Review and Comment

Last time, we watched a video panel presented at last week’s RSA Conference. The panel consisted of Whitfield Diffy [notice that I have spelled his name correctly, as he insisted at the end of the conference], Ron Rivest, Adi Shamir, and Martin Hellman. What struck me was how the panel represented the whole spectrum between pure theory and pure application.

Diffie, who reminded me of many hardware geeks, was all over the place. He emphasized that crypto was only part of the whole machine, and emphasized the importance of the “Command and Control” viewpoint, although he did not elaborate. Diffie also noted that one of the problems with RSA is that the user has no mechanism to judge the quality of a key, while noting that RSA has the advantage of not needing randomization. Diffie also noted that there is a big difference between battlefield crypto and commercial crypto: on the battlefield, there is no possibility of appeal or review.

Ron Rivest had the strongest theoretical orientation, and still works as a professor in the Artificial Intelligence Lab at MIT.

Hellman was the most practical: he talked about the lack of provably secure systems, and how extensive backups increase security risk. He talked about the original estimates of how long it would take to break RSA–129, and of the hardware problems in scaling up from Deep Crack, a custom chip that broke DES in 3 days and cost $250,000 in 1998. [For more, see the Wikipedia article on Deep Crack, at http://en.wikipedia.org/wiki/Deep_Crack.] Hellman also made the rather theoretical observation that the rise of the internet bypassed the government’s former restrictions in importing and exporting things like ideas, text, and algorithms. Notice that Hellman’s PhD was in Electrical Engineering.

Shamir, whose Computer Science PhD was very theoretical, addressed current problems in security in both ways. He predicted that RSA–768 would be factored in 2007, and noted that “Alice and Bob are not Turing machines:” in the real world, real people, with all of their foibles, can have a large influence on the security of any system. Additionally, increased complexity leads to unpredicted aspects of any system, and in the case of security this opens more risks.

Shamir took both sides of “crypto needs to be built in from the beginning,” and repeatedly insisted that he had seen “both ways:” systems with built-in crypto that were “garbage,” and systems with after-the-fact crypto that were secure. This seems very important to me, since Shamir probably gets to see most if not all newly proposed crypto systems.

I think it is noteworthy that this group, consisting of mathematicians, electrical engineers, and computer scientists developed techniques and algorithms that have become pervasive in our society and as important to its infrastructure as bridges and roads.
One thing that was mentioned in class by Richard: there is now a 16-qubit quantum computer on the market; once these become widespread, everything that we know will turn out to be useless at best and wrong at worst.

---

**Key Exchange**

We have talked a little about key exchange, and mentioned several prominent techniques. One way to avoid the whole issue of an eavesdropper is to personally deliver a hardware key, but imagine the havoc this would wreak at an airline’s reservation site: these rightly confidential transactions would be delayed while waiting for a courier from Atlanta to deliver the key so you would be able to enter your credit card number.

But this was a common form of key exchange in World War II. There would be monthly codebooks, and coding hardware on each aircraft or ship. An aircraft approaching an airport would calculate the day’s key (password) using some piece of equipment, and transmit it as a request/warning to the arrival airport. If the proper code is received, the aircraft is allowed to land, and if not fighters are dispatched to shoot it down. These devices were cumbersome and inaccurate, and there are many stories of friendly aircraft who were unable to convince the destination authorities of their legitimacy (one such story is in E. K. Gann’s *Fate is the Hunter*, in which a DC–3 approaching Reykjavik was unable to determine the proper code but snuck in by staying under a very low ceiling. Here are some excerpts:

“Long before approaching Iceland we must be very careful to identify ourselves properly by radio. Should we fail in this, such radio aids as Reykjavic offered would be shut down, which would not make the place any easier to find … Summers [the radio operator] had been presented with a special code kit to make friendly signals in the air, and I asked him now if he understood its complications.

“‘Well … I guess so.’”

After flying over featureless ocean for hours, the plane should have been near Reykjavic. “Johnson had been playing with the direction finder almost continuously, though without result. We should long ago have picked up some kind of signal from Reykjavic, yet the only sound of any consequence had been some rather primitive music from a station in Morocco.

“I asked Summers if he was certain he had sent out the properly coded identification signals. He held a metal folder toward me. Its interior contained miniature tracks upon which there were small sliding tabs. These could be adjusted to a previously agreed-on formula according to the date and hour.

