Saskatchewan Polytechnic, CTO Report for CEW Systems Canada Inc.

on

Post-Quantum: Bi-Symmetric Hybrid Encryption System

by

Dr. Cyril M. Coupal, PhD, ISP, ITCP
Senior Research Associate, DICE
March 2021

Introduction

CEW Systems Canada Inc. has asked the Saskatchewan Polytechnic Digital Integration Centre of Excellence (DICE) group to perform a short CTO-funded analysis on their Bi-Symmetric Hybrid Encryption System. The author is a member of DICE. In this capacity he has performed some investigation into the current state of encryption complexity and possible quantum computer-based threats to the encryption robustness employed in the CEW application. It is not our intent to describe the system or its workings in detail, other than as necessary in describing what has been performed and observed. We conclude with a short summary of strengths and weaknesses as determined by the author.

Overview of the Approach

Bi-Symmetric Encryption uses a unique and novel handshake incorporating encrypted session key combinations, allowing user's login credentials, biometric data, credit card data, or command/activation codes to be quickly and correctly processed, without directly transmitting this confidential data.[1] The plug-and-play, hybridized encryption system employs concepts like asymmetric encryption meshed with more secure symmetric encryption. A significant difference from commonly employed asymmetric encryption is that during the initial handshake to set up communication, no vulnerable data are exchanged. Should the sender key communication be intercepted by a hacker, they still cannot pretend to be the originator of the communication to the receiver.

The encryption itself is achieved by randomly generating keys and interweaving them with portions of unencrypted data to be transmitted, applied to single bytes of data rather than long byte collections. During the initial handshake, private keys are generated from or found in the form of login credentials, credit card information, biometric data, or other personal credential information or pre-shared private keys, which are then used to start the handshake and are never actually transmitted. Randomly generated data in the form of challenge codes, counter

[1] Embracing a new era of cyber security: Post-Quantum and Internet of Things, Bi-Symmetric Public/Private Hybrid Key Exchange Encryption System for the Automotive Industry, CEW Systems Canada Inc.

challenge codes and session keys are exchanged during the handshake. This allows for the client and server to ascertain that the communicator, at the other end, are who they say they are. Once the client is satisfied that the server is who they believe it to be (by the server properly modifying the client challenge code then sending it back) and the server is satisfied the client is who they say they are (again by correctly modifying and sending back the server challenge code) a fully encrypted session is established, and communication can proceed. During the regular portion of the session, data are encrypted multiple times in a specific way so that only the correct receiver can decrypt the communication directly. Mathematical formulas are not used for encryption/decryption. Instead, algorithmic protocols are used. The multiple levels of interweaving together the various encrypted bytes into new data sets which are finally transmitted to the receiver who then reverses the whole process to decrypt the message.

An important aspect of the encryption is that plain text characters in the data packets are modified individually instead of in groups or blocks, meaning that there are no overall mathematical relationships that can be identified. Each packet to be encrypted uses a different set of keys adding greatly to the complexity of the encrypted message. Several benefits result. Brute force attacks have no way in which to identify if an attempt to unencrypt a portion of the message results in valid useable data. Hence any possible outcome is as likely as any other outcome. When billions of possible outcomes exist, it becomes impossible to determine the correct one.

The Bi-Symmetric encryption system is intended for use in various levels of security requirements. Bluetooth level uses the smallest packet sizes, shortest key lengths, and least amount of interweaving. The intended context for this level is use in keyless lock entry (ex. car fob) or other Bluetooth based communication between devices, typical of private use applications and IoT devices.

Commercial applications form the next level of security robustness. These applications would include virtually any kind of transaction conducted over the internet, whether it be interaction with one's bank, credit card account, or making online purchases.

The highest level of security robustness includes government and military applications. This level of security is many factors more complex than either commercial or Bluetooth applications.

In all cases, the inherent complexity of hacking the message by determining the keys is many magnitudes greater than current standard internet encryption protocols; all of which are considered secure against present standard supercomputers.

Threat from Quantum Computing

Perhaps the most significant and imminent threat to encryption security and possible breaking of the encryption keys, are the rapidly increasing capabilities of quantum computers. The capabilities of quantum computing to reduce computationally hard problems, such as molecule modelling, to achievable solutions, are remarkable[2]. IBM is currently offering a series of quantum computers (up to 51 qubits) in the cloud for use by anyone. They have defined a path of achievement that suggests the number of qubits in their computers will double every year and by 2023, plan to debut a 1,121-qubit computer called the IBM Quantum Condor.

