I used is discontinued (at the time of editing the book) but hopefully I will soon earn enough and regain ownership of the domain name. Then I'll be able to better promote the cryptography method and the book.

Handwritten symbols in a stylized, abstract font.

26 Future of the method.

The method of handwriting allows to separate it into different areas of application, also for a different purpose than encrypting data. And it also allows to make use of the cryptographic part of the method without using hieroglyphs.

26.1 Writing without encryption.

When there's enough interest in different parts of the world then hopefully one day there will be groups of people who will be able to make another good use of the method of writing. It doesn't have to be cryptographic but quite the opposite – creation of a writing system for representing any language with the same symbols, for easy learning to speak any chosen language.

No language exists which is capable of representing all the sounds from all the existing languages without significant modifications and additions to the existing characters in any chosen alphabet. As this writing method (with glyphs) allows representation of symbols in millions of ways I'm quite confident it will be capable to tackle any phonetic challenge with an ease, simply needs to be worked out by linguists. There are not so many ways to produce a sound, thus by creating the rules of representation of sounds within a single complex symbol (glyph) it is possible to have covered all sounds without the need to use many letters in combination. I imagine the set of a few hundred symbols will cover all spoken languages and maybe even less symbols is needed if used together with attached characters for sound modification.

26.2 Encoding without hieroglyphs.

The encryption method can be used without the hieroglyphs, in an art for example – as I have mentioned already in the section 'art method' (chapter 7.3c) messages can be hidden in lines, curves, dots of a picture that has no resemblance to hieroglyphic script at all, not like in my 'artwork' in this book (chapter 27). The principles of encryption can well be used in steganographic way in many fields of life, encoding messages in sight of everyone while only those who know the key can get the message. Not sure that I myself be taking these steps in a serious manner, just having some fun, but these are the options to go further to anyone who would want to -- to develop specific techniques for encrypting messages hidden in plain sight, while others cannot see there nothing else but just a painting or drawing, or just a regular text message while the encrypted message is hidden in cryptic symbols added to the text. On the pages to print/copy (chapter 28) I give an example of writing cryptic messages hidden in a regular text.

I did try to find interested people in that field. In 2015 in Italy I talked to many gallery owners in an attempt to get in touch with painters who would like to collaborate with me on inserting cryptographic messages into art. I failed to find interest in it. Myself

I'm not a painter, very far from it, thus I know my limitations and won't be taking this field up without a professional assistance. One day, after some specific training I may take up artwork in a sophisticated level as well, who knows. So far someone else could try, test and write a book about it, with examples of code creation, different possibilities of hidden messages in paintings, with basic principles everyone should follow and so on. Then it will be easier to put the method into practice.

For example in front of a safe can be a picture with encrypted code to open the safe. If that encryption art has to be developed from scratch, using only the principles given in this book, it will be possible but perhaps too time consuming for a practical use. When there will be a manual to follow, similar to the hieroglyphic writing, then it will be worth consideration by many.

