PART ONE: Instructions for Confirming Necessity of Secure Line and Username

All intentions to begin secure line must be initiated by one member with the message in the exact syntactic structure as the following:

CODE DOUBLE DIAMOND VANTABLACK

This message should be sent via SMS/iMessage messaging. Immediately afterwards, the original sender must follow this code with a secure key to ensure that the intent is accurate to the extent of vital importance. Instructions on generating this key are as follows:

There are two people necessary for this to begin. The first person to begin (and the one who sent the initial signal "CODE DOUBLE DIAMOND VANTABLACK") is denoted the *sender* and the person who is sent this message is denoted the *receiver*.

To begin, the sender must determine the first key, or "proto-key." This proto-key will be determined by concatenating two pieces of data (or keys) in a certain order: the specific **date and time** at which "CODE…" was sent and the most recent text message sent by the **sender** (this must be **UTC**).

The date and time will be written in numerical form according to YYYYMMDDHHNN; where YYYY signifies the year, MM signifies the month, DD signifies the day, HH signifies the hour (military time), and NN signifies the minute. For example, if "CODE…" was sent on the 12th of January, 1992 at 8:03PM, the first section of this proto-key will read "199201122003." Divide the six-digit number comprised of the first six digits by the six-digit number comprised of the second six digits. The first twelve fractional non-significant-zero numbers is called the **DATE** key.[1]

Next to determine the second section of the proto-key, the *sender* will determine the most recent message sent by either of the sender or reciever. This message must be purely text, so if the last message is a link, image, video, or any other form that is not sent with pure textual information, use the message preceding this. With this, determine the first ten alphabetic characters in the message (this excludes special, spacing, numeric, non-Latin script letters, etc.) All letters will then be converted to uppercase. Should the message be sent with less than ten letters, the key reads as left justified with padded X's to the right (e.g., "Testing" becomes "TESTINGXXX"). Encrypt this text string via Base64 and remove any element that is not an alphabetic (letter) character. This is called the **TEXT** key.

From this, the two parts of the proto-key will be formed according the following rules on page 2). The vertical column of instructions pertain to the text key while the instructions listed along the horizontal row pertain to the date key. Use the symbols and notation on page 3) to determine the specific operation to be done asked by the chart. Following completion of the page 2) tables, you will form the key to be later encrypted. Follow the directions below the chart on page 3) to determine what to do with this key.

Further assistance with instructions can be found in footnotes and useful sources/information can be found in Appendix A.

[1] "Fractional non-significant-zero" numbers means that if division yields 23.0000100381123432, the date key would be then be "100381123432"

PART TWO: Instructions for Generating Username

Each command is listed with two lines, concatenate the result of the first line in front of the result of the second line.

If the date key ends in an odd number, observe the following:

	If more even numbers appear than odd[2]	If the date key is a prime number	If the average value of the sum of the elements of the date key is greater than 5.5	If more than one condition is met	If none of the conditions are met
If text key has at least three occurrences any of: B,F,J,S,V	**(TEXT)$^{-1}$||(TEXT)** **(DATE)**	**DATE + 3** **TEXT + (-2)**	**TEXT – 3** **DATE**	**TEXT/VOW.** **DATE**	**((TEXT)||(DATE))^** **TEXT**
If the text key contains an even number of vowels	**TEXT** **DATE/(1,3,4, and 5)**	**(TEXT)$^{-1}$** **DATE**	**TEXT** **DATE^{-1}||(DATE/(EVEN))**	**DATE^** **TEXT^**	**DATE/(7,8,9, and 0)** **TEXT/(VOW, J, and L)**
Using A=1, ... Z=26, if the sum of first two letters of the text key is less than or equal to 30	**DATE^{-1}** **(TEXT + 4) – (-2)**	**DATE/ODD** **TEXT + 1**	**TEXT + 4** **DATE||TEXT**	**DATE – (-5)** **TEXT + 2**	**DATE – (-3)** **TEXT + (-3)**
If exactly two of the above conditions are met	**DATE + 1** **TEXT**	**((TEXT)$^{-1}$ – (-3))** **DATE + 2**	**TEXT** **DATE/(PRIME)**	**TEXT/VOW.** **DATE**	**(DATE)^** **((TEXT)/(VOW.))^**
If all of the conditions are met	**TEXT/(R,S,T, and E)** **DATE**	**(DATE)^||(DATE)** **TEXT**	**DATE** **TEXT**	**(TEXT/VOW)||TEXT** **DATE^{-1}**	**DATE – 3** **(TEXT + 3)$^{-1}$**
If none of the conditions are met	**DATE – (4)** **TEXT^{-1}**	**(DATE)||(TEXT)$^{-1}$** **TEXT^**	**TEXT** **DATE||DATE**	**DATE/(0,1,2,3, and 4)** **TEXT + 4**	**TEXT/((DATE)^)** **DATE + (-4)** [3]

