My name is Peter Lablans. I am an inventor and engineer with a M.Sc. in Electrical Engineering. I am a prolific inventor in the field of camera technology and machine cryptography. I have worked as a Patent Engineer for a Patent Attorney and assisted in successfully prosecuting hundreds of patents in areas such as image and signal processing, optics and medical technology. During my career I was also a digital telecom engineer, a college professor, a sales and marketing executive and science/technology diplomat. I am the named inventor on a portfolio of dozens of US Patents. My patents and patent applications are cited numerous times as prior art in Office Actions and IDS documents of other inventors. My machine cryptography patents are assigned to Ternarylogic LLC, a small R&D company I founded.

This submission is in response to RFC PTO-P-2022-0026-0001 on Patent Subject Matter Eligibility Guidance. I am the named inventor on several patents directed to Machine Cryptography. Machine Cryptography, like some other important technological fields, applies mathematical modeling, but implements exclusively in machine instructions. In fact, without a machine or computer, the technology of machine cryptography (and other machine technologies) would not exist.

This submission is in support of patent eligibility of machine cryptography as already applied by the USPTO in its Guidance on Patent Eligibility as provided in Example 41. However, the reasoning of the USPTO for eligibility seems to focus on "integration into a practical application." This submission argues that the field of Machine Cryptography is strictly and exclusively directed to machine technology and should be considered to be patent eligible *per se*. It also recommends that a positive determination ('the claimed invention is directed to a machine') when appropriate, is included in a Notice of Allowance.

**Patent Eligibility Considerations in Cryptography Related Inventions**

The example 41 is related to the original RSA patent 4,405,829 to Rivest et al. (hereinafter "Rivest'). However, the text of the claim as used in example 41 is different from the actual claim language in Rivest.

Despite the differences, I believe that the analysis of example claim 41 as being patent eligible correctly applies to Rivest. I also believe it represents, correctly, the general patent eligibility of machine cryptography directed claims. One reason for being decided patent eligible in Example 41 was that the claim is "Integrated into a Practical Application" and was not directed "… to the recited judicial exception" which is a mathematical formula or calculation. However, I also believe that cryptography as claimed is machine cryptography and is ***fundamentally directed to machine operations*** and not to mathematics *per se* and accordingly is and should be patent eligible.

**If only we could have a working calculating machine**

In 1867, Frederick A. P. Barnard, a mathematician and the president of Columbia University in New York, served as a judge at the Exposition Universelle, a world's fair held in Paris. There he saw a calculating machine for the first time. In a report on the fair, Barnard wrote:

"To most persons the process of calculation involves a species of mental labor which is painful and irksome to the highest degree; and to such, no part of their educational experience recalls recollections of severer trials [sic], or of burdens more difficult to bear. That this toil of pure intelligence – for such it certainly seems to be – can possibly be performed by an unconscious machine is a proposition which is received with incredulity; and even when demonstrated to be true, ***is a phenomenon which is witnessed with unmingled astonishment***." (from https://americanhistory.si.edu/collections/object-groups/calculating-machines )

**Modern Cryptography is Machine Cryptography**

I believe that besides being "integrated into a practical application", patent claims such as example claim 41 are directed to machine operations *per se*. They are not directed to the recited judicial exception, being the mathematical expressions that are often recited in cryptography claims.

Modern cryptography is machine cryptography. Machine cryptography is a machine technology. It applies often (but not always) machine arithmetic, which is also a technology. Machine cryptography is a basic technology applied to create data security of machine processed data that is inherently not present in those data. Machine cryptography modifies data signals by applying a component that is kept secret from being accessed. Recovering the original data from the modified data without the secret component requires greater machine effort, preferably unreasonably greater processing effort, than with the application of the secret component.

How 'good' or secure a specific machine cryptography invention is, depends on the structure (or logic implementation) of the machine cryptography.

Machine cryptography finds much of its origins in electro-mechanical rotor machines. Nobody, I believe, has ever argued that cryptographic rotor machines like Enigma, Hagelin, SIGABA are patent ineligible because of being directed to an abstract idea.