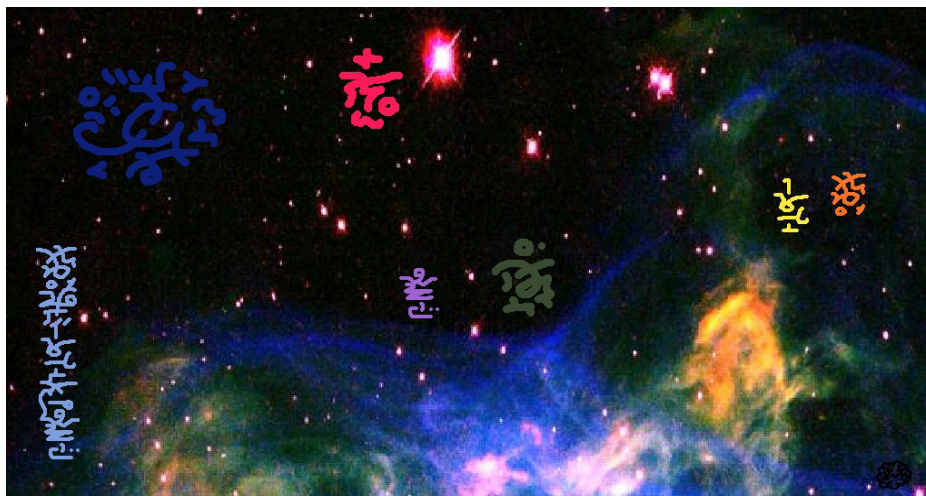
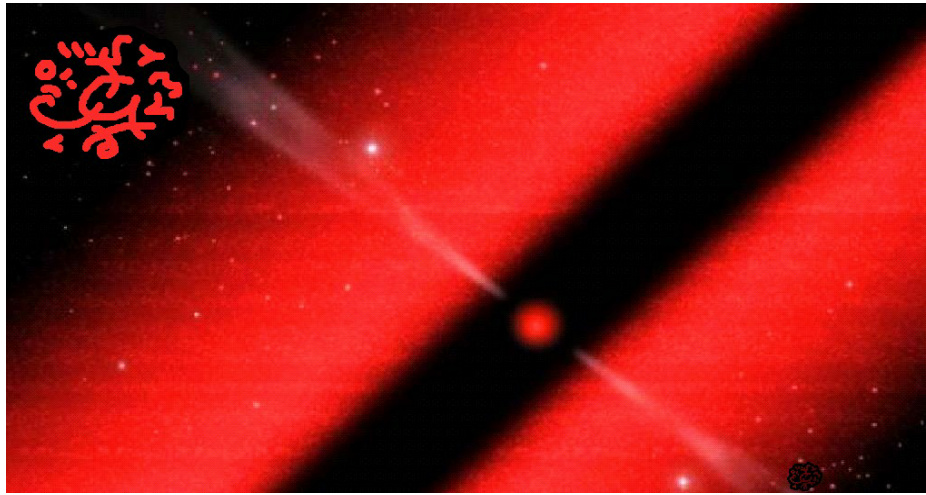
I'm far from thinking that I have already thought of and written about all of the possible ways of using the writing method and the encryption method. I'll be glad to learn from others the new ways of using the techniques of writing/encryption, so whenever free time I can take up a new journey in developing the methods further. The more challenging the task will be the better for keeping brain active and in good health.

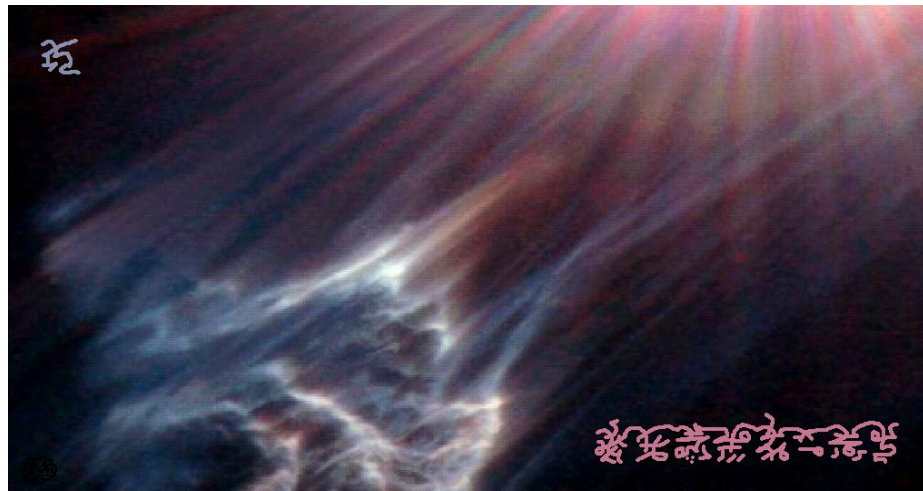
26.3 About the next challenge.