“‘Figure it out for yourself. I set it up just like they told me.’

“I declined. My brain was like a limp salad and held enough confusion without the seasoning of an intricate code machine.”

After landing successfully, “I learned later that there had been a mix-up, not uncommon to partner nations at war. We had been given the wrong code for the day and had been
reported as “approaching enemy aircraft.” The fighter planes had been out looking for us but had never thought we should be skimming the sea.”

This type of key distribution is more suited to a fixed duplex communications channel; that is, one where there exists a physical connection between two nodes, with no other nodes on the “network.” Perhaps the White House – Kremlin “hot line” is the best example.

Other ideas include sending the new key, encrypted using the existing key. The problem here is obvious: if the existing key has been broken, then the eavesdropper will be able to get the new key. One solution, which was part of Zimmerman’s PGP system, was to use slow public key crypto to exchange a private key.

The big need is for one time session keys, what you need to feel secure sending our credit card number to delta.com. The picture is complicated: we need a lot of keys, and 8 bytes of ASCII is nowhere near enough; this ruins the idea of a telephone key exchange, which has its own risk of eavesdroppers; when Newt Gingrich was Speaker of the House, some of his cell phone conversations discussing Republican strategy were intercepted using an ordinary scanner. While this is harder with CDMA and TDMA phones, a determined eavesdropper can still pull your data out of the æther.

---

Key Authorities

One solution to the key distribution problem is to use a central key distribution center (KDC) as described in Stallings. Each subscriber has his or her own secret key, delivered by courier or post; these do not change often.

For a first try, suppose that Alice wants to send a secure message to Bob. Alice’s secret key is $K_A$, while Bob’s is $K_B$. Alice begins with a request for a key from KDC, which generates $K_S$, the session key, and sends it to Alice encrypted with $K_A$; it also sends $K_S$ to Alice encrypted with $K_B$, Bob’s key. Alice sends the $K_B$-encrypted $K_S$ to Bob, who decrypts it using $K_B$. Now both have $K_S$, and can use it.

The problem here is that there are ways for an eavesdropper to intercept or alter $K_S$ to advantage. For example, if Eve the eavesdropper is between Alice and the KDC, she can intercept a session key. The next time Alice wants to communicate with Bob, Eve can replay the same session key, that is, trick Alice and Bob into using the same session key again. This weakens the session key, because Eve now has access to more material that has been encrypted with this key.

To prevent replay attacks, Alice and the KDC need to include and identifier, called a nonce, into the original request. Each request for a session key should use a different nonce, ideally a random number. The KDC then encrypts the nonce with Alice’s key and sends it back to her.

If Eve is between Alice and Bob, she can still try a replay attack. Thus, Bob should encode a nonce using $K_S$ and send it back oto Alice, who alters the nonce in a predictable way and sends it back. Eve could send the nonce back, so it is important that Alice change it.
Diffie–Hellman Key Exchange

The Diffie–Hellman Key Exchange, also called the Diffie-Hellman-Merkle key exchange, was the first asymmetric application in cryptography. It allows two users to generate a secret key in a distributed manner.

The starting point is a (large) prime number $q$ and a number $\alpha$ that is primitive mod $q$. This means that the sequence $\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{q-1}$ has no repeats. These numbers can be public.

Alice selects a private number $X_A$, where $X_A < q$, and similarly Bob selects $X_B$. These are secret. Each raises $\alpha$ to his or her key’s power: Alice calculates $Y_A = \alpha^{X_A}$, and Bob calculates $Y_B = \alpha^{X_B}$.

Notice that the ordinary laws of exponents imply that

$$Y_A^{X_B} = (\alpha^{X_A})^{X_B} = \alpha^{X_A X_B} = \alpha^{X_B X_A} = (\alpha^{X_B})^{X_A} = Y_B^{X_A}.$$ 

Bob, who knows $Y_A$ and $X_B$, can calculate this number, as can Alice, who knows $Y_B$ and $X_A$. This is the secret key.

An opponent knows the two values of $Y$, $q$, and $\alpha$, and knows that $Y_A$ (say) is $\alpha^{X_A}$. But raising to a power is easy, while taking logarithms (in this case, the process is called a discrete logarithm) is hard. Thus, the key is generated using a one-way function.