Another invention that still permeates cryptography today, is the Vernam cipher, which was awarded a US patent (US 1,310,719) in 1919. A Vernam cipher combines two data streams via, what now is called, an XOR function. Vernam realized that combination through relays devices using Baudot codes. Again, nobody (as I know) has argued that the Vernam machine as claimed is directed to an abstract idea. In fact, what Vernam does (especially nowadays in modern machine cryptography), is directed to technically combining or "enter' data into a cryptographic system by way of an XOR function. It is an essential technical component that makes modern machine cryptography (like SHA-256 as well as AES-256) possible.

**The Importance of Machine Cryptography**

Machine cryptography forms the basis of cybersecurity, including encryption/decryption, public key infrastructure (PKI), authentication and digital signatures and is the lifeblood of almost any kind of data exchange over the public Internet. Without machine cryptography, and more importantly without innovation in machine cryptography, security of data exchange in the very near future is under threat. Cybersecurity relies on an adequate technological implementation and realization of the machine cryptography.

**Machine Arithmetic**

Machine cryptography relies heavily on the technology of machine arithmetic. Machine arithmetic or computer arithmetic in binary form and as applied in cryptography or elsewhere is fairly young and intrinsically connected with physical switching devices. It is based on a model of physical switching operations, which is connected with Boolean Algebra. Despite what many people believe, Boole did not invent computer arithmetic. Binary computer arithmetic as an electrical technology was developed at multiple places independently in the first half of the 20th century. Arguably the most influential development was the paper written by Claude Shannon as his MIT master of science thesis in 1938. In that paper Shannon makes the connection between physical states of (electromechanical) switching devices and logic states. In his thesis on page 4, Shannon teaches: "This variable, a function of time, will be called the hinderance of the two terminal circuit a-b. The symbol 0 (zero) will be used to represent the hinderance of a closed circuit, and the symbol 1 (unity) to represent the hinderance of an open circuit." In effect Shannon technically applies the measurable impedance of a circuit as the tool for creating a machine adder. The impedance somewhat close to zero ohm, is represented by symbol 0 and the impedance of an open circuit, often called infinite, is symbolically represented by 1.

It cannot be repeated enough and I will say this repeatedly herein: the 0 and 1 or any logical representation in Shannon and in any computer do not actually exist as "numbers" in a computer. They are names or labels for technical states of a machine with structure. Actually, numbers do not exist at all inside computers. Only signals, or states of electronic and electrical devices exist.

The 0 and 1 represent a model, which may be called a Boolean model, of a physical switching device. Other models are available of electrical devices. Such as an electrical circuit model. They may describe an electrical device, such as a switch, in different ways. For instance, a circuit model of a switching device may apply differential equations. This allows one of ordinary skill to compute circuit parameters and input and output variables. It should be clear that an electrical circuit, despite the fact that it is described by differential equations, does not actually perform differential equations. In a similar vein, a trajectory of a projectile may be described by differential equations. However, the projectile does not perform those differential equations. Switching device may be described by logic equations, but they do not perform logic equations. Physically they are placed in different physical states that may be represented by for instance 0 and 1 or False and True. These logical states themselves cannot be implemented physically. They have to be implemented in the physical realization they represent. Thus, whatever some people claim or believe, a computer does not perform logic or mathematics. It performs physical operations that may be modelled as logic or arithmetic.

An important aspect of a physical switching device that performs in accordance with an arithmetical equation is that it takes a finite and measurable time to perform the switching operations. This is demonstrated in a video of a relay adder as posted on Youtube at https://www.youtube.com/watch?v=k1hJoalcK68. This switching delay is an unavoidable aspect of machine arithmetic or any computer based operation, be it in electromechanical, electronic or even quantum mechanical form. It is a fundamental aspect of a cryptography machine to achieve security.