Perhaps there isn't much to improve in the advanced handwriting cryptography method as such, because in the end everyone must come up with the details on his/her/its own, to make the personal code unique. The challenge thus is in making the method more convenient, to be able to write as fast as regular unencrypted handwriting, but not compromising the security of the code. There may be some tricks that I haven't come up to as of yet, which I may discover one day. So far the method requires quite a bit of mental work, but I see it not as a fault -- it's rather an additional benefit. The challenge still is out there and I'm giving to it some thought from time to time.



27 Artwork.





The hieroglyphic art on the photos of the universe has no encrypted messages in them, nor the random samples of the glyphs in the end of many chapters of the book, which are there as well just for a design, kind of art for decoration.



28 To print / copy.

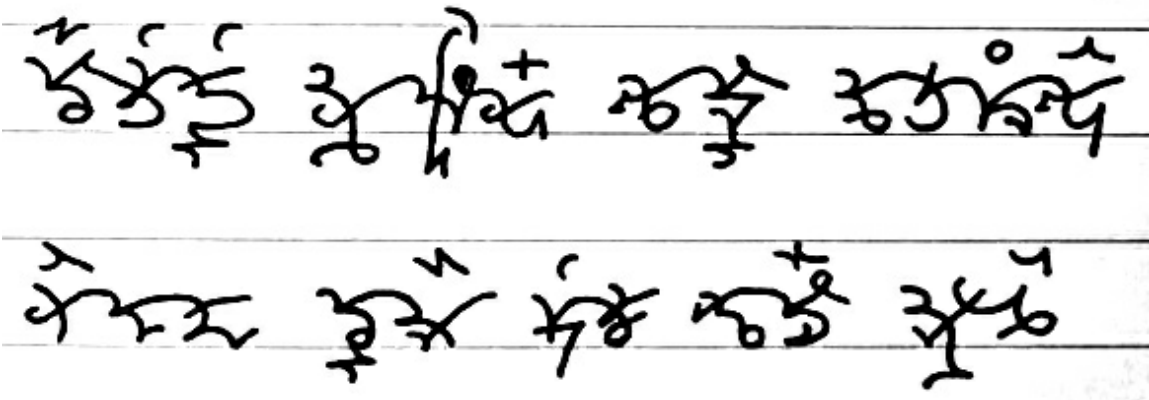
R	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	a A	b b B	c	f	i	3	/	%	°	?	.	+	u													
d	d	e	h	2	+ * #	" <	7 u	u	,																	
e	- + =	({ [
f	u	u																								

R	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	a A	b b B	c	f	i	3	/	%	°	?	.	+	u													
d	d	e	h	2	+ * #	" <	7 u	u	,																	
e	- + =	({ [
f	u	u																								

The image displays several handwritten cipher systems on a grid background. The systems are organized as follows:

- System 1:** A 4x6 grid of handwritten symbols. The first row contains two groups of symbols labeled 'A' and 'B' at the bottom. The rest of the grid contains various symbol combinations.
- System 2:** A 4x6 grid of symbols. The first row has two groups labeled 'A' and 'B'. The second row has two groups labeled 'a' and 'b'. The remaining rows contain symbols. Arrows indicate the direction of writing (upward and downward).
- System 3:** A 4x6 grid of symbols. The first row has two groups labeled 'A' and 'B'. The second row has two groups labeled 'a' and 'b'. The remaining rows contain symbols. Arrows indicate the direction of writing.
- System 4:** A 4x6 grid of symbols. The first row has two groups labeled 'A' and 'B'. The second row has two groups labeled 'a' and 'b'. The remaining rows contain symbols. Arrows indicate the direction of writing.
- System 5:** A 4x6 grid of symbols. The first row has two groups labeled 'A' and 'B'. The second row has two groups labeled 'a' and 'b'. The remaining rows contain symbols. Arrows indicate the direction of writing.

On previous 2 pages: R-matrix and C-list; below: example of encryption ..



Drawing exercise of script parts (refer to chapter 8) -- inline drawing:



Drawing exercises (also the script above can be used for drawing over it):



On the next 2 pages: exercises in bigger scale, with/without the assisting lines..