It has been observed that a large enough number of qubits and support memory will be able to break RSA-2048 keys[3]. With a 4000-qubit computer and 100 million gates, it theoretically becomes possible to factor 2048-bit keys; Shor's algorithm[4] may take 10,000 qubits. However, if IBM succeeds in their development path, a 10,000-qubit computer may be available as early as 2027, a mere 6 years from now. Of course, these capabilities must still work within an acceptable amount of time to be useful to a hacker[5]. If the processing requires days, weeks, or months to complete, they lose their effectiveness as a code-breaking solution. However, the threat is still real and not that far off. The approach taken by the CEW system is suitable to thwart even a quantum computing threat because of the nature in which they use multiple keys at both ends of the communication. Even with a quantum computer to assist in breaking the encryption keys, the large number of keys and seemingly random way in which they are applied still makes it unlikely that a key discovery will occur. In addition, since they apply and change keys, during the interweaving steps, even if keys were determined, the decrypted message would have many possible values making deciding which is the correct value virtually impossible.

Interweaving produces a result that has a very large, effective key set. For example, a 40-byte x 8-bit packet, generates approximately 7.16E+172 possibilities. By comparison, a standard 40-byte RSA key generates 1E+40 possibilities[6]. The characteristic curve for factorization[7] is shown here:

---

[2] https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/

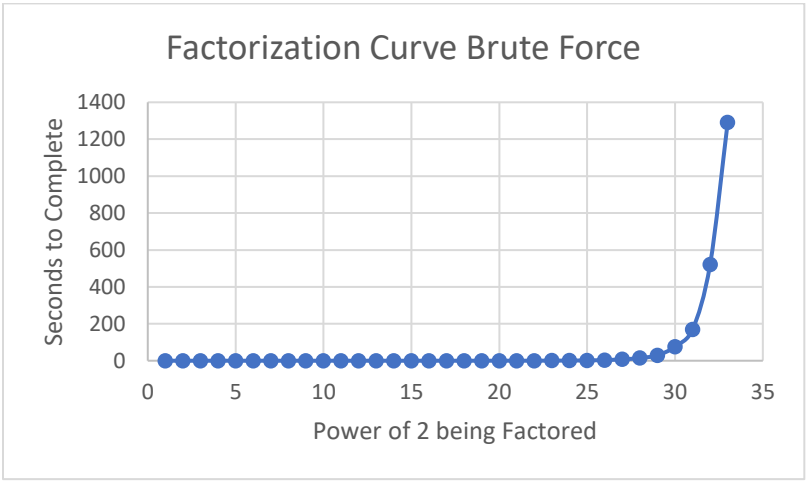
[3] https://security.stackexchange.com/questions/87345/how-many-qubits-are-needed-to-factor-2048-bit-rsa-keys-on-a-quantum-computer

[4] https://arxiv.org/pdf/quant-ph/0301141.pdf: Shor's discrete logarithm quantum algorithm for elliptic curves, John Proos and Christof Zalka, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario Canada N2L 3G1 e-mail: japroos@math.uwaterloo.ca zalka@iqc.ca February 1, 2008

[5] It is often stated that a quantum computer can achieve the breaking of encryption keys, but it is not stated within what time frame are we theorizing

[6] It would require a classical computer roughly 3.17098E+23 years to break the code at 1,000,000,000 operations per second trying all permutations of the 40-byte key.

