A report for Summer School on Real-World Crypto and Privacy June 17–21, 2019 Šibenik, Croatia

Farzane Karami[1]

[1]Department of Informatics, University of Oslo

July 23, 2019

In this report, I summarize some of the talks. In "Real-World AKE" talk Tibor Jager gave an overview of real-world authenticated key exchange protocols like TLS (Transport Layer Security) and SSL (Secure Sockets Layer). In this talk, he covered the Diffie-Hellman (DH) key exchange, Man-in-the-Middle attacks, and forward security. He also introduced TLS 1.3 which is more secure than previous versions. Diffie-Hellman is a way of creating a shared secret key between two parties to encrypt their data over the internet. For example, if Alice and Bob want to communicate securely, they first agree on a large prime number p, and a generator (or base) g (where $0 < g < p$). Alice chooses a secret integer a as her private key and then calculates her public key $g^a mod\ p$. Similarly, Bob chooses his private key b, and calculates his public key ($g^b mod\ p$). Then they can exchange their public keys; however, neither of them can calculate the other person's private key since it is a hard mathematical problem. Then with the public keys they can calculate the shared secret key as $g^{ab} mod\ p$. An attacker cannot calculate the shared key since they do not know the private keys of Alice and Bob.

This protocol is vulnerable to an attack named "Man-in-the-Middle", where an attacker in the middle takes Alice's public value and sends his own public value to Bob. On the other side, when Bob sends his public value to Alice, the attacker sends his own public value instead. Therefore, the attacker and Alice calculate one shared key, and the attacker and Bob calculate another. Now the attacker can decrypt or encrypt any messages sent out by Alice or Bob. This vulnerability can be solved by the signed Diffie-Hellman protocol.

Forward security for TLS and SSL is a way of guaranteeing that previous sessions of communication are not compromised just because a current session's key has been compromised. In a forward security mechanism, a new and unique key is generated for each session; therefore, a stolen secret key only discloses that particular session. Forward security of signed DH is a safer protocol for TLS and SSL, but generating new keys for each session, distribution of public keys, and the handshake protocol for key confirmation in the real world is not easy. He also explained TLS 1.3 handshake protocol.

Another talk by Ahmad-Reza Sadeghi was: "From Smart Cities to Smart Sex Toys: A Hitchhiker's Security & Privacy Guide to The Galaxy of Things". He talked mostly about IoT (internet of things) and how the smart devices in IoT can become compromised and launch attacks like DDoS or botnet. He presented approaches for automatic device identification and reliable detection of compromised devices, in which by machine learning abnormal communication behavior of devices in IoT systems are detected and identified.

Another talk by Lujo Bauer was "Adversarial machine learning: curiosity, benefit, or threat?". In this talk, he presented that the use of machine learning (ML) algorithms in safety- and security applications (like face recognition systems) can be risky. He talked about their research on face-recognition systems and machine learning algorithms. They have shown that machine learning algorithms in face-recognition systems are vulnerable to physical attacks. An attacker evades a face-recognition system by impersonating another person. An attacker wears a pair of specific and colorful sunglasses and then the algorithm recognizes the attacker as another person. They generated many different sunglasses, printed on papers, with different colors and shapes to simulate such attacks. They tested what frames and colors are more effecting for evading a face recognition system.

Patrick Schaumont gave a talk about "Hardware Acceleration in Cryptography". It is important to make the hardware implementation of cryptography efficient in both performance and energy consumption. This way in electronic devices, a cryptography computation uses less battery, and it reduces the power consumption. The goal of hardware-accelerated cryptography is to have less run-time overhead. Moreover, the hardware implementation should not be prone to hardware attacks (side channel attacks). One way to avoid hardware attacks is to isolate hardware-based secrets from unauthorized access. He explained "loosely-coupled memory-mapped structures and tightly-coupled custom-instructions" for isolation of hardware secrets. He discussed techniques that are used to transform sequential algorithms into parallel architectures for accelerating execution of a program. He also discussed the application of this technique to the hardware implementation of cryptography algorithms.