Again, numbers in a mathematical sense do not exist inside a computer, by which I mean a digital computer. In mathematics a number may be selected from the set of integers, natural numbers, rational numbers, irrational numbers, real numbers, complex numbers, Gaussian integers and the like. Computers do not manage numbers, they process physical states of matter that humans, by a physical machine transformation (into a printed character for instance), are able to recognize as a number. Digital computers are only capable of handling discrete physical states. This means that all machine arithmetic performed by a digital computer is limited by the characteristics (such as word length and memory) of the processor.

Machine arithmetic is limited by the switching capabilities of its components. Switching in present day computers is generally binary (or 2-state) switching. While it is popular to say that computers only use 0s and 1s, it is well known that these elements do not exist as such inside a computer. Inside computer circuitry, switching takes place in physical states such as voltages which are designated as LOW and HIGH as evidenced by data sheets of commercially available electronic "logic" components. For instance in 3.3V CMOS LOW is between 0 and .8 Volt while HIGH may be between 2 and 3.3 Volt.

Furthermore, implementing machine calculation, like binary adding in binary switching technology, has to deal with carry generation, which creates a switching bottleneck in the digital machine adder, as for instance provided in the Shannon thesis. This limitation is well known and may be addressed in different ways. A 'textbook' realization and implementation of an adder is the 'ripple' adder, where the machine determines for each cycle a carry bit and then moves to the next cycle, the total number of switching cycles being determined by the total of possible carry generations. There exist several ways to improve the speed of addition of 2 k-bit operands. They may be carry look ahead adders, carry skip adders, prefix adders and the like. These machine adders are faster than standard carry ripple adders, generally at the cost of extra components and/or some pre-processing. Most people, familiar with the mathematics of addition will be unable to recognize the functionality of such fast adders as they are a technical solution to a technical problem of processing signals. Similar issues exist around machine arithmetic of multiplication, exponentiation, inversion and the like.

Machine arithmetic has an abundance of these seemingly mathematical functional terms, which in fact are short-hand terms for technical operations on a defined structure. It is in that "technical" and "structural" sense that machine arithmetical operational terms in patents claims should be interpreted.

In my own cases, such as US Patent 11,336,425, on which I am the named inventor, I make sure to explain that mathematical functional terms are to be interpreted as structural machine implementations. This has a very sound and well explained technical foundation.

I base it on the computer design practices instilled in me by my late Professor Prof. Dr. Gerrit "Gerry" Blaauw, one of the three co-architects with Dr. Fred Brooks and Dr. Gene Amdahl, of the legendary IBM System/360, arguably the first true "general purpose computer."

Dr. Blaauw considered the design of a digital computer to be described at 3 levels: 1) the architecture, what a system programmer experiences; 2) the implementation or logic design and

3) the physical realization. One may adequately describe the computer on any of the three levels. Ultimately a computer design has to be realized in physical components. But a physical description, circuit wise, of a computer may not be helpful for a user to understand the computer. The implementation, or logic design of a computer, provides in logic statements or instructions (which refer to working hardware) the switching behavior of the computer. For instance XOR(a(1:8),b(1:8)) describe the bitwise XOR of 2 words (*a* and *b*) of 8 bits. When a computer successfully executes such or equivalent statement, a real circuit actually exists that executes this operation.

For a function in an ALU, an architect may want to define a function in an Arithmetic and Logic Unit (ALU) to have a general function OP1=GFADD2(OP1,OP2) which defines an addition over a binary finite field of two operands OP1 and OP2 of which OP1 is an accumulator to which a second operand OP2 is bitwise XORed. This may be part of the architectural instruction set of a processor. Any of the instructions executable on a processor, be it on architectural or implementation level, create a physical realization in a machine. In that sense executed computer instructions are not dis-embodied, they are an inherent (and physical) part of a working computer.

Dr. Blaauw provided and executed designs of computer circuits both on implementations (logic) and architectural level in the computer language APL as exemplified in his books Digital System Implementation and in Computer Architecture (in collaboration with Dr. Fred Brooks). I personally use Matlab for computer implementation design. However, it does not matter how a computer is instructed. I have applied APL, Python and C as well as Visual Basic to implement some of my inventions on a logic level. They all work correctly. The moment instructions are executed a working and physical and identifiable physical structure exists.