<p>          </p>	<p>          </p>	<p>          </p>	<p>          </p>	<p>          </p>	<p>          </p>	<p>          </p>	<p>          </p>	<p>          </p>	<p>          </p>
---	---	---	---	---	---	---	---	---	---

Samples of alphabets/hieroglyphs/shorthand. (NB! I have no idea what's written in the scripts -- I have only selected samples of writing styles, for visual help in creation of your code). Part 1, versions of horizontal writing..

Handwritten script in a cursive style, possibly a form of shorthand or a specific dialect, consisting of two lines of text.

Handwritten script consisting of two lines of text, featuring a mix of letters and symbols, possibly representing a specific code or shorthand.

Handwritten script consisting of two lines of text, featuring a mix of letters and symbols, possibly representing a specific code or shorthand.

Handwritten script consisting of two lines of text, featuring a mix of letters and symbols, possibly representing a specific code or shorthand.

Handwritten script consisting of two lines of text, featuring a mix of letters and symbols, possibly representing a specific code or shorthand.

Handwritten script consisting of two lines of text, featuring a mix of letters and symbols, possibly representing a specific code or shorthand.

Handwritten script consisting of two lines of text, featuring a mix of letters and symbols, possibly representing a specific code or shorthand.

۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰
 ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰
 ۳۱ ۳۲ ۳۳ ۳۴ ۳۵ ۳۶ ۳۷ ۳۸ ۳۹ ۴۰
 ۴۱ ۴۲ ۴۳ ۴۴ ۴۵ ۴۶ ۴۷ ۴۸ ۴۹ ۵۰

۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰
 ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰
 ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰
 ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰
 ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰
 ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰
 ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

۱۱ ۱۲ ۱۳ ۱۴ ۱۵ ۱۶ ۱۷ ۱۸ ۱۹ ۲۰
 ۲۱ ۲۲ ۲۳ ۲۴ ۲۵ ۲۶ ۲۷ ۲۸ ۲۹ ۳۰

Handwritten text in a highly stylized, cursive script, possibly representing a specific cipher or code.

Handwritten text in a highly stylized, cursive script, possibly representing a specific cipher or code.

Handwritten text in a highly stylized, cursive script, possibly representing a specific cipher or code.

Handwritten text in a highly stylized, cursive script, possibly representing a specific cipher or code.

Handwritten text in a highly stylized, cursive script, possibly representing a specific cipher or code.

Handwritten text in a highly stylized, cursive script, possibly representing a specific cipher or code.

Handwritten text in a highly stylized, cursive script, possibly representing a specific cipher or code.

Handwritten text in a highly stylized, cursive script, possibly representing a specific cipher or code.

Handwritten text in Arabic script, appearing to be a list or a set of instructions.

Handwritten text in Arabic script, appearing to be a list or a set of instructions.

Handwritten text in Arabic script, appearing to be a list or a set of instructions.

Handwritten text in Arabic script, appearing to be a list or a set of instructions.

Handwritten text in Arabic script, appearing to be a list or a set of instructions.

Handwritten text in Arabic script, appearing to be a list or a set of instructions.

Handwritten text in Arabic script, appearing to be a list or a set of instructions.

Handwritten text in a cursive script, consisting of two lines of characters.

Handwritten text in a cursive script, consisting of two lines of characters.

