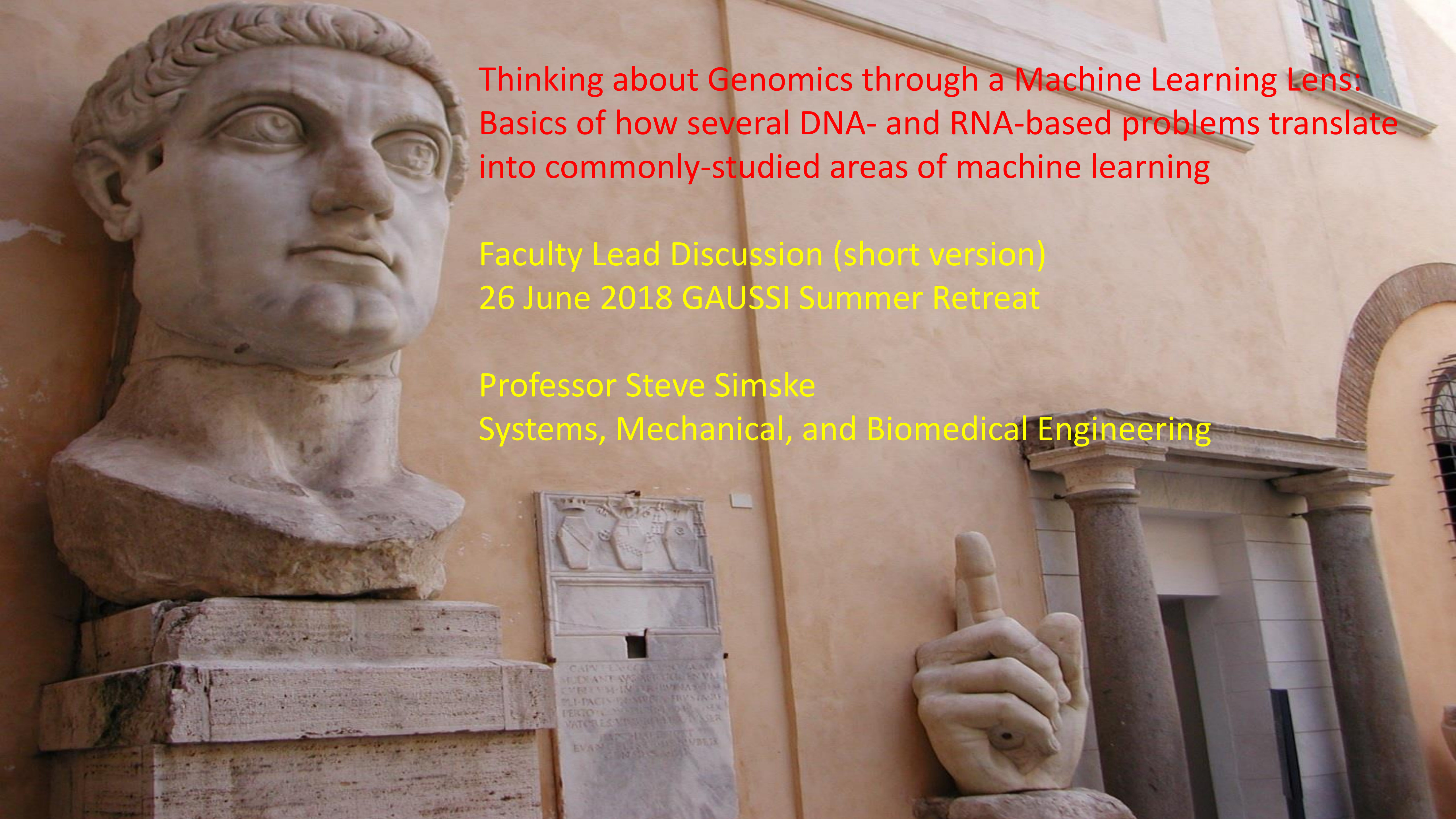


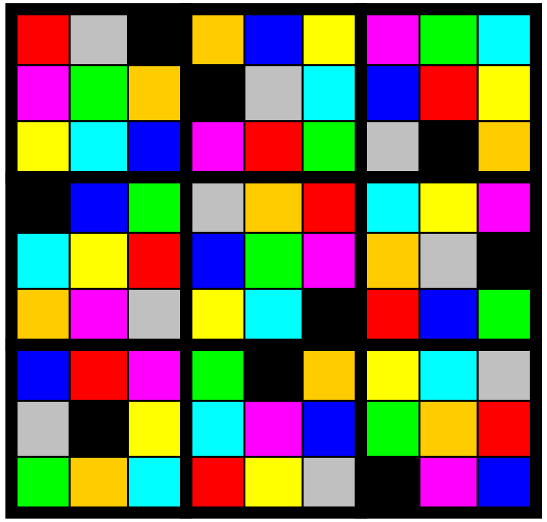
Thinking about Genomics through a Machine Learning Lens: Basics of how several DNA- and RNA-based problems translate into commonly-studied areas of machine learning

Faculty Lead Discussion (short version)
26 June 2018 GAUSSI Summer Retreat

Professor Steve Simske
Systems, Mechanical, and Biomedical Engineering

Outline for this lecture

1. Sudoku Security
2. Genetic Approaches to System Security



$$e = -\sum_{i=1}^{N} p(i) * log_2(p(i))$$

Overview

With the Sudoku, we explore a model for “Secure Transmission Using Structured Deterrents”, which means that the shared secret is, instead of telling the recipient how to decrypt the data, telling her how to organize the data upon receipt to generate dependent data

With genomic approaches, we can view the amino acid residue sequence to be one form of digital signature of the codon sequence, with the codon to residue translation being a trapdoor function

http://news.nextgendistribution.com.au/internet-of-things-the-data-is-coming-from-inside-the-house/

TRANSITION

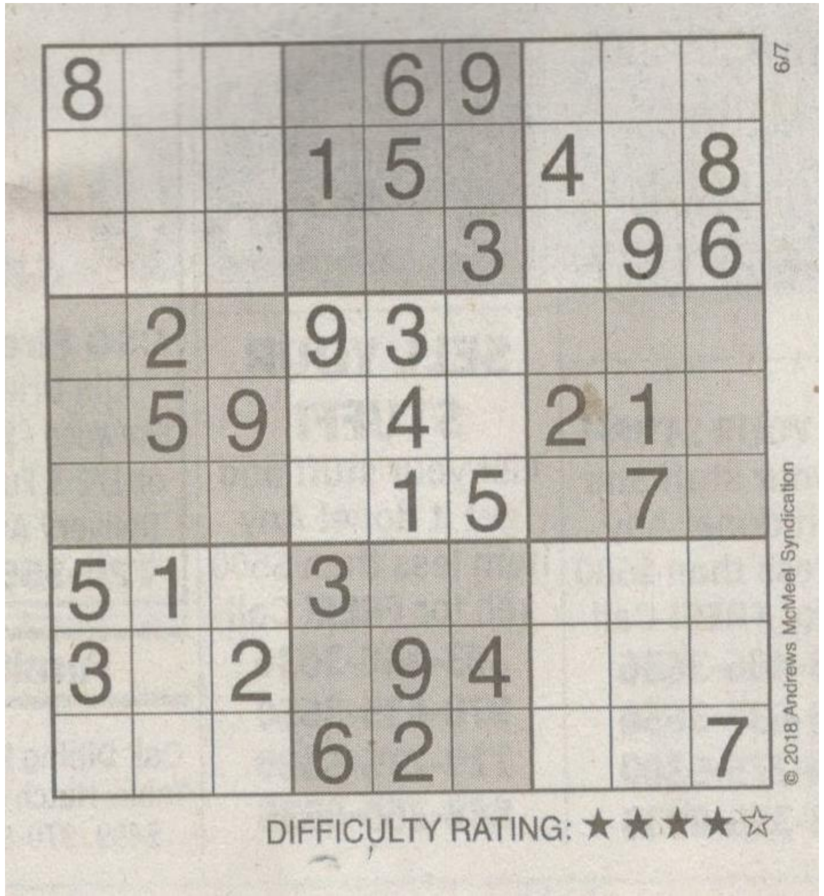
Sudoku Security: Secure Transmission Using Structured Deterrents