If the date key ends in an even number, observe the following:

	If any three successive numbers of the date key form a 3-digit prime	If at least one date key digit is repeated at least three times	If at least four successive numbers in the date key are ordered ascendingly	If more than one condition is met	If none of the conditions are met
If the third, fourth, and fifth letters are consonants	**(TEXT)$^{-1}$** **DATE**	**((TEXT)$^{-1}$ – (-3))** **DATE + 2**	**TEXT/VOW.** **DATE**	**(TEXT/VOW)||TEXT** **DATE^{-1}**	**DATE/(0,1,2,3, and 4)** **TEXT + 4**
If there are less than or equal to four letters in the text key from the first half of the alphabet	**DATE + (-3)** **TEXT + (-3)**	**DATE/EVEN** **TEXT + 3**	**TEXT** **DATE/PRIME**	**TEXT – 3** **DATE^{-1}**	**TEXT^{-1}||TEXT** **DATE||DATE^{-1}**
If less than half of the letters from the text key have at least one line of symmetry	**DATE^** **TEXT^**	**TEXT** **DATE/PRIME**	**TEXT** **DATE^{-1}||(DATE/(ODD))**	**DATE – 3** **(TEXT + 3)$^{-1}$**	**(TEXT)$^{-1}$||(TEXT)** **(DATE)**
If exactly two of the above conditions are met	**TEXT/(M,N,O, and P)** **DATE**	**(TEXT)$^{-1}$||(TEXT)** **(DATE)**	**DATE/(0,1,6,9, and 4)** **TEXT + 4**	**DATE/(0,1,2, and 8)** **TEXT + 4**	**DATE – (2)** **TEXT + (6)**
If all of the conditions are met	**TEXT** **DATE/(NON-PRIME)**	**DATE + 1** **TEXT + (-2)**	**DATE – (3)** **TEXT + (-1)**	**TEXT/((DATE)^)** **DATE + (-4)**	**((TEXT)^)$^{-1}$** **((DATE)$^{-1}$)^**
If none of the conditions are met	**TEXT/(A,)** **DATE/(EVEN)**	**DATE** **TEXT**	**DATE/(SQUARE NUMS)** **TEXT + (-2)**	**DATE – (-1)** **(TEXT – (1))$^{-1}$**	**TEXT + 4** **DATE^||TEXT^{-1}**

Notation Information

[2] Assume 0 is even, "VOW" signifies vowels

[3] For first line, remove letters in TEXT that are the same as letters in the value inverted DATE

The appearance of the either the proto-key "DATE" or "TEXT" denotes the respective key as determined on page 1). Here, however, to demonstrate notation, only the word "TEST" will be used but note that the two proto-keys can be inserted in replacement. Here TEST represents the key "CALCULATOR"