**The bitwise XOR as a data entry operation**

Theory of underlying aspects of cryptography is pre-dominantly mathematical theory. This is exemplified in titles of textbooks such as "An Introduction to Mathematical Cryptography," by Hoffstein et al. One should realize that all this theory is a mathematical model of what is implementable and implemented in discrete switching devices. It is comparable to theory of for instance digital signal filters. Which is ultimately a model of implementable structures (in Blaauw sense) on digital computers. Neither in digital filters nor in cryptography are the models directed to the mathematics *per se*, but rather to the desired modification of input signals by a processing circuit to achieve output signals.

The bitwise XOR of words of k bits can conveniently be modeled as an addition over a finite field GF($n=2^k$). The description for convenience in terms of finite fields causes a limitation in actual implementation of an "entry" switch to enable data to be entered into the computer machine to be further processed. Further analysis, as done in my patents, show that there are certain requirements related to a reversible n-state entry function that allow a whole range of novel and unpredictable unbiased entry functions that go way beyond additions over GF($n=2^k$).

I show in my patents that there are actually a great number (in the order of factorial n) possible reversible n-state entry functions that meet the requirement to provide reversible data entry of

signals represented by n-state symbols. My intention never was (and is still is not) to find a mathematical solution. My intention is and always was to create a digital apparatus that works correctly in an unpredictable way by modifying the switching devices (or functionally the switching functions). A switching function in this context maps directly to, and is equivalent to, a digital circuit.

**Security of Machine Cryptography**

The usefulness of machine cryptography is expressed the time it requires for a machine to perform a machine operation. Machine arithmetic is not abstract or conceptual. It requires and provides a measurable result. Conceptually, a Diffie Hellman process requires as a result something like $(g^b)^a$ mod-p = $(g^a)^b$ mod-p=$g^{ab}$ mod-p. This is conceptually simple and one of ordinary skill understands this almost instantly. However, in machine arithmetic it requires a certain amount of processor time to generate this result. One would want to generate it as fast as possible, which can be achieved by selecting the terms *g*, *a*, *b* and *p* small enough. One keeps terms *a* and *b* secret and only publishes $g^a$ mod-p and $g^b$ mod-p. To maintain a level of security, to prevent an attacker machine from easily determine *a* and *b* from published $g^a$ mod-p and $g^b$ mod-p, parameters *a*, *b* and *p* have to be as large as possible. Because if these terms are large enough, successful machine attacks are projected to take hundreds of years (if not longer) to break the security.

Hence, the security and thus usefulness of machine cryptography comes from the impossibility of attackers to reverse the machine processes with attack machines based on the available data. As computer technology advances, attack methods become more powerful and machine parameters have to be increased to maintain certain levels of security. For instance, the time that 512-bit RSA was considered secure has long passed and recommendations are to use at least 2048 bit and preferably 4096-bit RSA.

A downside of this is that machine computations for generating secure data become more time consuming as computers and attacks are improved and may create bottlenecks as levels of security have to be maintained. In addition, new technology forms a further threat to security, among which is Quantum Computing (QC). It is assumed that QC will make machine cryptography, in particular public key systems, fundamentally insecure. That is because certain computer arithmetical implementations on QC devices (like for machine factoring and machine discrete logarithm) are extremely efficient and fast, undermining Internet security.

Virtually all known machine implemented cryptographic methods and devices are insecure when using parameter that are too small. Security comes from too many possibilities to reconstruct the secret data from available public data. Usefulness comes from an unbalance in relative brief machine time to generate public data and recover secret data by authorized machine versus enormously long times required to derive secret data strictly from public data only. No matter how impressive or complex the mathematical models are, many of these seemingly complex mathematics oriented systems have been broken. A post-quantum key exchange system based on isogenies that had received high marks on PQ security was recently broken on a very classical PC system within a one or more days. Clearly the basic mathematics of isogenies, which is

brilliant in its theoretical scope, is not sufficient to guarantee security. The same applies to multi-variate systems, which have consistently been broken.