What is a Sudoku?

It is first and foremost a reverse compression mapping

The original Sudoku contains as little as 17 digits which provides an unambiguous forward mapping to 81 digits

Once the puzzle is completed, there are a virtually “infinite” number of possible back-mappings…

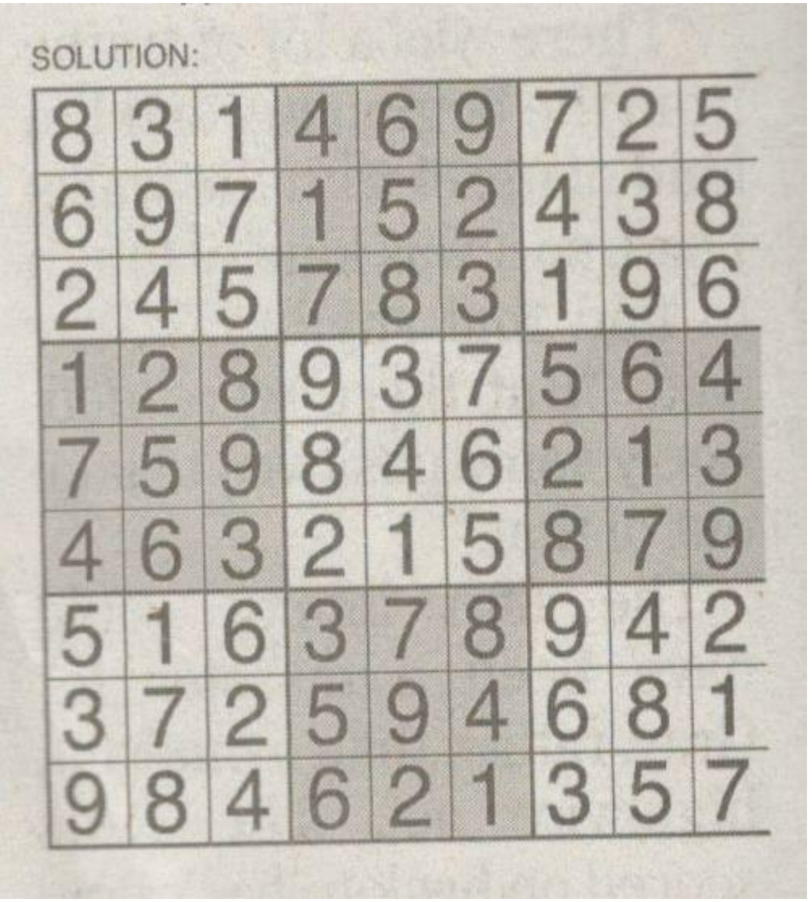


6/7

8				6	9			
			1	5		4		8
					3		9	6
	2		9	3				
	5	9		4		2	1	
				1	5		7	
5	1		3					
3		2		9	4			
			6	2				7

© 2018 Andrews McMeel Syndication

DIFFICULTY RATING: ★★★★☆



SOLUTION:

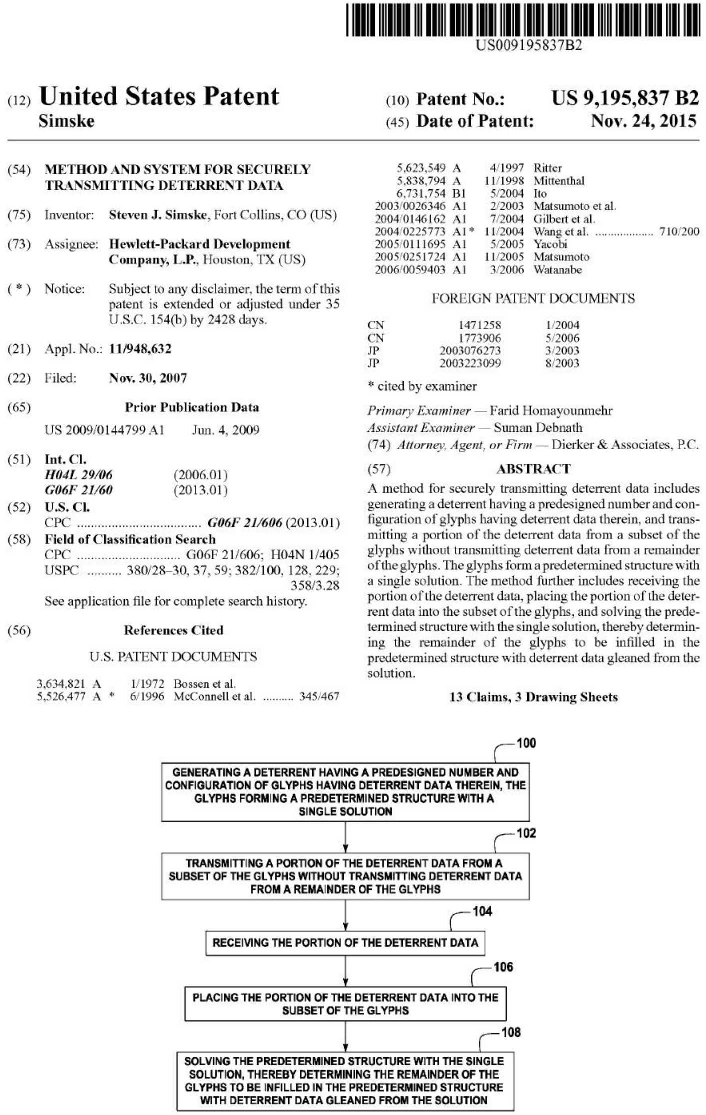
8	3	1	4	6	9	7	2	5
6	9	7	1	5	2	4	3	8
2	4	5	7	8	3	1	9	6
1	2	8	9	3	7	5	6	4
7	5	9	8	4	6	2	1	3
4	6	3	2	1	5	8	7	9
5	1	6	3	7	8	9	4	2
3	7	2	5	9	4	6	8	1
9	8	4	6	2	1	3	5	7