(TEST)	*Parentheses*: e.g., CALCULATOR
TEST‖TEST	*Concatenation*: e.g., CALCULATORCALCULATOR
$(TEST)^{-1}$	*Inversion of Order*: e.g., ROTALUCLAC
(TEST)^	*Inversion of Value*: Each element will be switched from alphabetic to numeric and vice versa with respect to the following. Letters will be converted via A=1,…, Z=26 and numbers will be converted vice versa. e.g., (TEST)^ → 3112321121201518. See below for further explanation[4]
TEST + (x)	*Digit Shift*: each element of key is shifted to a value x away, likewise to ASCII arithmetic. If the alphabetic digit exceeds "Z," the next digit will be "A." Similarly, if the numeric digit exceeds "9" the next digit will be 0.[5]
TEST – (x)	*Deletion*: x amount of digits will be removed from a side of the key. A positive value denotes the right side (last x numbers) and a negative value denotes the left side (first x numbers); e.g., TEST – (-4) → ULATOR
TEST/(COMMAND)	*Excise due to Command*: remove all elements of the key that match or are in common with the "command" word; e.g., TEST/(VOWELS) → CLCLTR

With the new generated key, use A=1, … Z=26 and the methodology in Footnote 3, to convert all numbers into letters. From this seperate the first half of the letters from the second half. The *first* half is the **KEYWORD**. The *second* half is the **PLAINTEXT**. If n represents the number of elements and is odd, then the $\lceil n/2 \rceil^{th}$ element will be last in the keyword. Using a *Playfair Cipher*, encrypt the plaintext using the keyword (assume to omit the letter "Q"). This is the **CIPHERTEXT.** Finally, using Base64 encode the ciphertext with UTF-8 output charset (removing spaces and changing each letter to uppercase). This is the **USERNAME**.

Once the username has been determined by the sender, only the *keyword* will be sent by the sender. To confirm that this is accurate, the receiver will use the same method to determine the key. If the receiver determines the key to be the same as what was sent by the sender, he will reply "PROCEED" as a sign that he has received the accurate message. Ensure the security of the username by not sending it – the "sender" should only divulge the keyword as means to check with the "receiver".

If the key received by the sender differs from the key the receiver deciphers, the receiver will reply "FAIL" and presume the initial message to be from an unknown source. Should the sender or receiver deliver an incorrect key, the above process will be restarted. Now, use the exact time the word "FAIL" is sent as the **DATE** key and the next most recent message (previous) sent by the sender as the **TEXT** key.

PART THREE: Generating a Pre-Number Key for Password

To begin, determine the two letter string texts – the *plaintext* and the *ciphertext*. Convert each element of both texts from letters into numbers using A=1,...,Z=26. Henceforth, the terms "plaintext" and

[4] Since there are two-digit equivalent letters, to convert numbers to letters using A=1, … Z=26, always chose the largest possible letter that can be made out of the given number string. For example, the number "42138910" will begin with the letter "D" for 4 since 42 is not a viable number. However, since 21 is a viable number, the next letter will not be "B" for 2 but "U" for 21. Since the next number is 3 and 38 is not viable, the next letter is "C". By this rule, the original number can be rewritten for assistance as 4-21-3-8-9-10 and converted to DUCHIJ. Lingering zeroes will be ignored, so 4003 is simply rewritten as 4-003 or 4-3 or DC.

[5] For example, if "TEST" denotes the key "BAY385," then (TEST + 5) → GFD830. If the value of x is to be negative denoting a change in the opposite direction, parenthesis shall be used the clarify and the addition symbol will still be present (i.e. TEST + (-3)).

"ciphertext" will refer to these new numbers – the original strings for both can be discarded. From this, observe the following on how to combine the converted numeral plaintext and ciphertext to form the **TARGET** key.