[7] Python script run on a Dell GEFORCE i9

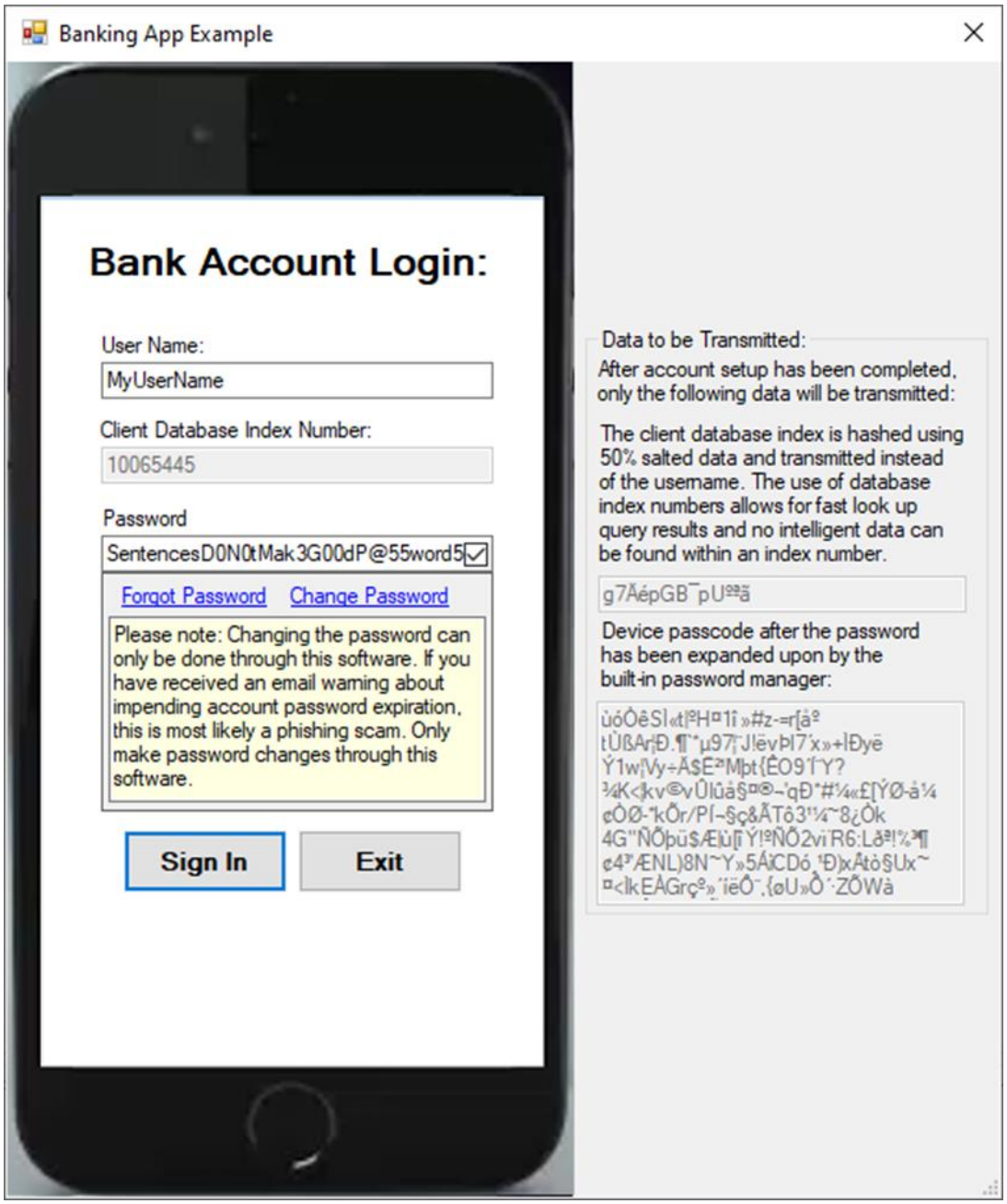


Shown is the time taken to compute all factors for a given power of two. As can be seen, each loop takes twice as long as the previous. In a short number of loops the time will become unmanageable. For standard computers, the current factorization problem for determining the encryption keys of 2048 bits, is beyond capability.