Secure Transmission Using Structured Deterrents

The Sudoku creates a means of forming a model for “Secure Transmission Using Structured Deterrents”, which means that the shared secret is, instead of telling the recipient how to decrypt the data, telling her how to organize the data upon receipt to generate dependent data

Sudoku Facts:

1. Total number of 81-cell Latin squares with {1,2,3,4,5,6,7,8,9} as the set: 9^{81}=1.966x10^{77}
2. Total number of 81-cell Latin squares with {1,2,3,4,5,6,7,8,9} as the set and the Sudoku requirements for 3x3 cells, rows and columns: 6.67x10^{21}
3. From this we see the huge reduction in search afforded by just a relatively simple structure
4. Overall, these types of Latin squares provide $\log_2 9$=3.17 bits/cell, and thus 81 cells provide 256.76 bits, or 32.1 bytes, of data
5. But, a Sudoku can take as little as 53.89 bits to fully prescribe (the sample shown on previous slide took 98.27, since it was not the hardest to solve), meaning 202.87 bits (25.4 bytes) are left over for a second channel of information



US009195837B2

(12) **United States Patent**
Simske

(10) **Patent No.: US 9,195,837 B2**
(45) **Date of Patent: Nov. 24, 2015**

(54) **METHOD AND SYSTEM FOR SECURELY TRANSMITTING DETERRENT DATA**

(75) Inventor: **Steven J. Simske**, Fort Collins, CO (US)

(73) Assignee: **Hewlett-Packard Development Company, L.P.**, Houston, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 2428 days.

(21) Appl. No.: **11/948,632**

(22) Filed: **Nov. 30, 2007**

(65) **Prior Publication Data**

US 2009/0144799 A1 Jun. 4, 2009

(51) **Int. Cl.**
H04L 29/06 (2006.01)
G06F 21/60 (2013.01)

(52) **U.S. Cl.**
CPC ***G06F 21/606*** (2013.01)

(58) **Field of Classification Search**
CPC G06F 21/606; H04N 1/405
USPC 380/28–30, 37, 59; 382/100, 128, 229; 358/3.28
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,634,821	A	1/1972	Bossen et al.	
5,526,477	A *	6/1996	McConnell et al.	345/467
5,623,549	A	4/1997	Ritter	
5,838,794	A	11/1998	Mittenthal	
6,731,754	B1	5/2004	Ito	
2003/0026346	A1	2/2003	Matsumoto et al.	
2004/0146162	A1	7/2004	Gilbert et al.	
2004/0225773	A1 *	11/2004	Wang et al.	710/200
2005/0111695	A1	5/2005	Yacobi	
2005/0251724	A1	11/2005	Matsumoto	
2006/0059403	A1	3/2006	Watanabe	

FOREIGN PATENT DOCUMENTS

CN	1471258	1/2004
CN	1773906	5/2006
JP	2003076273	3/2003
JP	2003223099	8/2003

* cited by examiner

Primary Examiner — Farid Homayounmehr
Assistant Examiner — Suman Debnath
(74) *Attorney, Agent, or Firm* — Dierker & Associates, P.C.

(57) **ABSTRACT**

A method for securely transmitting deterrent data includes generating a deterrent having a predesigned number and configuration of glyphs having deterrent data therein, and transmitting a portion of the deterrent data from a subset of the glyphs without transmitting deterrent data from a remainder of the glyphs. The glyphs form a predetermined structure with a single solution. The method further includes receiving the portion of the deterrent data, placing the portion of the deterrent data into the subset of the glyphs, and solving the predetermined structure with the single solution, thereby determining the remainder of the glyphs to be infilled in the predetermined structure with deterrent data gleaned from the solution.

13 Claims, 3 Drawing Sheets

100 GENERATING A DETERRENT HAVING A PREDESIGNED NUMBER AND CONFIGURATION OF GLYPHS HAVING DETERRENT DATA THEREIN, THE GLYPHS FORMING A PREDETERMINED STRUCTURE WITH A SINGLE SOLUTION

102 TRANSMITTING A PORTION OF THE DETERRENT DATA FROM A SUBSET OF THE GLYPHS WITHOUT TRANSMITTING DETERRENT DATA FROM A REMAINDER OF THE GLYPHS

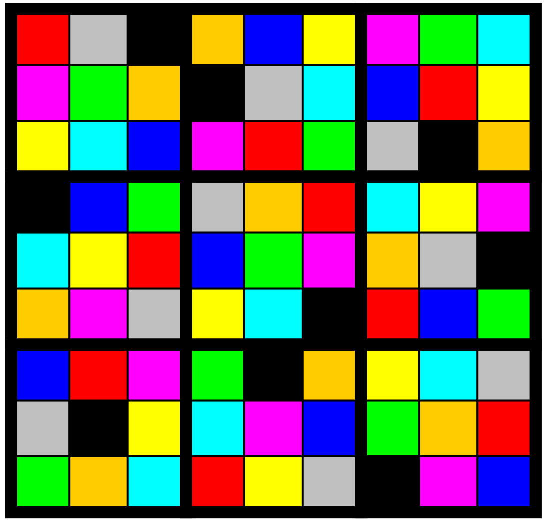
104 RECEIVING THE PORTION OF THE DETERRENT DATA

106 PLACING THE PORTION OF THE DETERRENT DATA INTO THE SUBSET OF THE GLYPHS

108 SOLVING THE PREDETERMINED STRUCTURE WITH THE SINGLE SOLUTION, THEREBY DETERMINING THE REMAINDER OF THE GLYPHS TO BE INFILLED IN THE PREDETERMINED STRUCTURE WITH DETERRENT DATA GLEANED FROM THE SOLUTION

Secure Transmission Using Structured Deterrents

- Sudoku (literally, “Su doku”, or “number place”) is a puzzle typically 9x9 tiles in dimension, in which each of the rows and columns, along with each 3x3 cell, contains the numerals {1,2,3,4,5,6,7,8,9}. This is a specialized form of a Latin square, and there is no general solution to the number of permutations
- However, using a combination of theory and simulations, the number of ways of filling in a blank Sudoku grid was shown in May 2005 to be 6,670,903,752,021,072,936,960 (~6.67×10^{21}). This gives up to 72 bits of information, provided the 6.67×10^{21} permutations can be represented sequentially (in practice, since there is no closed form, considerably less bits will be represented, although the reference http://www.afjarvis.staff.shef.ac.uk/Sudoku/Sudoku.pdf demonstrates 362880 * 2612736 * 2612726 = 2.477×10^{18} permutations, or 61 bits, that are readily represented sequentially just by using the uppermost and leftmost 3x3 cells, or 5 cells, total)
- When multiplied by the number of bits encoded by 9 different choices for each tile (log(9)/log(2)), this results in 229 bits in a specific Sudoku, and a somewhat lower 193 bits in one of the 5-cell specified Sudokus. That is, 2^{193} unique sequences (just over 2^3 bits per tile x just over 2^{60} permutations that can be readily encoded into a Sudoku without specifying the four 3x3 cells in the lower left). This demonstrates that a Sudoku contains a large amount of information (as much as two 96-bit RFID chips).