Let m be the amount of digits in the plaintext and let n be the amount of digits in the ciphertext.
Note: "eight fractional non-zero digits" means a result of "4.00003320718024" would yield "33207180" as a key

Instruction	Operation
If the digit in the n^{th} index of the plaintext has the same parity as the digit in the $\|m-n\|^{th}$ index of the ciphertext. If and only if the above does not make logical sense, determine if the digit in the m^{th} index of the ciphertext has the same parity as the digit in the $\|m-n\|^{th}$ index of the plaintext.	The first eight non-zero digits of the fractional part of: $\frac{\sum_{k=1}^{m}(n(k+m)^n}{\sum_{k=1}^{n}(k+\frac{1}{m}-1)^{ln(n-1)}}$ is the target key.
If the plaintext or ciphertext contains more than four successive single-digit prime numbers	Determine the prime factorization of $m\,(m+n)^n$. Concatenate the prime numbers in ascending order into one number, then concatenate the respective exponents (multiplicity) in the same order as the primes (not ascending). The first eight non-zero digits of the fractional part of the first concatenated number divided by the second is the target key. If the fractional part does not exist, add m to each concatenated number and repeat until found.
If the number obtained by concatenating digits in the odd indices of the plaintext is a different parity as the number obtained by concatenating digits in the even indices of the ciphertext.	Using the number made up of the first five digits in the ciphertext (pad zeroes on the left if number is less than five), find the first occurence of this five-digit number in π. Then, let the next eight digits following the found number be the target key.
If the first and third above conditions are met.	Repeat the operation done if the third condition is met except use the number obtained by the first *five* non-zero digits of the fractional part of the quotient obtained in the division operation from the first condition.
If a different combination of the two above conditions are met.	Determine the number obtained by multiplying the plaintext by the ciphertext and call this "NUM." Within this, generate a new number by ordering each digit in NUM in descending order, but keep track of the index of each digit in NUM that is being ordered. If two digits share the same value, the smaller indexed digit can be treated as larger. Concatenate these index digits together to obtain a new number. The first non-zero eight digits of the fractional part of the indices digit number divided by the descending ordered number is the target key.
If all of the above conditions are met	The target key is the first eight digits composing the fractional (decimal) part of the larger of the roots of: mx^2+ny-m^n. If the roots are imaginary, observe the complex portion of the root. If the result is a double root, still use the fractional part unless otherwise contradicting the statement below. If the root is whole or if there is more than three insignificant successive decimal zeroes, use the operation listed below (for "none of the above")
If none of the above conditions are met	If the value of n or m is less than 10, pad zeroes on the rightmost side of both the plaintext and ciphertext are at least ten digits long. Next, each *single* digit in the plaintext serves as a index within the ciphertext ("0" value is tenth index). Generate a number by determining the value of the digit in the ciphertext at the index given from the value of the digit in the plaintext. Generate a new number by inverting the process via using digits in the ciphertext as indices for digits' values in the plaintext. After dividing the first generated number over the second (i.e. N_1/N_2), the first eight non-zero digits of the fractional part is the target key.

PART FOUR: Generating Alphabetic Key for Password

After the above, you will obtain an eight digit number. Follow each instruction below in the table on transforming four of the digits in the number you obtained above.

Index in	Instruction

Number Affected	
1	If the numbers in the *first* and *second* index are **both** odd, change the number in the *first* index to a "1" Otherwise, change the number in the *first* index to a "0"
2	If the number in the first index was changed to a "1", observe the following: Let x be the value of the digit in the *seventh* index. Change the number in the *second* index to the value of x *modulo* 3 Otherwise, if the number in the first index was changed to a "0" leave the number in the second index as is. If this number is a "0" change it to "1"
3	Let y be the value of the digit in the *eighth* index. Change the number in the *third* index to the value of y *modulo* 3 If y *modulo* 3 and the digit in the fourth position are both zeroes, change the digit in the third position to a "1"
5	Change the number in the *fifth* index to a "1"

This number now yields the **DATE** key in the form MMDDYYYY. As is the eponymous name, this number string corresponds to a date within the second millenium AD of human history. The instructions below now detail what is to be done to determine the alphabetic character portion of the password.

Use the information given in Appendix B: Monarchs of the World to determine which monarch began his/her reign on the date nearest to what was found as the date key (*always* rounding up). For example, suppose in the appendix, the two Gondorian kings Narmacil I and Calmacil were adjacent and began their reigns respectively on January 12, 1226 and March 16, 1227. If your date key read "12171226", you would read the row that lists Calmacil and the relevant information. If you are rounding up the date and find more than one monarch listed on the same date, read the first one listed sharing the date. However, if you find the exact date and there are more than one monarchs listed, choose the one listed last.