Other Benefits of the System

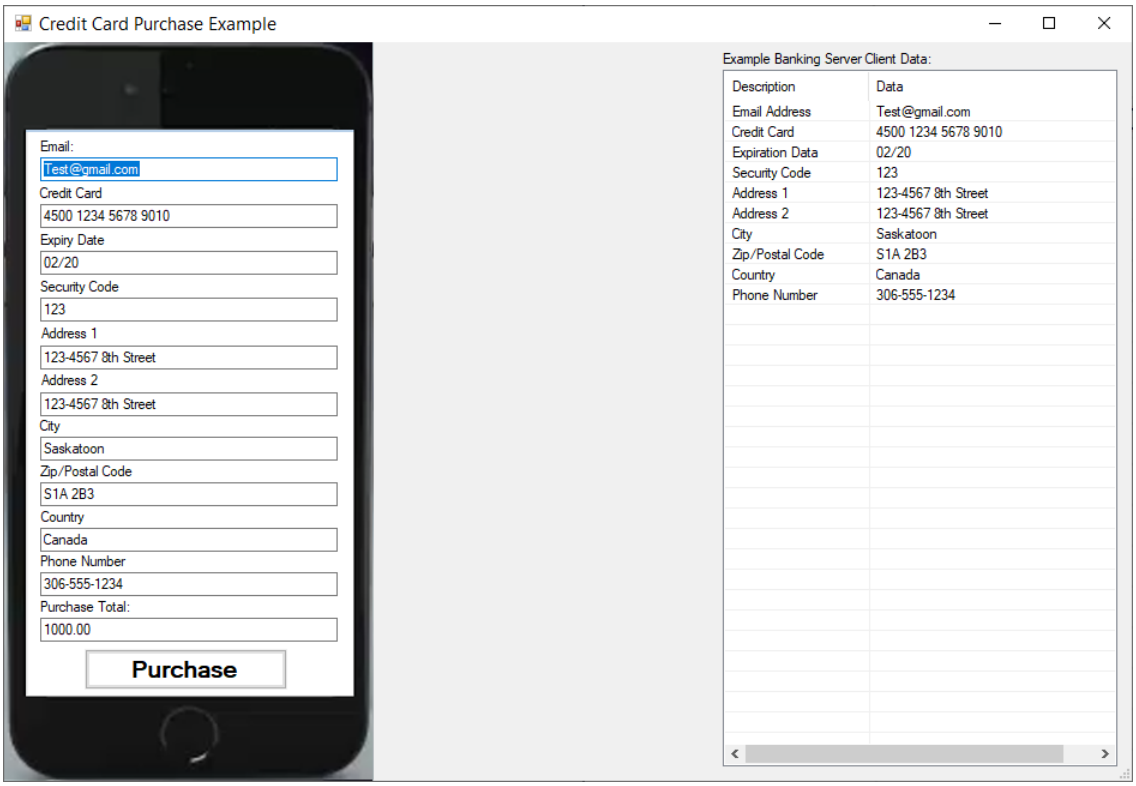
It may seem contrary that overhead processing of the Bi-Symmetric Encrypted message does not add significant delays to encryption/decryption (as reported by CEW during a series of encryption runtime tests). This seems reasonable when one understands that instead of processing large byte sets in encrypted blocks, the system encrypts small blocks but with a large set of keys. Thus, processing is very fast while still secure. This is why the CEW white paper calls the system the fastest, smallest, and largest of the encryption techniques.

Much has been explored about ways in which the Bi-Symmetric Encryption handshake is ideally suited for encrypting pay-per-use e-commerce transactions. Consistent with all communication, the customer's account information is never actually transmitted. The customer's username, ID number, or database lookup index number is sent as salted encrypted data, so the server can lookup the customer's account. The use of a database lookup index number allows the database server to access the user's account more quickly without having to do a database query search. The following image shows the data that is actually transmitted during the exchange:

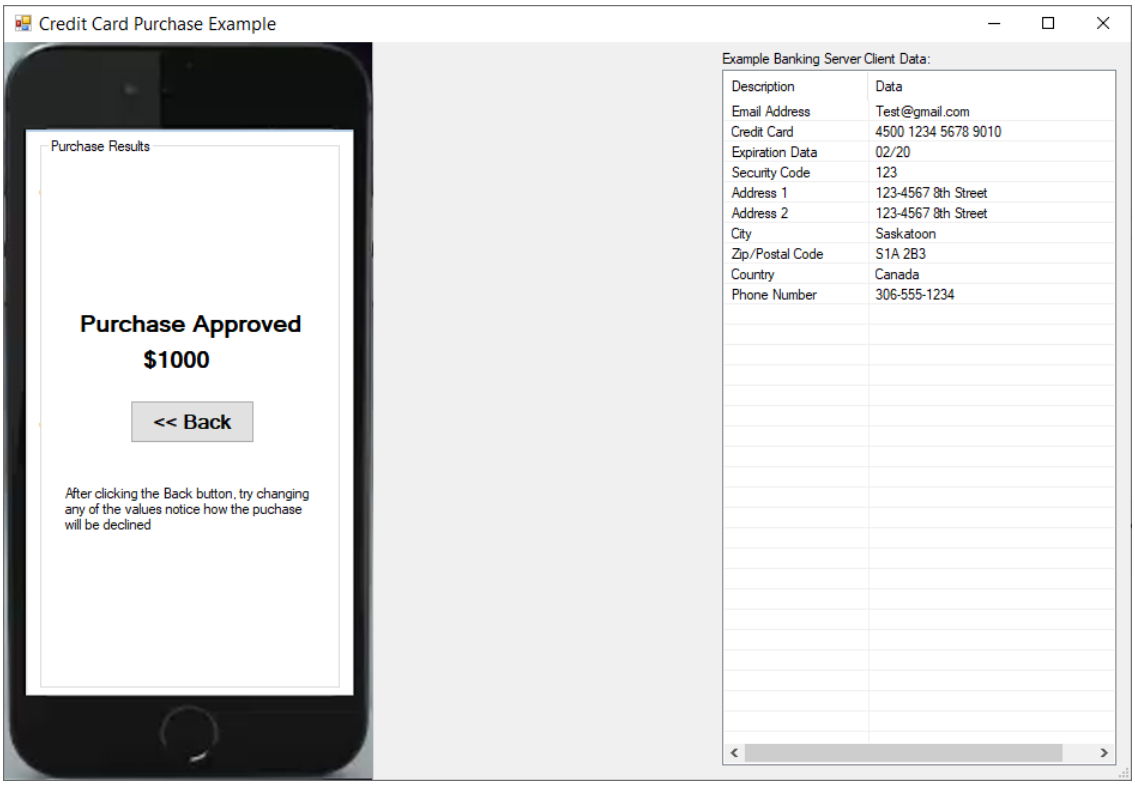


The crucial security point is that the customer's account card containing the private pre-shared keys would be used as the master keys for the handshake allowing a purchase to occur without transmitting the private keys. The use of internal database values helps to further protect from MIM attacks. Note that the database index must not be tied to physical data locations because those locations could change if the database were moved or restored from a backup. Any database-managed keys (index, constraint, etc.) would be suitable.

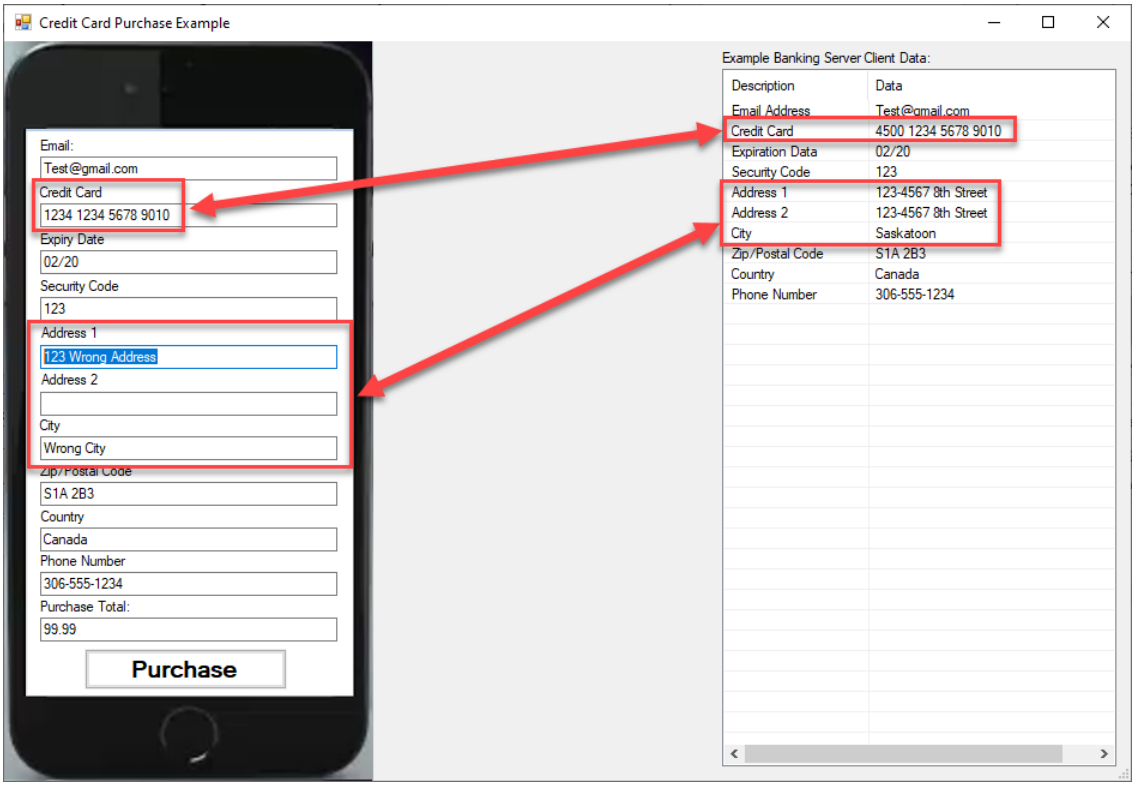
Another important feature is that the non-transmitted data that each system (client/server) has, must match or the interaction is dismissed. For example, consider a credit card purchase. Here are the screens of the purchaser (client) and the retailer (server):