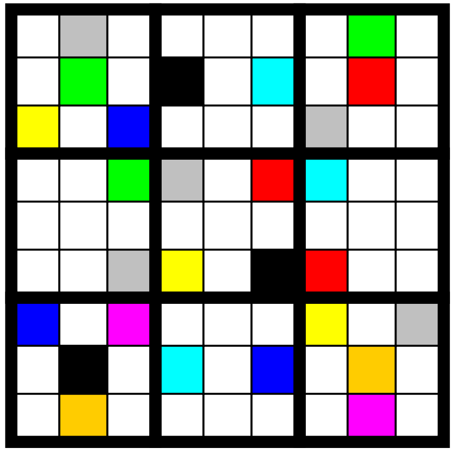
A Sudoku using {RGBCMYKEO} or red, green, blue, cyan, magenta, yellow, black, grey and orange colored tiles is shown here:

Secure Transmission Using Structured Deterrents

A Sudoku is a built-in error check, since each row, column and 3x3 cell has a built-in checkbit (by the rules of the Sudoku, all 9 colors must appear in each of these 27 subregions). Effectively, 1/3 of the Sudoku tiles are checkbits seen from this perspective.

Thus, if a Sudoku-based color tile deterrent is specified, the error check on the authentication is instantaneous. If any row, column or 3x3 cell does not represent all of the colors, then there is an authentication error.

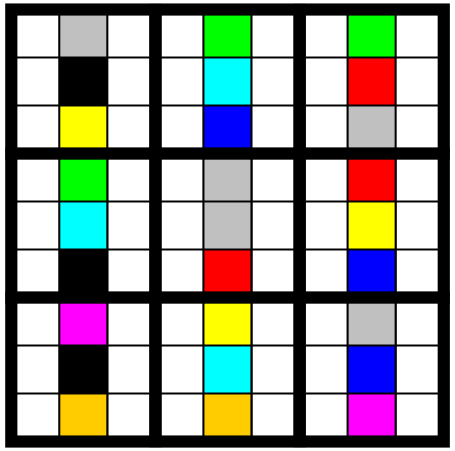
We go one step further and use the solution to the Sudoku as a means of transmitting the information to encode in the deterrent. This allows us to send the deterrent specification over an open line between two trusted parties. One, the deterrent provider, generates the Sudoku deterrents. Next, the deterrent provider sends a subset of the Sudoku grid (such as the 27 colored tiles shown in the unsolved Sudoku to the right)



These 27 colored tiles can be exactly solved at the receiving end by a Sudoku completion algorithm (Sudoku completion is a relatively straightforward machine task), and the overall Sudoku deterrent generated. The shared secret is simply the locations of the tiles that will be filled in by the Sudoku sequence. In the unsolved Sudoku above (which exactly specifies the fully solved Sudoku described previously), these locations are, in reading order, locations 2, 8, 11, 13, 15, 17, …, 80. A "person in the middle" reading the corresponding message would only see the color information—E, G, G, K, C, R, …, M—and without the location information for these 27 tiles would be unable to easily compute the Sudoku.

Secure Transmission Using Structured Deterrents

For example, equally spacing these colors would result in a non-legitimate (unsolvable) Sudoku as shown here:



In practice, sending roughly half of the 81 tiles (as a sequence of colors) provides a robust solution—the Sudoku is overspecified, and so speedily filled in by the Sudoku completing algorithm, and the overspecified "extra" tiles make it difficult for the counterfeiter to guess the correct locations.

<u>Note on implementation:</u>

Note that Sudokus of other sizes (e.g. 16x16, 25x25) are possible, and of course a deterrent may be comprised of NxM Sudokus where N and M are (not necessarily equal) positive integers to provide any desired number of bits or match a desired size. For example, there are many Sudoku variations, such as 2x2, 3x2 and 2x3. Related to Sudoku, magic squares and Latin squares can provide the same "structured" set of tiles. Customized checkbits can be used to map variants to the same 9x9 tile structure.

Due to the imposed structure of a Sudoku/Latin square/magic square, a non-full set of bits may be sent and the missing elements reconstructed on that end by placing the sent data in the proper rows and columns and computing the remaining data from the structure. A transmission snoop cannot infer the missing information if he does not know how the data maps into the structure.

Secure Transmission Using Structured Deterrents

Advantages

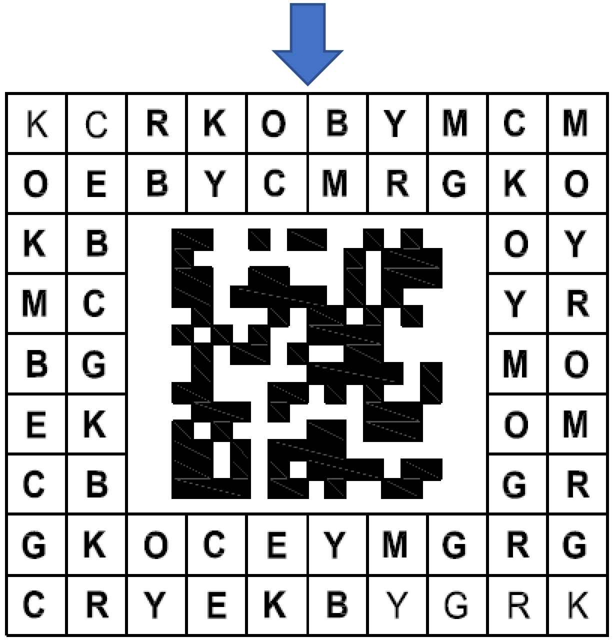
→The Sudoku approach provides additional error detection (by row, by column, and by cluster simultaneously) and encryption (by sending a partially filled deterrent and relying on the end device to compute the overall deterrent) advantages. Error code checking is innately performed in the encoding (as it turns out, the Sudoku approach corresponds to a roughly 4:1 redundancy.

→The Sudoku approach allows spot inspection (since only ~25% of the tiles are independent).

→Verification can be on a different data set than the data sent…even 100% different, making data translation between the two difficult. This means that, for example, 40% of the tiles are sent to the end user, and a completely different 40% of the tiles are "read" during inspection/authentication. Both sets completely specify the actual Sudoku layout of tiles, but are not correlated with each other (making packet snooping and other forms of transmission monitoring less useful to the would-be counterfeiter). This is a form of *a posteriori* secret sharing verification.

