Mobile Networking Security, from Wordle to BlockChain

Albert Li
UML CS Student

Albert_Li@students.uml.edu
Lowell, Mass, USA

***Abstract*—This paper surveys the building blocks that provide security and privacy for mobile and web-based computing. The target readers are college students or digital citizens in general. Understanding the basic components is crucial to becoming a responsible adult in the digital world: not only can we avoid the pitfalls and leverage the technology tools effectively, but also can we adapt these building blocks and come up with novel solutions to problems in the real world.**

***Keywords*—Mobile Network Security, Cipher, Hash Function, Key Exchange, Digital Certificate, Central Authorities, Post Quantum Cryptography, Digital Literacy, Non-Fungible Token-based Advertising**

I. Introduction

It takes experts from multiple displines to explain the phenomenal success of the Wordle game; it took many brilliant mathematicians to break the Nazi's Enigma machine. Both the Wordle players and the WWII code breakers took advantage of language properties like letter frequencies, word frequencies, vowel patterns, bigram and trigram patterns in their perspective games. But today's Wordle players have to thank those WWII era scientists, and mathematicians for the ubiquitous mobile and web-based computing devices. The Turing Machine, von Neumann Architecture, and Shanon's Information Theory are among the obvious. Still, many Wordle players gripe about the recent New York Times acquisition of the Wordle game due to some glitches experienced during the transition.

II. The Building Blocks

A. *The TCP/IP Communication Stack*

The internet started as a research effort initiated by the US government[4]. The IP (Internet Protocol) is packet-based and has a few layers; but there is no built-in security or encryption originally. At the very bottom is the physical layer that is responsible for moving data packets around locally. The IP layer is responsible for sending and receiving data packets beyond the local area, via one or more routers, reaching the final destination identified by the IP address. UDP(User Datagram Protocol, a connectionless, less reliable but more efficient in a low error rate local environment) and TCP (Transmission Control Protocol, guarantees data integrity, handles lost or duplicated data packet) are communication protocols one layer above IP. They are both transport layer protocols. A combination of an IP address and a TCP port number corresponds to an application that uses TCP. Similarly, A combination of IP address and UDP port number corresponds to an application that uses UDP. A server listens (waits for incoming data and then takes action on that data) on either a TCP or UDP port number that is well-known. A client connects to a server using an unused port number.

B. *Security at the Physical Layer*

The free Wi-Fi (Wireless Network) at coffee shops is often without security or privacy; Early versions of the Wi-Fi encryption standard have been proven to be ineffective due to design flaws[2]. One hundred percent perfect security at the Wi-Fi layer can only deter bad actors at the coffee shop. A bad actor accessing a router on the internet can still eavesdrop the un-encrypted data.

C. *Security at the IP Layer using VPN (Virtual Private Network)*

VPN secures information at the IP layer. Workers at the west branch can access east branch computing resources via VPN and vice versa (Figure 1 [2]). . Consumer-oriented VPN vendors let people living in Singapore watch ESPN sports programs that are not available for IP addresses outside the US via VPN. Since the data is encrypted and decrypted at the IP layer, VPN has to handle IP packets that arrive out of order or are duplicated, essentially performing roles belonging to the TCP layer[2]. Most popular operating systems implement VPN together with TCP/IP stack as part of the OS kernel or kernel module.

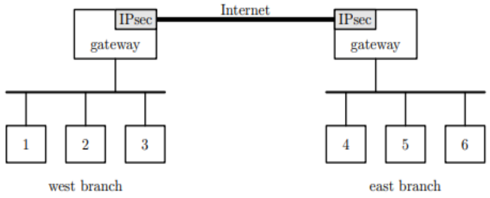

Figure 1 A virtual private network (VPN) between east and west office branches

D. *Security at the Application layer, SSH as an Example*

SSH (Secure SHell) is a widely used user mode command-line application to remote login to Linux/Unix or macOS servers without the need for a VPN. Researchers[2] explained two known attacks on the design and implementation flaws found in SSH version 2 protocol as case studies. Users of SSH should at least keep their software up to date to avoid pitfalls, and software implementers should learn from the lessons to avoid similar mistakes; and whenever possible, always consider using a standard implementation library described in the next section.

E. *General Purpose Security at the Transport Layer with TLS*

An e-commerce website is a different domain from a news website; It is impractical to establish a separate VPN link for each domain. A web browser uses HTTPS (Secure HyperText Transport Protocol) to protect users' securities and privacies. Underneath, the web browser in turn relies on the TLS (Transport Layer Security) protocol. TLS was originally called SSL (Secure Socket Layer) protocol, namely to provide a secure implementation of the regular Unix Socket programming interface. Netscape, transformed into today's Mozilla Foundation, designed SSL in 1994. Wagner et al. [8] analyzes SSL version 3.0 (considered to be close to TLS version 1) and also documents the growing pains for SSL up to version 3. The IETF (Internet Engineering Task Force) created the TLS working group and produced the latest TLS 1.3 protocol specification in 2017[2]. There are no major vulnerabilities known currently. TLS usually is implemented in a library that ships with the OS so that other applications besides the web browser can take advantage of it. For historical reasons, the SSH application does not use the TLS library.

F. *Modern Encryption Protocols*

The 1970s marks the modern era of cryptography with the research on public-key encryption, also called asymmetric encryption, from R. Merckle[5], Diffie-Hellman[3], and RSA(Rivest–Shamir–Adleman)[7]. It solves the problem of how to create or share a secret key between communications parties. The traditional secret key-based encryption, also referred to as symmetric encryption, still has its place in modern encryption because of its performance.