After determining the monarch of interest, use the information listed in the row combined with the following to determine the letters that are to comprise the alphabetic portion of the password. The term "of interest" in the following refers to the selected monarch and "of interest" will not refer to any other row.

From all words listed, remove all special characters, spaces, hyphens, and any non-Latin alphabetic characters. If a letter is seen with a symbol, use the English letter equivalent (i.e. ü → u). Convert all letters into upper case.

If the regnal name has more than one word, concatenate them together to obtain the new name (do not include the regnal number if it is within two names; e.g., Harald II Svendsen → HARALDSVENDSEN)
If the regnal number of the monarch is greater than one (assume its absence means one), observe the following:

If the regnal number is odd, shift each letter in the regnal name to the *right* by that number[6]
If the regnal number is even, shift each letter in the regnal name to the *left* by that number
Otherwise, if the regnal number is one, reverse the order of the letters in the regnal name such that the last letter becomes the first.
Call the new string composed of the converted regnal name, the **MONARCH** string

If the nation the monarch hails from is Asian (assume the Ottomans and Byzantine Empire are), let v be the number *month* in the monarch's beginning reigning year. Excise the number of letters from the right of the House/Dynasty corresponding to $v \bmod 4$ (i.e. May: HABSBURGS → HABSBURG).
If the nation the monarch hails from is European, let w be the *day* of the month the monarch began to reign at. Excise the number of letters from the left of the House/Dynasty corresponding to $w \bmod 5$ (i.e. May 23th: HABSBURGS → SBURGS).
Otherwise, convert the first letter of the House/Dynasty into a number (A=1,..., Z=26) and let that number be x. Remove each letter in the House/Dynasty who's index in the word is a multiple of $x \bmod 3$.
Call the new string, the **HOUSE** string

[7]If the nation listed *below* the nation of interest is from the same country, excise the first letter from the nation of interest.
If the nation listed *above* the nation of interest is from the same country, excise the last letter from the nation of interest.
If the nation above is the same as the nation below, remove both the first and last letter from the nation of interest.
If the nation listed begins with a vowel, remove all vowels from the nation of interest
Otherwise, if *none* of the above four statements applied, keep only the first five letters and remove any others past.
Call the new string, the **NATION** string.

Order the monarch, house, and nation string in order of length, from longest to shortest. If two strings are the same length, use this expression to order them correctly (NATION > HOUSE > MONARCH). From this, concatenate all three strings into one. With this, select every *third* letter from the concatenated string until you have six letters altogether (wrap back around if there are not enough letters;
e.g., "ZOLLERN" → "LROEZL"). This is the **ALPHABETIC** key.

Use the target key from part three and likewise, select every *third* number until you have six numbers altogether (wrap back around when you finish the first pass; e.g., "12345678" → "361472"). This the **NUMERIC** key.

Finally, if the numeric key is *odd*, interweave and concatenate the alphabetic and numeric keys such that the combined string reads *letter first then number*. (e.g., "LROEZL", "274163" → "L2R7O4E1Z6L3").

[6] Using A=1,...,Z=26 as used throughout this document. Note also that adding beyond 26 will wrap around back to A.
[7] Each iteration through these "if statements" should be treated as if the nation of interest is being spelled differently throughout (e.g., the nation of interest is "ARGENTINA" and the nation listed below is "ARGENTINA" so the first letter is excised to become "RGENTINA". The nation listed above is also "ARGENTINA", but since it is the same as listed above the last letter of "RGENTINA" is *not* excised. Finally, since the new string "RGENTINA" does not begin a vowel, the vowels will not be removed.

If the numeric key is *even*, interweave and concatenate the alphabetic and numeric keys such that the combined string reads *number first then letter*. (e.g., “LROEZL”, “361472” → “3L6R1O4E7Z2L”).