Implementation of the Public Key function of the Structured Deterrent:

R		K	O	B	Y	M		C
M		O		E		B		Y
	C		M	R	G		K	O
K	B			O			Y	M
C	Y	R	B	G	M	O	E	K
O	M			C			B	G
	R		G	K	O		C	
E		Y		M		G		R
G		C	R	Y	E	K		B



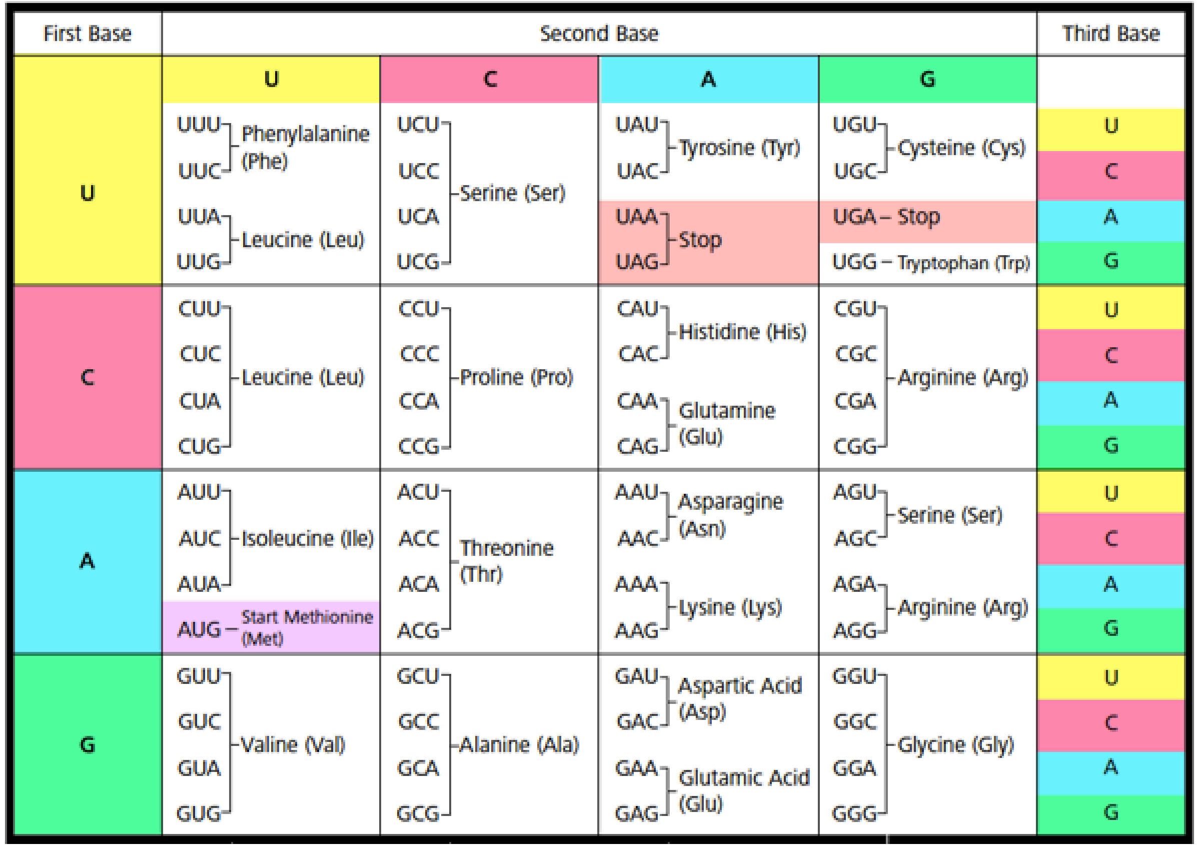
K	C	R	K	O	B	Y	M	C	M
O	E	B	Y	C	M	R	G	K	O
K	B							O	Y
M	C							Y	R
B	G							M	O
E	K							O	M
C	B							G	R
G	K	O	C	E	Y	M	G	R	G
C	R	Y	E	K	B	Y	G	R	K

TRANSITION

Genetic Approaches to System Security

Thinking about Genomics through a Machine Learning Lens: Basics of how several DNA- and RNA-based problems translate into commonly-studied areas of machine (natural language processing, classification, and regression).

mRNA Codon/Amino Acid Chart



First Base	Second Base				Third Base
	U	C	A	G	
U	UUU Phenylalanine (Phe)	UCU Serine (Ser)	UAU Tyrosine (Tyr)	UGU Cysteine (Cys)	U
	UUC	UCC	UAC	UGC	C
	UUA Leucine (Leu)	UCA	UAA Stop	UGA – Stop	A
	UUG	UCG	UAG	UGG – Tryptophan (Trp)	G
C	CUU Leucine (Leu)	CCU Proline (Pro)	CAU Histidine (His)	CGU Arginine (Arg)	U
	CUC	CCC	CAC	CGC	C
	CUA	CCA	CAA Glutamine (Glu)	CGA	A
	CUG	CCG	CAG	CGG	G
A	AUU Isoleucine (Ile)	ACU Threonine (Thr)	AAU Asparagine (Asn)	AGU Serine (Ser)	U
	AUC	ACC	AAC	AGC	C
	AUA	ACA	AAA Lysine (Lys)	AGA Arginine (Arg)	A
	AUG – Start Methionine (Met)	ACG	AAG	AGG	G
G	GUU Valine (Val)	GCU Alanine (Ala)	GAU Aspartic Acid (Asp)	GGU Glycine (Gly)	U
	GUC	GCC	GAC	GGC	C
	GUA	GCA	GAA Glutamic Acid (Glu)	GGA	A
	GUG	GCG	GAG	GGG	G

https://students.ga.desire2learn.com/d2l/lor/viewer/viewFile.d2lfile/1798/12708/dna-rna13.html

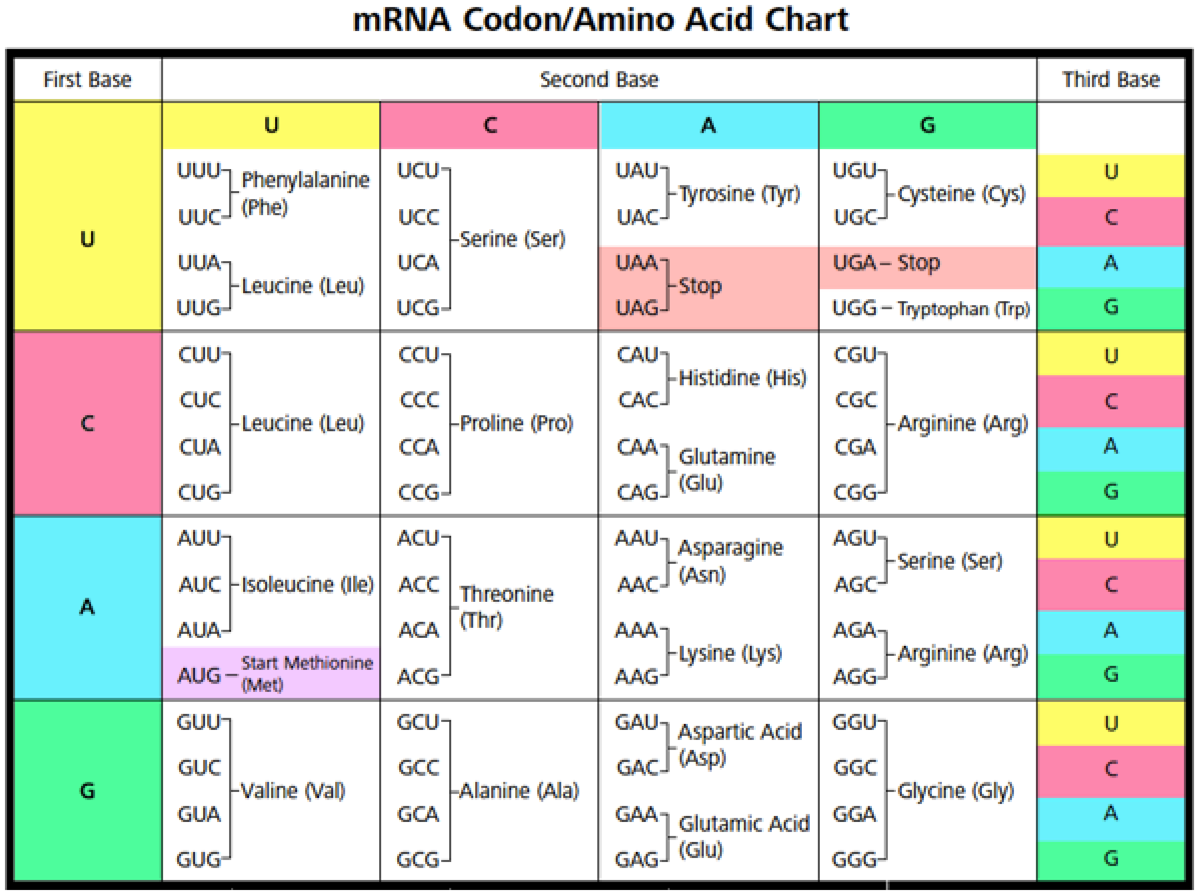
Translation is the last step from DNA to protein: the synthesis of proteins directed by an **mRNA** template. The information contained in the nucleotide sequence of the **mRNA** is read as three letter words (triplets), called codons.