In the first "handshake" phase, asymmetric encryption is the method to establish agreements between the communication parties on hash function, cipher, and secret keys for the second phase.

The handshake protocol in TLS 1.3 can choose to use either a modified version of Diffie-Hellman key exchange or RSA to establish 4 shared secret keys between client and server.

The asymmetric encryption, revolutionary as it is, can still suffer from a man-in-the-middle attack.

G. *Digital Certificate and Central Authorities*

To thawt the man-in-the-middle attack, the TLS protocol requires the web server to prove its identity during the handshake phase using a digital certificate. The browser can also prove its identity to the server, but it is rarely done. Most people would prefer to stay as anonymously as possible. Few people would take the extra effort to obtain and install browser digital certificates.

To prove that a digital certificate is valid requires public key encryption. The public key of a server and its related information are encrypted with the Central Authority's private key to form the server's digital certificate. A web browser uses the built-in certificate authority's public key to verify the server's digital certificate during the handshake phase. The user has to trust the web browser vendor to build a list of correct Central Authorities' public keys into the browser's executable file; The web browser vendor will also update its executable file if a particular Central Authority's public key is revoked because the corresponding private key is compromised. It is the best practice for the user to update to the latest browser version for this particular reason.

H. *Public Key Encryption and Quantum Computing*

The Diffie-Hellman key exchange protocol assumes that computing powers of a prime in a finite field is an easy task, but the inverse question of solving a discrete log in a finite field requires enormous computing power. Similarly, RSA assumes that multiplying two large primes is an easy task but factoring the product into 2 primes is impossible with current computer technology. But those two instances of public key encryption may fall victim to Peter Shor's quantum algorithm developed in 1994[6]. To break today's public-key encryption, a quantum computer must have on the order of 1000 logical (fault-tolerant stable) quantum bits[6]. Current state of the art quantum computer has less than 100 quantum bits. Those quantum bits are not stable enough to be cryptographically relevant. Researchers are actively working on many proposed replacement algorithms for the post quantum era. One suggested post-quantum public-key encryption is based on the math problem called "Super Singular Isogeny Graph[2]". What keeps them from sleeping is that adversaries can store the encrypted data now and analyze them when a quantum computer powerful enough is available.

I. *Blockchain and Quantum Computer*

Those who wish to break the blockchain using future quantum computers don't even need to buy storage. Blockchain is a public ledger that is distributed all over the world. Researchers[1] indicate the elliptic curve signature scheme used by Bitcoin might fall victim to quantum computers. The hashing function SHA256 used for "prove-of-work" is quantum resistant. In other words, the future quantum computers won't be powerful miners but they can comprise the private keys. Bitcoins' latest "Taproot" software upgrade mainly focused on how to make transactions more private instead of switching to quantum resistant public key encryption.

III. Innovations on top of the build blocks

The building blocks are improving bit by bit through iterations. While no technology is perfect, it is impossible to throw out all the building blocks we have today and replace them with something new. All the innovations will happen on top of those. Web2.0 just demonstrated that. The foundation blocks are unchanged. We discover and then fix bugs and vulnerabilities, but the architecture remains the same. The "new" social media platforms build on top of the existing technologies and innovate on top of those. If the majority of the population were more educated, there would be no need for these new social platforms. The complexity of security and privacy goes beyond the technology building blocks. Social media firms start to work with lawmakers. The EU passed the "right to be forgotten" law. The evolution of blockchain technology is also interesting yet concerning. For example, as the bitcoin mining competition intensifies, individuals lost to corporations backed by investors using the latest ASIC (Application Specific Integrated Chip, in this case, chips designed only to do SHA256 hash in parallel), which does not bode well with building distributed and decentralized trust.

NY Times acquired the Wordle game hoping to drive smart and literate people to its website. If the Wordle players choose to store their winning streaks as NTF tokens on a blockchain, then NY Times, along with a grocery store and a coffee shop can all co-sponsor the game. The players can choose preferred rewards. Promotions to the potential customers will be more effective and consumers will get less annoyed by ads filled in the web and mobile applications.

Acknowledgment

Thanks to Prof. Levkowitz and all classmates who presented many interesting topics.

References

[1] D. Aggarwal, G. K. Brennen, T. Lee, M. Santha, and M. Tomamichel, “Quantum attacks on Bitcoin, and how to protect against them,” Quantum Physics, Submitted on 28 Oct 2017

[2] D. Boneh and V. Shoup, “A Graduate Course in Applied Cryptography,” version 0.5, January 2020, http://toc.cryptobook.us/

[3] W. Diffie and M. E. Hellman, “New Directions in Cryptography,” IEEE Transactions on Information Theory, November 1976

[4] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts and S. Wolff, “Brief History of the Internet,” Internet Society, 1997

[5] R. C. Merkle, “Secure Communications Over Insecure Channels,” Communications of the ACM, vol. 21, April 1978

[6] M. Roetteler, M. Naehrig, K. M. Svore, and K. Lauter, “Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms,” Proc. ASIACRYPT 2017, December 2017 https://www.microsoft.com/en-us/research/wp-content/uploads/2017/09/1706.06752.pdf

[7] R.L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Communications of the ACM, 1978

[8] D. Wagner and B. Schneier, “Analysis of the SSL 3.0 protocol”, https://www.schneier.com/wp-content/uploads/2011/09/paper-ssl.pdf