That is why in machine cryptography the structural aspects in a machine, and its performance in how fast a result is generated and how difficult it would be for an attacking machine to reconstruct a result, determine the usefulness of cryptographic machines.

It is further observed that for all modern cryptographic machine operations that are deemed secure, the performance parameters are such that no human or even unlimited number of humans could do what a machine does in a reasonable time. Machine cryptography is designed for machines and is intended not to be even roughly approached in performance by a human, be it with or without paper and pencil. In machine cryptography operations take place at a rate of at least millions if not billions of machine arithmetic operations per second, completely and utterly out of scope and capability of humans

**Directed to?**

My cryptography inventions are directed to novel digital circuitry to process data. The circuitry that is modified in part belongs to the class of computer arithmetic circuitry such as available in an ALU. The Blaauw doctrine/explanation guarantees that even when implemented in terms of computer arithmetic, or functional terms, if the statements are executable then there is an underlying physical structure. Circuity is modified in accordance with a specific structure (which I call the FLT or Finite Lab-Transform). This causes digital circuitry to modify signals in an unpredictable way. The intent, (directed to) is to increase the security of signal exchange between machines. Furthermore, the computer per se in my inventions has been improved by now having a structure that was not available prior to the invention.

Furthermore, no claim is made on the mathematics *per se* of the invention. Anyone is free to apply the FLT by hand or apply it in unclaimed computer applications.

**Guidance**

The current status of mathematics in a patent claim is that it creates an assumption of ineligibility under Alice. It is of course stunning that mathematics, the language of science and engineering, would be a reason for patent ineligibility. This should be resolved by Congress. The USPTO has to operate under Alice, Mayo, Benson and the like and has often no choice to apply these cases.

However, as the Guidance Examples show, the Office has some choice in how to apply the law. As an inventor I consider the Examiner Corps as “one of us.” That is, even though we may appear to be adversaries, we are all scientists, engineers, or technologists. All Examiners have at least a college level of training in engineering which includes mathematics. We all are aware (or should be) of the role of mathematics in engineering. The Judicial in general doesn’t have that insight and is boxed in by (non-engineering) rules and opinions.

The role of mathematics in engineering is well established. It appears in many roles. It is applied as a modeling tool that describes a structure or a phenomenon. Mathematics also is applied as a science *per se* with no (current) physical meaning and is abstract in its goal. As such it does not

describe any physical structure or phenomenon. However, human ingenuity often allows the use of novel mathematical concepts to be applied to predict and/or model physical phenomena. An example is complex Function Theory and Laplace Transforms which allow the design of electrical and electronic filters. Mathematical physics is another application of mathematics that allow the prediction and modeling of physical phenomena based on (usually very simple) initial mathematical conditions. For instance fairly simple principles, like Kirchhoff's Circuit Laws, have wide and important implications.

Finally, machine mathematics (which I call machine arithmetic) is applied as instructions (in expression form such as Finite /Discrete Mathematics and Z-transform and the like) to computers to perform a useful operation. Digital filters and signal processing, error-correcting codes, image processing, radar, control devices and machine cryptography are examples of specifically computer or machine directed technology. That is: those technologies would not exist and could not thrive without computers. Computer devices are the *sine qua non* for those technologies.

I have focused specifically on machine cryptography, because my own patents and patent applications are directed to that subject matter. But similar explanations of machine directed technologies can be provided for other computer based technologies.

I believe it is part of the USPTO's responsibility to provide inventors as well as Courts guidance on what technology patent applications and patent claims are objectively directed to. Especially because the Courts themselves provide no guidance what the terms "abstract idea" and "directed to an abstract idea" are supposed to mean and sometimes follow technologically incorrect allegations by infringers related to "abstract ideas."