Translation provides a one-way (trapdoor) function: A *trapdoor function* is a function that is uncomplicated to perform in one direction, either requires or highly benefits from a secret to perform the inverse calculation at all, or at least efficiently

Methionine and Tryptophan are singly-encoded; the other 18 amino acids are multi-encoded (up to 6 as for leucine, serine, and arginine).

Genetic Approaches to System Security

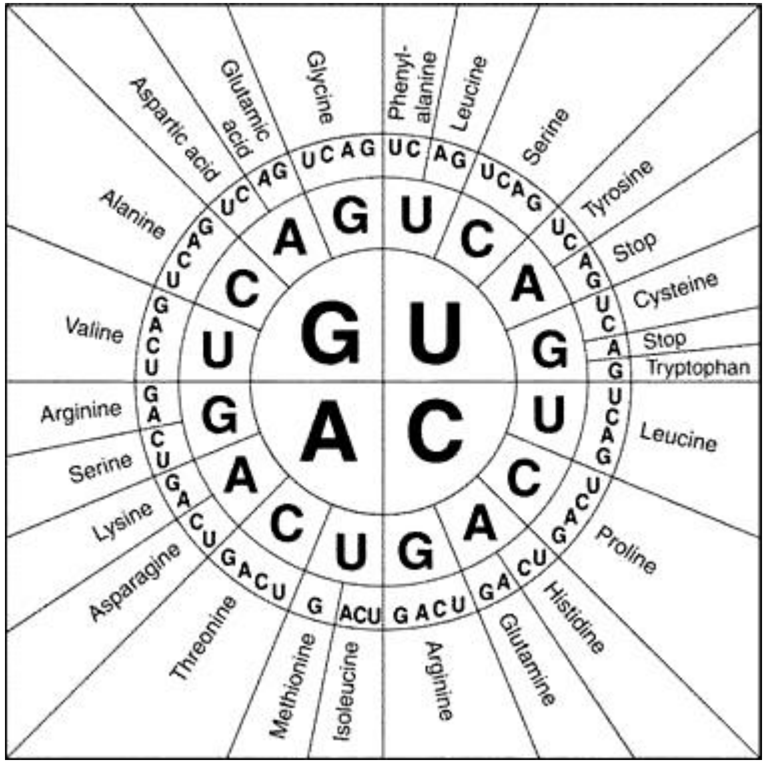
Sometimes a different look at the mapping provides better insight into the relative “stochasticity” of the mapping



mRNA Codon/Amino Acid Chart

First Base	Second Base				Third Base
	U	C	A	G	
U	UUU Phenylalanine (Phe)	UCU Serine (Ser)	UAU Tyrosine (Tyr)	UGU Cysteine (Cys)	U
	UUC	UCC	UAC	UGC	C
	UUA Leucine (Leu)	UCA	UAA Stop	UGA – Stop	A
	UUG	UCG	UAG	UGG – Tryptophan (Trp)	G
C	CUU Leucine (Leu)	CCU Proline (Pro)	CAU Histidine (His)	CGU Arginine (Arg)	U
	CUC	CCC	CAC	CGC	C
	CUA	CCA	CAA Glutamine (Glu)	CGA	A
	CUG	CCG	CAG	CGG	G
A	AUU Isoleucine (Ile)	ACU Threonine (Thr)	AAU Asparagine (Asn)	AGU Serine (Ser)	U
	AUC	ACC	AAC	AGC	C
	AUA	ACA	AAA Lysine (Lys)	AGA Arginine (Arg)	A
	AUG – Start Methionine (Met)	ACG	AAG	AGG	G
G	GUU Valine (Val)	GCU Alanine (Ala)	GAU Aspartic Acid (Asp)	GGU Glycine (Gly)	U
	GUC	GCC	GAC	GGC	C
	GUA	GCA	GAA Glutamic Acid (Glu)	GGA	A
	GUG	GCG	GAG	GGG	G

https://students.ga.desire2learn.com/d2l/lor/viewer/viewFile.d2lfile/1798/12708/dna-rna13.html



https://rbssbiology11ilos.wikispaces.com/Codon+Wheel

Genetic Approaches to System Security

What about the data itself?

Mapping	# Amino Acids so Mapped
1	2 (Met, Trp)
2	9 (Phe, Tyr, His, Glu, Asn, Lys, Asp, Glu, Cys)
3	1 (Ile)
4	5 (Val, Pro, Thr, Ala, Gly)
5	0
6	3 (Leu, Ser, Arg)

Glu=Glutine and Glutamic Acid

As we will see in the next slide, the 1.585 extra bits possible for this distribution is close to the theoretical maximum, which is 1.609 bits.

i	p(i)	log$_2$(p(i))	-p(i)*log$_2$(p(i))
1	0.10	-3.322	0.332
2	0.45	-1.152	0.518
3	0.05	-4.322	0.216
4	0.25	-2.000	0.500
5	0.00	Undefined	0.00
6	0.15	-2.737	0.411

The entropy of the above table is given by:

$$e = -\sum_{i=1}^{6} p(i) * log_2(p(i))$$

Its value is 1.977, and its minimum and maximum values are 1.000 and 2.585, respectively. This means instead of the codon mapping carrying as much as 1.585 “extra” bits of information, it carries only 0.977 “extra” bits

Genetic Approaches to System Security

There is another "information gain" associated with the codon mapping

There are 64 codons, which is 6 bits exactly, and it is translated into 21 outputs (20 amino acids and STOP), which is 4.392 bits (since $2^{4.392}$ = 21). That means that there are 6.000-4.392 = 1.608 extra bits to obfuscate the trapdoor (one way) nature of the translation.