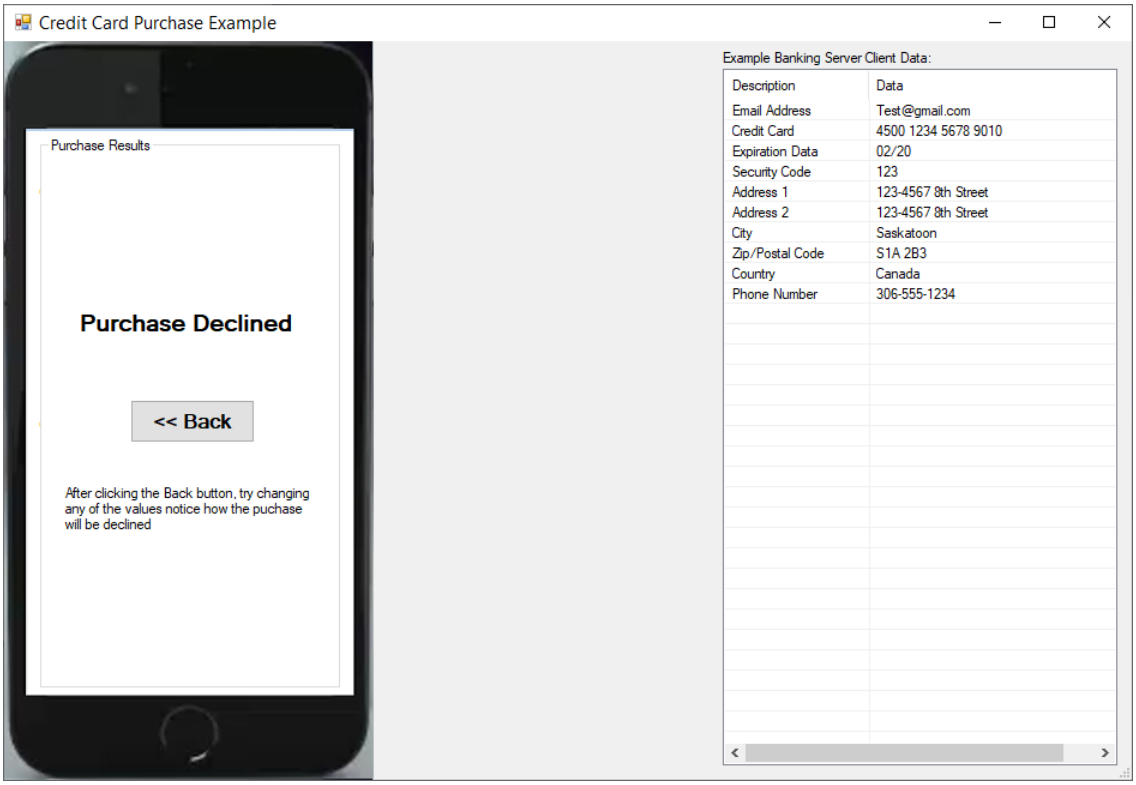
In the above scenario, the information on the client matches that on the server. Even though the data itself is not exchanged, session keys based on the data are generated at each end of the communication. In this case, when the client clicks the “Purchase” button, the purchase will be approved because the session keys at each end encrypted and decrypted correctly.



Now consider where some information is different (perhaps it has been illegally obtained).



When the “Purchase” button is clicked, the session keys at each end of the communication will be different because the critical data upon which they were created are different. Since the server cannot decrypt the message correctly, it will decline the transaction:



What about initial secret exchange and account setup?