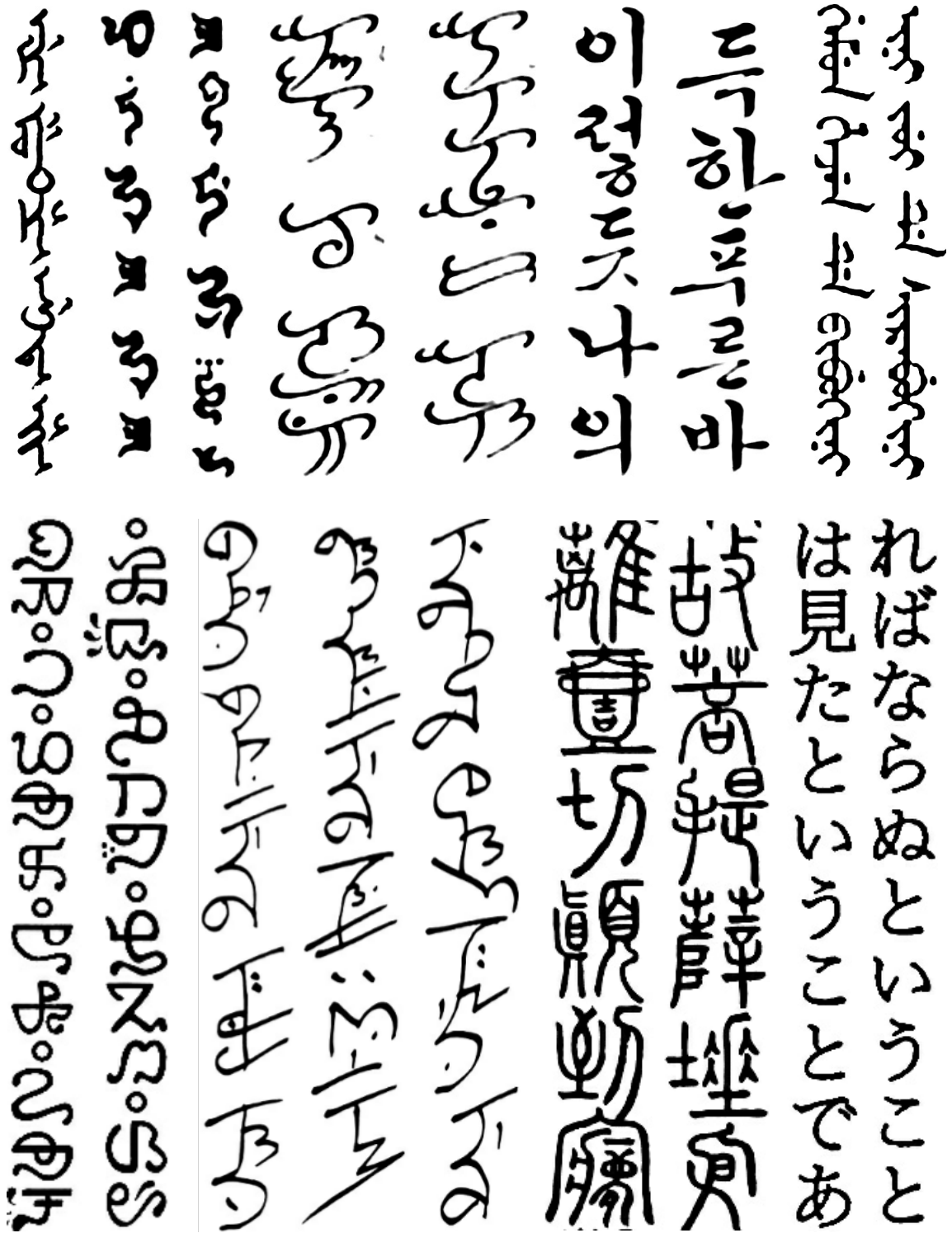
Handwritten text in a cursive script, consisting of two lines of characters.

Handwritten text in a cursive script, consisting of two lines of characters.

Handwritten text in a cursive script, consisting of two lines of characters.

Handwritten text in a cursive script, consisting of two lines of characters.

Samples of alphabets/hieroglyphs/shorthand. (NB! I have no idea what's written in the scripts -- I have only selected samples of writing styles, for visual help in creation of your code). Part 2, versions of vertical writing..