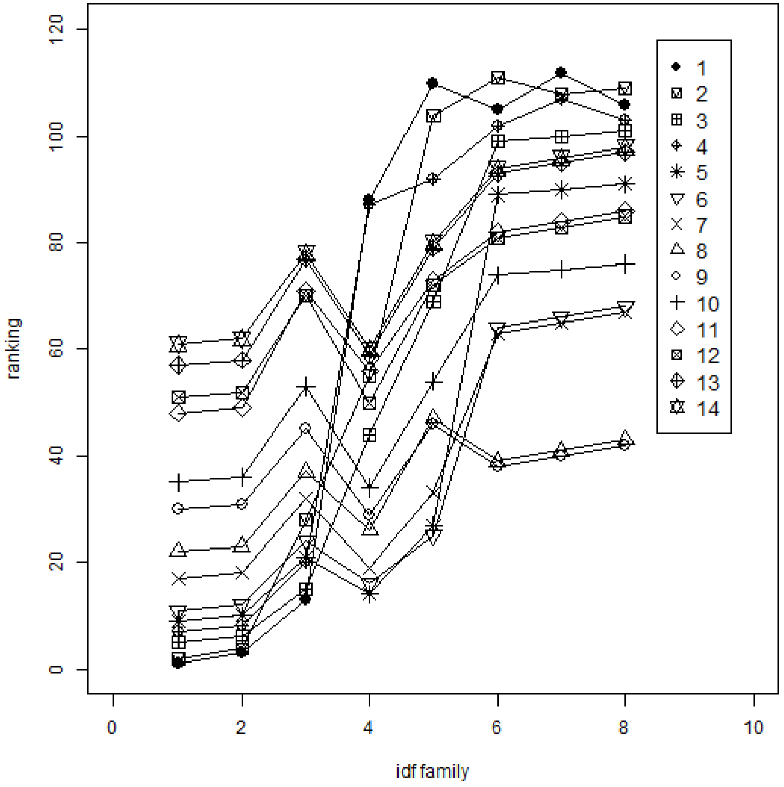
Alternatively, we can consider 61 codons, which is 5.931 bits (Since $2^{5.931}$ = 61), which are translated into 20 amino acids, which is 4.322 bits (since $2^{4.322}$ = 20). This means that there are 5.931-4.322 = 1.609 extra bits to obfuscate the trapdoor nature of the translation.

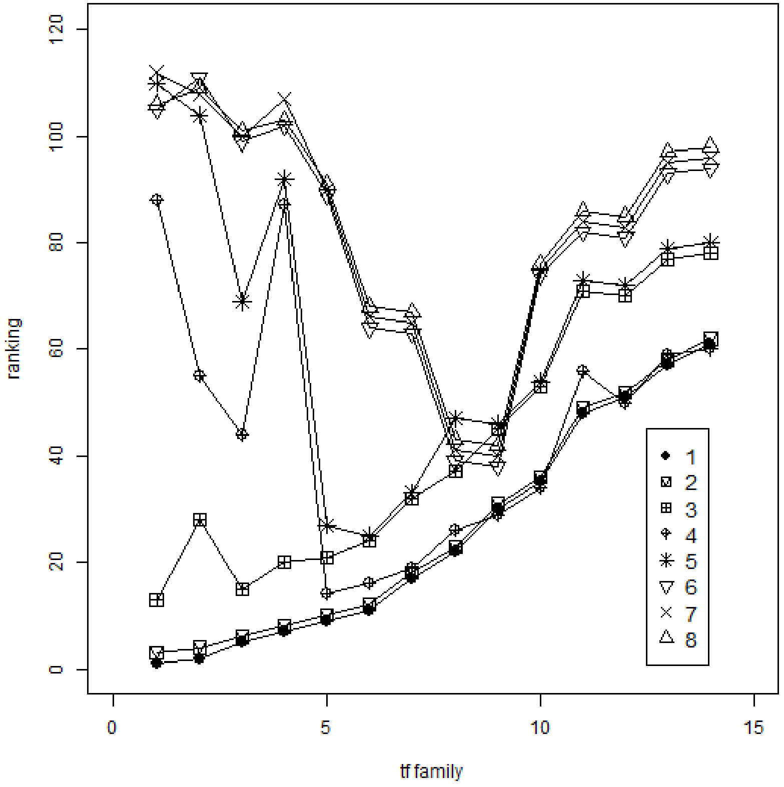
The use of information theory shows us that nature selected an intermediate amount of obfuscation bits (0.977) in the range [0, 1.609]. Not surprisingly, as this is generally consistent with a system that has been optimized through natural selection.

We now turn to some specific aspects of machine learning and genomics

Genetic Approaches to System Security: Natural Language Processing

Thinking about Genomics through a Machine Learning Lens: Basics of how several DNA- and RNA-based problems translate into commonly-studied areas of machine (natural language processing, classification, and regression).



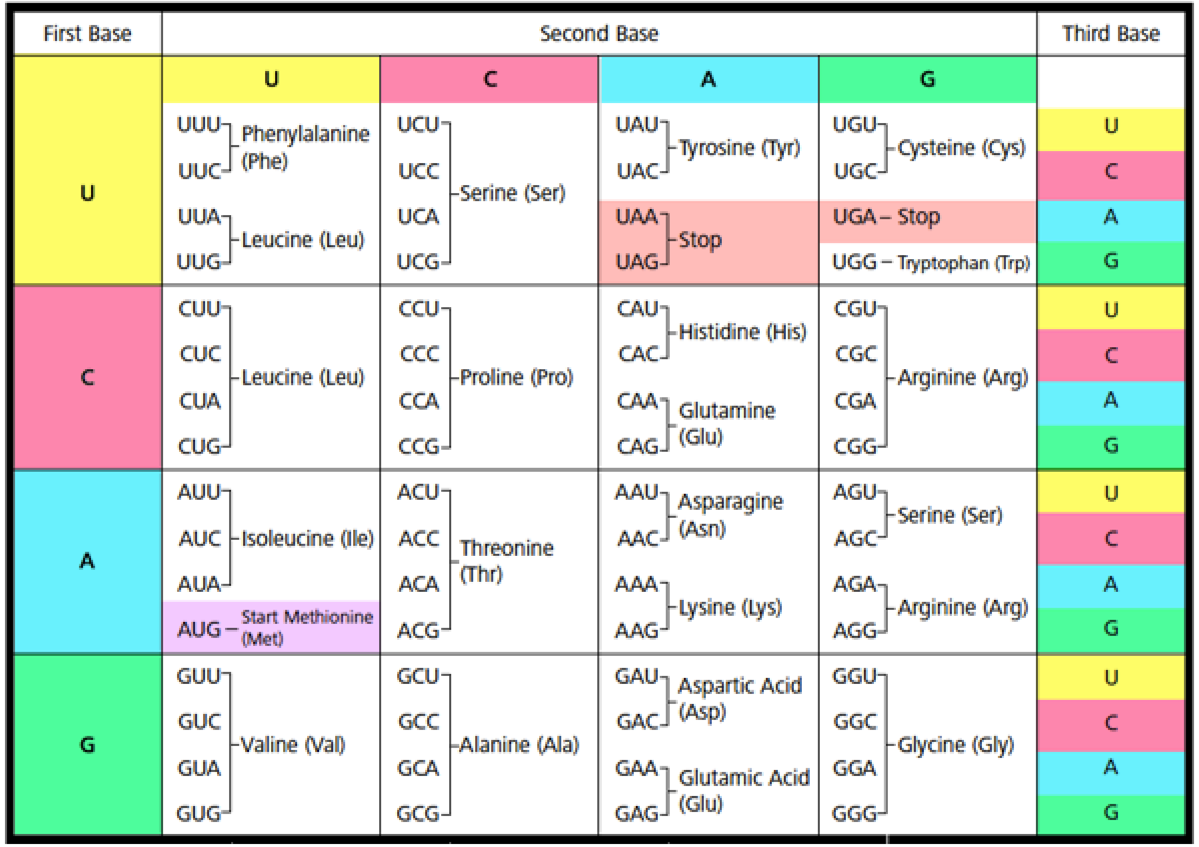


In terms of TF*IDF, where TF=Term Frequency and IDF=Inverse of Document Frequency, we can choose the ambiguous terms in the encoding and change them to obtain desired behavior…