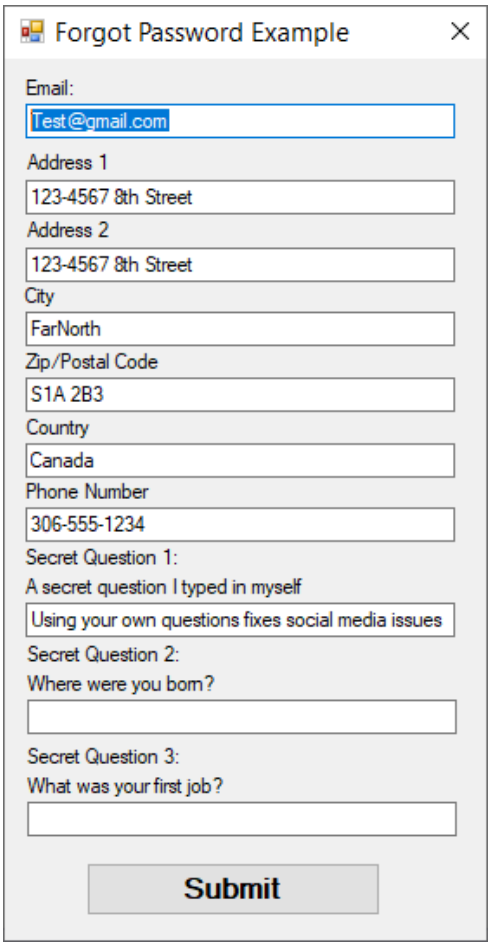
Common secrets, known to both server and client, must be exchanged when initial set up of accounts is made. Various methods exist to do this, but most involve the human factor, which is dangerous.

If setting up an account with a bank, or credit card, one must (still) visit in person to initially establish identity. Once this is accomplished, remaining interaction with the bank or credit institution is likely to be electronic. If in-person is not possible, a secure postal service, such as registered mail, can be an alternative to sharing the common secrets.

When the initial interaction is online only, such as when setting up an account with an online retailer, the initial secrets have to be exchanged online as well. A suggested approach is based upon using a 3rd party where a person would already have exchanged pre-share private keys. It is a variation of the “trusted party” identification used in standard RSA to ensure the response, encrypted with the public key, is in fact coming from the person it is supposed to and

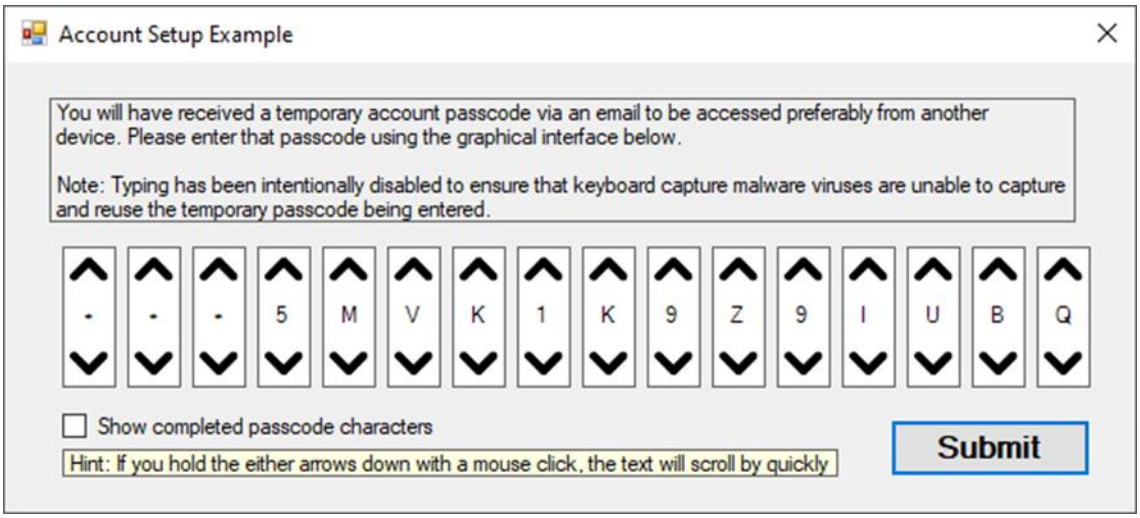
not a hacker. In the Bi-Symmetric case, the trusted third party could be a credit card company. The credit card companies could be incentivised to provide this third-party confirmation/key exchange as a service that could be offered for a small fee. The following describes a potential interchange.