三 夕 田 田 五 十 十
飛 龍 故 墨 為 在 尾 後

目 羽 田 田 田 田 田 田
精 翰 者 如 古 洛 神 美 子

故 故 故 故 故 故 故 故
生 活 之 名 在 三 三 三 三

故 故 故 故 故 故 故 故
之 故 故 故 故 故 故 故

故 故 故 故 故 故 故 故
之 故 故 故 故 故 故 故

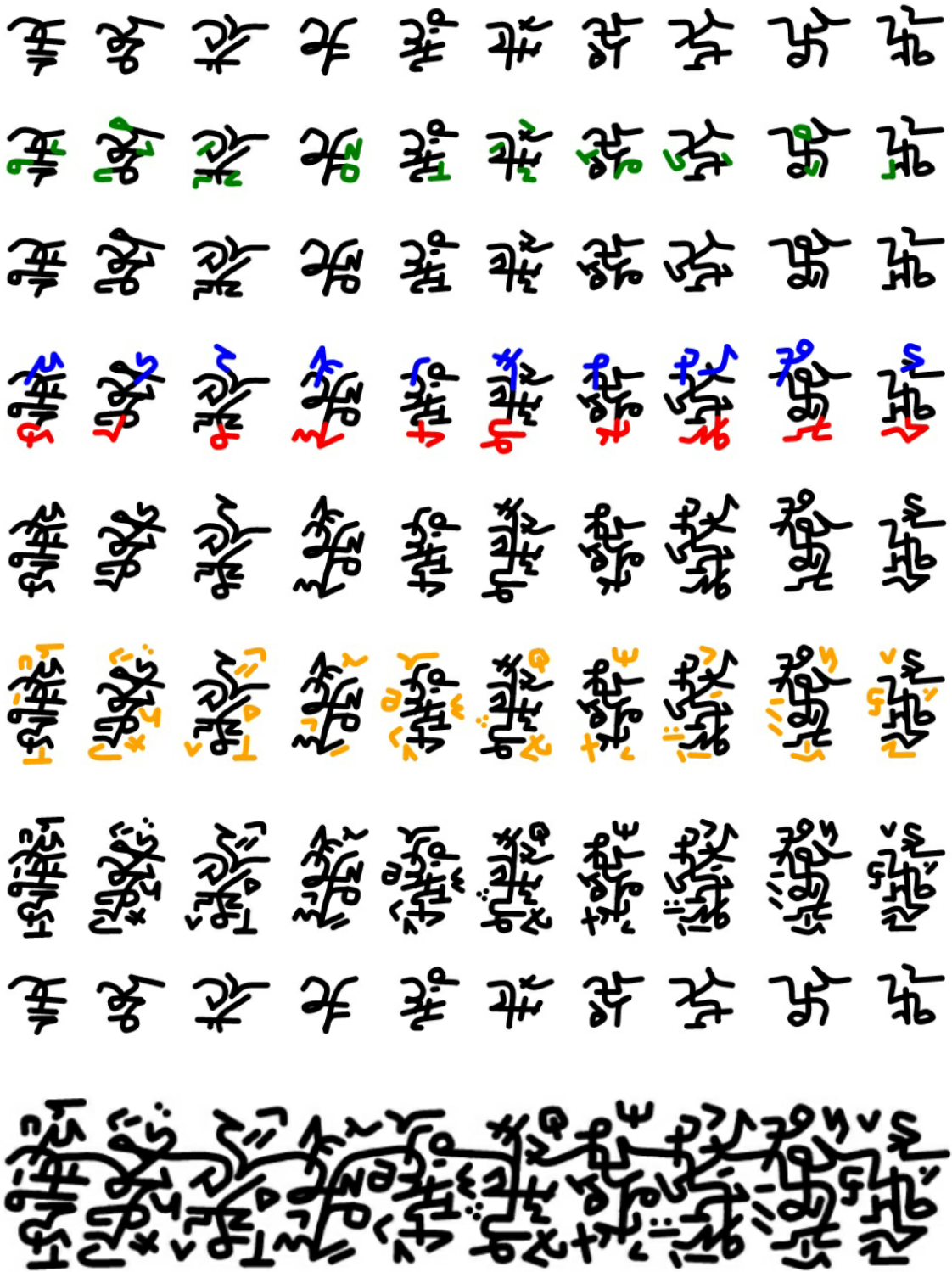
故 故 故 故 故 故 故 故
之 故 故 故 故 故 故 故

故 故 故 故 故 故 故 故
之 故 故 故 故 故 故 故

故 故 故 故 故 故 故 故
之 故 故 故 故 故 故 故

故 故 故 故 故 故 故 故
之 故 故 故 故 故 故 故

Encryption symbols - example of different parts (explained in chapter 8) ..