The final string that is generated by the previous step is the **PASSWORD**. With this information, the sender should generate an email (via GMail) with the username being USERNAME and the password being PASSWORD. If prompted for any other information, the sender can put any information that allows the email to be set up. Any special characters that are left in the username (i.e. equal signs, question marks, etc.) should be deleted.

When the receiver has deciphered the message, he should be able to login to an email with the respective username and password he has found as well. If both members have found the same information, they should be able to email themselves and contact.

All discourse messaging should be run through Base64 encryption. Subject headers should be one of the following: ROUTINE, PRIORITY, IMMEDIATE, or FLASH.

Appendix A – Assistance

There are many necessary conversions, calculations, and encryptions necessary. You can use the following:

To check if a **number is prime**.
http://www.math.com/students/calculators/source/prime-number.htm

To search through **digits of pi**
http://www.angio.net/pi/

To determine the **prime factorization** of a number.
https://www.calculatorsoup.com/calculators/math/prime-factors.php

To compute **large numbers** or find **extended decimal places**
http://www.calculator.net/big-number-calculator.html

To convert **alphabetic characters to numbers**. Be weary of converting numbers to letters as there is a certain method required by this manual that may not be incorporated by this conversion software.
http://rumkin.com/tools/cipher/numbers.php

To encrypt using **Playfair Cipher**. Insert the calculated keyword and plaintext where asked at the bottom of the page. Remember to ensure to check "Omit Q."
https://www.braingle.com/brainteasers/codes/playfair.php

To encrypt using **Base64**. Remember to make sure to have UTF-8 as the output charset.
https://www.base64encode.org/

Appendix B – Monarchs of the World

The following is a table relevant to Part Four. The first column represents the monarch, the second represents the date, the third represents the nationality/empire, and the final represents the house/dynasty the monarch prevails from. Each row is ordered in chronological order going from top to bottom on one column of the page then moving to the next column on the right and repeating until a new page.

To obtain this document, please refer to the document entitled "Appendix B – Monarchs of the World" listed in the same folder as this document.

Appendix C – Example

Using the example DATE and TEXT key listed on the initial page, we will determine the confirmation key and the secure line.

DATE: 199201122003
TEXT: TESTINGXXX

Since the date key ends in an odd number, we will use the respective first table. Moving down the vertical column, there are three X's which is greater than or equal to three. The text key does contain an even number of vowels (two, "E" and "I"). Finally, since "T" corresponds to 20 and "E" to 5, 25 *is* less than or equal to 30. Thus, all of the vertical conditions are met.

Next, since there are 6 odd numbers and 6 even numbers, there is *not* more even numbers than odd. The date key is *not* prime as it is divisible by 3. The sum of the digits in the date key is 30, and the average of this over 12 numbers is 2.5 which is *not* greater than 4. Thus, none of the conditions are met.

With these conditions, we move the command that shares both: (DATE)^||((TEXT)/(VOW.))^[8]. Firstly, to compute (DATE)^, it is helpful to rewrite it as 19-9-20-11-22-003 which is simply "SITKVC." Next, ((TEXT)/(VOW))^ is simply just (TSTNGXXX)^ as E and I are the only vowels. Thus, this becomes 201920147242424. The two are concatenated together to form the new code SITKVC201920147242424.

From this, observe converted to letters, the key becomes SITKVCTSTNGXXX. Since there are 14 elements, the first seven (SITKVCT) becomes the keyword and the second seven (STNGXXX) becomes the plaintext. Using the Playfair Cipher, the ciphertext obtained reads "IKWNYYYYYY". Finally, via Base64, we obtain the username of "SUtXTllZWVlZWQ==" which will be recorded (and the equal signs will be deleted with respect to username syntax rules). The sender will send the keyword "SITKVCT" and if the receiver believes this to be accurate, he will reply "PROCEED" and they will move to set up a secure line.

Now, to determine the password in part three, both the plaintext and the ciphertext needs to be converted into numbers as so (STNGXXX / IKWNYYYYYY) → (1920147242424 / 9112314252525252525). Now, from these two numbers we can observe that $m = 13$ and $n = 19$. Now, we will check each instruction.