A standard credit card purchase using the Bi-Symmetric handshake, which relies upon pre-shared private keys, uses the purchaser's credit card data and billing address as the pre-shared private keys. When a user goes to sign up for a new account with, say Netflix, the Bi-Symmetric Handshake requires pre-shared private keys. Using a credit card transaction at account setup can provide the required pre-shared private keys. Upon account setup, a user would be prompted to first enter their credit card information to confirm that they have both a valid credit card, while also confirming who they truly are. The standard Bi-Symmetric encryption handshake occurs between the e-commerce app (could be provided by Netflix or a third-party e-commerce app) and the credit card company. The credit card transaction occurs which creates the required set of Bi-Symmetric session keys as part of the standard transaction. If the transaction is approved, the server then starts a second Bi-Symmetric handshake directly with Netflix, then transmits the original transaction session keys to Netflix. Now the user and Netflix both have the identical transaction session keys, and the user can finish the account setup. The user's e-commerce app then uploads the user's address information to Netflix. The user is then free to enter additional required data specific to Netflix's product offerings and approved family member users, all securely encrypted. Subsequent logins to the account happen in the normal manner. Additional identification authentication using usual methods can be added to the trusted data. For example, answers to security questions can be added for additional authentication. It is suggested that a user-supplied question be added to the typical set of standard questions to help thwart social media scraping. This additional information can be used to support changing passwords, or other secret data shared by both client and server:



How does Netflix receive the credit card data so that they can then process the user's monthly subscription? This is accomplished in that when the approval transaction occurs, the credit card server takes Netflix's corporate account private keys, combines them with the user's private credit card keys and encrypts them together with Bi-Symmetric encryption. The combined key set is then sent to Netflix during the initial transaction as an additional data packet. These combined keys cannot be separated by hackers and cannot be used to purchase anything through an ordinary e-commerce purchase user interface. Of course, it is always possible that, somehow, a hacker could non-digitally steal the set of secrets. However, the nature of the protocol means the worst that could happen is the hacker would end up giving money to the company (Netflix).

Initial account setup often requires additional steps to establish and record secrets which must consider possible key logging, screen capture and other attempts to invade the privacy of the data during the set-up transaction. One approach for entering such data, suggested by CEW, involves using rotation wheels of characters (special and alphanumeric depending on what is being entered) similar to a rotating wheel found on luggage, as shown here:



Each time an entry of data is required, a number of wheels appear with randomly generated characters. Thus, it can not be predicted from which start character the user will click the wheel to find the correct data element. Key logging can count the clicks but without knowing the start position of the wheel, the count is meaningless. Although this is an issue with all security systems during critical initial data entry, the approach suggested here decreases even further any likelihood of easily capturing secret data.

The implementation algorithms cannot be open source.

Knowing the procedures would aid in hacking the keys, therefore, the actual implementation of the algorithms, as well as the algorithms themselves, must be kept secret. The interweaving protocol is not mathematically based, but procedurally based. Of course, the data secrets for each client-server interchange must also be known, which is highly unlikely. CEW has many protocols in place to keep their application code secure. However, this may cause difficulty in obtaining certification by security agencies if they cannot inspect the code for security issues and thoroughness. Finally, it is not currently known how easy it would be to reverse engineer a copy of the executable code. In a manner akin to salting data values before hashing, extraneous code could be added throughout the actual code to hide the true nature of the functions within the application. This is something that can be undertaken in the future should it become an issue.

Conclusion

The new and novel Bi-Symmetric Encryption system reviewed here offers multilevel quantum resilient encryption technology that has been specifically designed to be immune to brute force

attacks, man-in-the-middle, and rolljam[8] attacks. Wherein other encryption programs only provide a token key exchange, or 2-Factor Authentication (2FA), the Bi-Symmetric Encryption system is designed with an exponentially leveled multi-factor authentication system. The Bi-Symmetric Encryption handshake allows for pre-shared private keys, login credentials and command codes to be processed by a receiving device or server without the need to transmit the data directly. Bi-Symmetric Encryption is designed to be embedded within electronic devices and systems such as Internet of Things (IoT), automotive Remote Keyless Systems (RKS), autonomous systems such as driverless vehicles, as well as being ideally suited for online downloading of keys to allow smart devices to be used by vehicle owners to connect to their vehicles.

Through integration with online retailors, credit card companies and financial institutions, a higher level of security can be achieved for the millions of transactions that occur daily over the internet.

[8] A method to break into an automobile by blocking and recording the signal transmitted by a car key fob and then used by the recording device to access the vehicle. (https://www.hackster.io)