When a party in a patent case argues that a "communication channel" is an abstract idea, all EEs know (or should know) that not to be the case. We could of course add in the specification further data like examples of "communication channels between two devices" such as wired and wireless channels formed by optical, galvanic or electro-magnetic connections. However, this adds little to the understanding of a "communication channel" as known by one of ordinary skill. In that sense the USPTO is able to act as an arbiter prior to going to court on what is abstract in engineering and what is not. This may prevent nonsensical interpretation of engineering and technology before the courts.

I personally would say that an inventor knows best what an invention is directed to, as it indicates an intent. An infringer, of course, may then argue that they know better than the inventor what the intent is or was and allege that "it is all an abstract idea." It that sense it may be helpful that an Examiner, based on factual information in the specification, and well established engineering practices, confirms that an invention itself is directed to a machine and forms patent eligible subject matter. It seems pretty much nonsensical to have a non-specialist with no experience in practicing or studying engineering to decide what an engineering discipline entails. An Examiner should be able to make that analysis in an objective manner on instruction and guidance by the USPTO leadership.

The USPTO is exceptionally well positioned to interpret the law and rules in terms of engineering concepts. This applies especially to what is real or abstract in technology. It allows

an Examiner to make a distinction between a mathematical expression and a physical structure or event described by the same mathematical expression. An Examiner can provide her reasoning why it is believed that a claimed invention is directed to a patent eligible idea. It also allows an inventor, supported by evidence in an affidavit, to dispute a machine oriented or abstract directed decision. It is of course absurd that in this time of cryptography, smart phones, Artificial Intelligence, Computational Biology and what else takes place in mathematical modeling, the use of a mathematical expression raises doubt about the validity/eligibility of a patent. It ignores completely the basic teachings of engineering science.

**USPTO Guidance on cryptography patent applications**

My own cryptographic claimed inventions are:
a) expressly directed to machines and machine operations
b) while they may include mathematically related terms, they are directed to machine arithmetic and machine arithmetic is a machine technology, not a human capability
c) useful, and improve security of data exchange between machines
d) are outside the scope of being done by humans even with paper and pencil
e) do not pre-empt human use of any of the perceived mathematics
f) provide novel, unconventional and non-routine functionality in computer devices
g) and yes, they are integrated in a practical application

It seems that when the Office during examination has to consider "integration into a practical application" it may already conclude that my claimed inventions are NOT directed to an abstract idea, but to a specific aspect of machine design (known under the somewhat unfortunate name of "machine arithmetic") or even more specific directed to novel machine (computer) functionality.

Ultimately, the Courts will decide if a patent claim is directed to an "abstract idea." However, a statement by the USPTO that the claimed subject matter and allowed claimed are directed to machine technology and not to an abstract idea will assist a Court to prepare an objective and science based decision on Patent Eligibility of a claimed invention. It will go a long way to strengthen a patent against ineligibility attacks and create robust and defensible patents.

An attacker (presumably) then has to provide evidence that the subject matter is NOT directed to patent eligible subject matter to override the Examiner. It is of course one thing to claim willy-nilly see-if-it-sticks that a communication channel is an abstract idea. It is a more difficult challenge for an attacker to have successfully argue why a communication technology that enables physical transmission of signals between two devices is engineering-wise an abstract idea (which it is not). It may also go a long way to remove mythical, irrational and unscientific thinking out of what essentially is supposed to be a scientific and rational bureaucratic effort of patent administration as envisioned by the Framers, who were convinced believers in the principles of Enlightenment.

Personally, I would like to see the 101 Eligibility requirement disappear completely. Methodologically and scientifically it is no longer of this time, certainly not in the form as it is applied. But that is not up to the USPTO. However, the Office and its Examiner Corps are well positioned to objectively assess if a claimed invention is directed to a machine or "directed to

pre-empting an abstract idea." In that context Example 41 already made an objective contribution. The USPTO can further strengthen the robustness and quality of a patent by including "being directed to a machine" as part of its Examination and included in a Notice of Allowance.

Peter Lablans
October 11, 2022
ip@ternarylogic.com