When there's a need to inform selected people in public places the way that only those people could get the secret message and nobody else will understand, use a steganographic variation of the encryption method. Others must not get suspicious of a secret message, thus you can't use the type of writing which others can't read. Write a public note and modify the letters like in a calligraphic writing. The hidden message will be in the extensions to the letters. Create the personal encryption method and teach it to the selected people. Once it's done you can deliver secret messages through publicity in journals or by other public means while the readers not dedicated to the code will suspect nothing. The letters in the public message shouldn't be used as a reference to the secret message but as attachment points to the cryptic symbols. You must make sure that the public message will have no errors of syntax prior to delivering to a publisher or it may be edited and the code may be seriously corrupted as the result of it. To avoid editing by publisher the text message should not be on a clear part of advertisement, a picture must be background of the message.

29 Get in touch.

Anyone willing to learn more about the advanced handwriting cryptography can get in touch with me to find out about upcoming seminars and lectures, and of course about new publications on the subject as well.

As I travel a lot changing my phone contacts, and as it will be physically impossible to respond every single mail, it will be more convenient for everyone to learn news about the cryptography methods on my blog..

youthextension.wordpress.com

I will post there news about upcoming events, under the section 'ENCRYPTED', and where suitable also in some other sections at the same time, for reference. All the up to date contact information will be there available, when necessary.



30 Legal Notice.

The author of this book have used his best efforts in preparing this book. The author makes no representation or warranties with respect to the accuracy, applicability, fitness, or completeness of the contents of this book. The information contained in this book is strictly for educational purposes. Therefore, if you wish to apply ideas contained in this book, you are taking full responsibility for your actions.

The author disclaims any warranties (express or implied), merchantability, or fitness for any particular purpose. The author shall in no event be held liable to any party for any direct, indirect, punitive, special, incidental or other consequential damages arising directly or indirectly from any use of this material, which is provided “as is”, and without warranties.

31 **Content.**

	page..
1 Introduction.	3
2 A bit of history.	5
3 Why handwriting?	8
4 Relevance.	9
5 The method.	10
5.1 Getting an idea.	11
5.2 Seeking the information.	11
5.3 Learning all the details.	12
5.4 Working on your own.	13
6 Basic principles of handwriting encryption.	15
7 Techniques of encryption.	19
7.1 Confusion / substitution.	19
7.1a Information carriers.	21
7.1b Noise and distraction elements.	25
7.2 Diffusion / transposition.	26
7.3 Other techniques.	28
7.3a Data points' method.	28
7.3b Mathematical method.	28
7.3c Art method.	28
7.3d Hiding method.	29
7.4 Writing styles.	29
7.5 Using switches.	30
8 Encryption symbols.	32
8.1 Base hieroglyphs.	32
8.2 Characters.	33
8.3 Additional signs.	34
9 Lists of symbols.	37
9.1 Classic text symbols (TS).	37
9.1a Alphabets.	37
9.1b Numerals.	37
9.1c Mathematical symbols.	38
9.1d Other symbols.	38
9.1e Highlighting and more.	39
9.2 Cryptic symbols (CS).	39
9.2a Diffusion symbols.	40
9.2b Correction symbols.	40