Since the 19th position in the plaintext does not make logical sense, we observe the number in the 13th position in the ciphertext — a "5" which is odd. The number in the $|13\text{-}19|^{th}$ position in plaintext is equivalent to the 6th position — a "4" which is even. Since the two are not the same parity, the operation will not be evaluated.

Next, since the ciphertext more than three successive prime numbers (seen in the "2525…" string), this operation may be evaluated pending if the third instruction is not true.

The number obtained via the odd positioned numbers in the plaintext is 1217444 and the number obtained via the even positioned numbers in the ciphertext is 121222222. Since both have the same parity of even, the operation is not evaluated.

Thus, to determine the target key, we must first find the value of $m\,(m+n)^n$. The number obtained is 514983056342718194258035677184. Using a prime factorization calculator, we obtain $2^{11} * 29^1 * 73^1 * 79^1 * 719^1 * 142271^1 * 502501^1 * 29250469^1$. With this expression, we now determine the two numbers we are later going to divide: the first being the base primes and the second being the exponents. We find the two numbers being 22973797191422715025012925 0469 and 111111111. To twenty significant figures the answer we get is 2067641749295686101546.84935576784935576784; however, this is not what we need. We only need the first eight non-zero fractional digits being "84935576" — which is also the target key.

With this number, we can move onto Part 4. Since the numbers in the first and second position are both *even*, the number in the first position is changed to a "0". Next, since the number was changed to a "0" the second position number can be left alone. Next, to determine y we find the digit in the eighth (or last)

[8] The concatenation symbol (||) has been added to save space. It is logically equivalent to the appearance on page 2)

position being a “6.” Since 6 *modulo* 3 is “0” the digit in the third position is changed to a “0”. Finally the fifth digit is changed to a “1”. Altogether, we arrive at the date key of “04031576”

Now, using Appendix B, we find the date of 04/03/1576 corresponds to Rudolph II as the date gets rounded up to 10/12/1576. The first string of interest in the monarch string. Since the regnal number is greater than one (it is two), we observe the statements listed. Since the regnal number is two, an even number, we shift each letter to the left by two. From “RUDOLPH”, we now obtain “PSBMJNF” for the monarch string.

Next, since the Holy Roman Empire is in Europe, we look at the second statement. We let w be the day of month the monarch began to reign — here it is “3”. We now are going to excise three numbers from the left of the house name (as 3 mod 5 is 3). Thus, HABSBURG becomes SBURG. Thus “SBURG” is the house string.

Finally, since both the nation listed above and below are not the same as the Holy Roman Empire, the first and letter won’t be excised (with respect to the first two instructions). However, since the nation listed above and below the Holy Roman Empire are both the same – Russia – the first and letter letters will be excised anyways. We now obtain HOLYROMANEMPIRE → OLYROMANEMPIR. Next, since the new string does begin with a vowel, all vowels will be removed to obtain “LYRMNMPR”. This now is the nation string.

To obtain the alphabetic string, we first must concatenate the three strings in order of length. Here the order is nation, monarch, house (as 8 > 7 > 5 letters). Thus, we first find the string of “LYRMNMPRPSBMJNFSBURG”. With this we now select every third letter until we have six. For assistance, it may be helpful to rewrite the string as “LYR-MNM-PRP-SBM-JNF-SBU-RG” to visualize. Thus, the six letter alphabetic key now is “RMPMFU”.

To obtain the numeric string, we compute the same process on the target key from before — “84935576”. Again for assistance it may be helpful to rewrite the string as “849-355-768-493-557-684-935-576”. Thus, the six number numeric key is “958374”

Lastly since the numeric key is even, the two will be interwoven and concatenated such that the number precedes the letter. Thus, we get “9R5M8P3M7F4U” as the password.

From the initial date and text message that was recorded, we determined that the username for the secure email is “SUtXTllZWVlZWQ” and the password is “9R5M8P3M7F4U”

This is the completion of the example.