Genetic Approaches to System Security: Natural Language Processing

Thinking about Genomics through a Machine Learning Lens: Basics of how several DNA- and RNA-based problems translate into commonly-studied areas of machine (natural language processing, classification, and regression).

mRNA Codon/Amino Acid Chart



First Base	Second Base				Third Base
	U	C	A	G	
U	UUU Phenylalanine (Phe)	UCU Serine (Ser)	UAU Tyrosine (Tyr)	UGU Cysteine (Cys)	U
	UUC	UCC	UAC	UGC	C
	UUA Leucine (Leu)	UCA	UAA Stop	UGA – Stop	A
	UUG	UCG	UAG	UGG – Tryptophan (Trp)	G
C	CUU Leucine (Leu)	CCU Proline (Pro)	CAU Histidine (His)	CGU Arginine (Arg)	U
	CUC	CCC	CAC	CGC	C
	CUA	CCA	CAA Glutamine (Glu)	CGA	A
	CUG	CCG	CAG	CGG	G
A	AUU Isoleucine (Ile)	ACU Threonine (Thr)	AAU Asparagine (Asn)	AGU Serine (Ser)	U
	AUC	ACC	AAC	AGC	C
	AUA	ACA	AAA Lysine (Lys)	AGA Arginine (Arg)	A
	AUG – Start Methionine (Met)	ACG	AAG	AGG	G
G	GUU Valine (Val)	GCU Alanine (Ala)	GAU Aspartic Acid (Asp)	GGU Glycine (Gly)	U
	GUC	GCC	GAC	GGC	C
	GUA	GCA	GAA Glutamic Acid (Glu)	GGA	A
	GUG	GCG	GAG	GGG	G

Suppose we wish to encode the peptide:
Leu-Pro-His-Gly
Then we have 6x4x2x4 = 192 different codons

{UUA,UUG,CUU,CUC,CUA,CUG}=Leu
{CCU,CCC,CCA,CCG}=Pro
{CAU,CAC}=His
{GGU,GGC,GGA,GGG}=Gly

We can choose different strategies:
UUACCGCAUGGA = high entropy (3 each, no runs of 3)
CUCCCCCACGGC = “C” bias (high compression)
CUGCCGCACGGC = “CG” bias (low entropy)

Genetic Approaches to System Security: Classification

Thinking about Genomics through a Machine Learning Lens: Basics of how several DNA- and RNA-based problems translate into commonly-studied areas of machine (natural language processing, classification, and regression).

Classification:

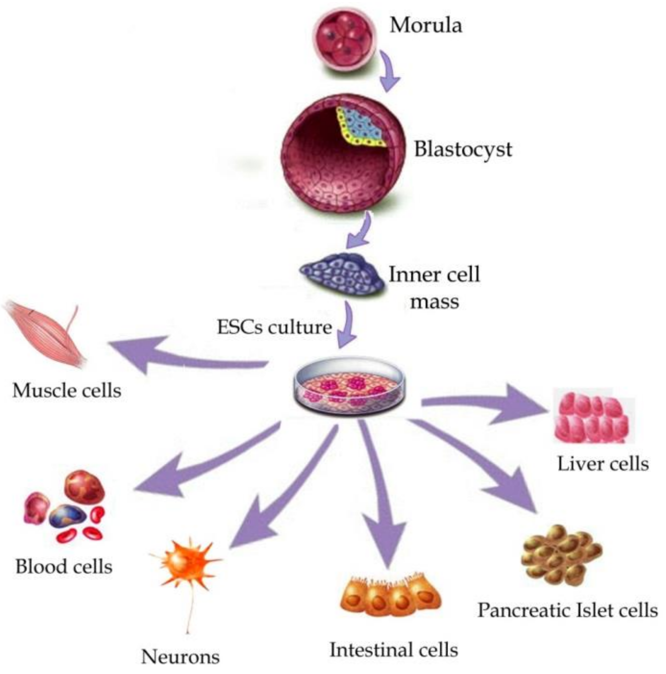
Classify different metagenome skimming approaches (high throughput sequencing, environmental genomics, ecogenomics or community genomics) based on the distributions of base pairs (bps)…note that different approaches can be used: SOLiD at ~50bp, Ion Torrent/pyrosequencing at ~400bp, and Illumina MiSeq at ~500bp, which provide the fodder for meta-analytics applied to metagenomics!

TF*IDF of given sequences, overall, and within the different metagenomic approaches, can identify different cellular behavior, including quiescence vs. proliferation, differentiation, activation, stage in cell cycle, etc.

Regression:

From genomics to proteomics and metabolomics

Can we predict the functional behavior of the DNA within organism(s)—levels of expression, proliferation, activation, synthesis, etc.



http://thebeautybrains.com/2014/07/do-stem-cells-work-in-cosmetics/

Genetic Approaches to System Security: Classification

Thinking about Genomics through a Machine Learning Lens: Basics of how several DNA- and RNA-based problems translate into commonly-studied areas of machine (natural language processing, classification, and regression).

New use of genetically engineered peptides as storage and as a means of **multi-channel security**

Multi-channel information:

1. Statistics (percentage, sequence lengths, distribution) of each nucleotide
2. Statistics (percentage, distribution) of each amino acid
3. Statistics (percentage, distribution) of each peptide of interest
4. Statistics (percentage, distribution) of each protein of interest

Shared secret/public key is the amino acid sequence
Private key is the actual sequence of codons (disambiguated)
Odds of guessing the codon sequence for a 20-residue peptide with each amino acid in it is:

$$\prod_{i=1}^{2} 1 \prod_{j=1}^{9} \frac{1}{2} \prod_{k=1}^{1} \frac{1}{3} \prod_{l=1}^{5} \frac{1}{4} \prod_{m=1}^{3} \frac{1}{6} = \frac{1}{339{,}738{,}624}$$

Could it be the next blockchain? E.g., find the codon sequence with the right leading number of A, C, G, U? I hope not!!!

DNA as a storage medium…

- From Jurassic Park to Jurisdiction Park
- Use of Introns for signature

x	A	C	G	T
A	C	G	T	A
C	G	A	C	T
G	T	C	A	G
T	A	T	G	C

e.g.

ACG TTA AGC (Bob)
X
TGC GAA TCC (Alice)
=
ACC GAC ACA (Codons for the peptide)

Thanks!