9.2c	Assisting symbols.	40
9.2d	Modifiers.	41
9.2e	Switches.	41
9.2f	Other tools.	42
10	How complex code is good enough?	45
11	Systematization and memorizing.	46
12	Sharing and personalization.	49
12.1	Sharing with one person.	50
12.2	Sharing with several people.	51
12.3	Temporary sharing.	51
12.4	Urgent sharing.	52
12.5	Key delivery.	52
13	Securing the method for recovery.	54
13.1	Recovery of encryption method.	54
13.2	Recovery of encrypted data.	55
14	For developers.	56
15	Application.	57
16	Example of glyph parts.	59
17	Examples of code creation.	61
17.1	Basic code.	61
17.2	Medium level code.	64
17.2a	Matrix of symbols.	64
17.2b	Encryption symbols.	66
17.2c	The creation process.	67
17.2d	Description of rules.	74
17.2d - 01	Basic rules of the code.	75
17.2d - 02	Base cryptic symbols (BCS).	77
17.2d - 03	Attached cryptic symbols (ACS).	78
17.2d - 04	Separate cryptic symbols (SCS).	79
17.2d - 05	Creating additional rules.	81
17.2e	Writing sample.	82
18	Exercises.	85
18.1	Attention exercises.	85
18.2	Drawing exercises.	86
19	Commercial uses of the cryptography method.	88
19.1	Cryptic messages project.	88
19.1a	Guessing of messages.	89
19.1b	Creating a lottery.	90
19.1c	Promotional ads.	90
19.2	Cryptic signatures.	91

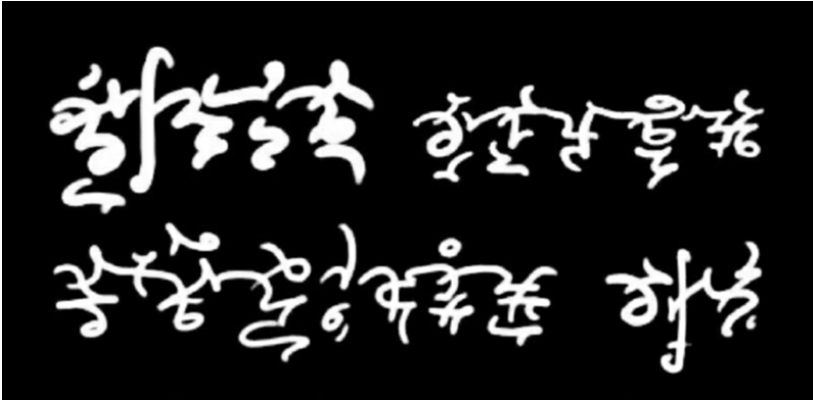
19.3	Cryptic orders.	91
19.4	Other commercial uses.	92
20	Writing music.	93
21	Advanced games for training.	94
22	Cryptography in social life.	96
23	Cryptography in gambling.	97
24	The highest level method.	98
25	Work on the book.	101
26	Future of the method.	111
26.1	Writing without encryption.	111
26.2	Encoding without hieroglyphs.	111
26.3	About the next challenge.	112
27	Artwork.	113
28	To print / copy.	115
	R-matrix	115
	C-list	116
	Drawing exercises	117
	Samples of world writings, horizontal	120
	Samples of world writings, vertical	126
	Encryption symbols, example of different parts	128
	Example of steganographic encryption	129
29	Get in touch.	130
30	Legal notice.	131
31	Content.	132

You can also keep an eye on my weblog:

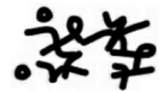
youthextension.wordpress.com

..when time for it then there can also show up some encrypted messages with valuable information related to the topic.





For several millennia, almost since the birth of writing, it's been the dream of rulers and warlords to exchange letters without worries that someone could read their exchanges when intercepted. In the age of internet when almost anyone having proper hacking tools can read your messages, this dream of ancient powerful has found its way into the minds and hearts of common people. And now the dream has come true - the writing system which anyone can learn and customize the way that no unwanted parties can read, has been created. The new era of writing has begun. It's the era when people can write with no worries about disclosure of their private messages.



—

The photo on the second page of the book: in Paris, 2008. Photos, pictures, drawings and text, Copyright © 2008-2016 by Alex A. O. Kobold. Copyright exceptions: the photos of Hubble telescope from public and free to use sources, as base for the hieroglyphic art, chapter 27; the pictures of alphabets/hieroglyphs/shorthand from Wikipedia and other free to use sources, chapter 28.

The book “Advanced Handwriting Cryptography”, Copyright © 2016 by Alex A. O. KOBOLD, All Rights Reserved.