Securing the Lock after the Key is Stolen

TAL MALKIN

Associate Professor of Computer Science

From online transactions and ATM machines to databases and voting, cryptography lets us share critical information while keeping it safe. Yet cryptographic systems have a weakness. They rely on keys to code and decode messages, and keys can be cracked or stolen.

“Traditional cryptography depends on the assumption that an attacker has no access to secret keys,” Tal Malkin said. “Yet sometimes an attacker can hack into a computer or tamper with your hardware. Part of my work is to maintain security even against such adversaries.”

She envisions systems that respond when attacked. “We can build systems where the key evolves to protect against an adversary who reads or changes part of the key. Even if an adversary reads the entire key, we can protect future transactions,” she said.

Another assumption underlying cryptographic systems is that there is some hard problem that no attacker can solve. For example, the public key software used for secure Internet transactions often relies on the assumption that it is hard to factor the product of two very large prime numbers.

“No one can prove the factoring problem is hard to solve,” said Malkin. “We assume it is because people have worked on this problem for decades. They have developed sophisticated techniques that are much better than the more obvious approaches, but even those procedures require as many operations as the number of atoms in the universe. But if someone does find an efficient solution, it would break all encryption on the Internet.”

As part of her research, Malkin also studies the mathematical foundations of cryptography, searching for the minimal assumptions needed to guarantee security. This starts with studying primitives, such as one-way functions, that act as cryptographic building blocks. “Primitives are small, simple to describe, easy to compute, and hard to crack,” Malkin said. “Cryptographers can combine small primitives to form complex, multilayered security systems.”

Malkin has also focused on general systems for secure computation among two or more parties, as well as optimizing their performance for specific purposes. One example is the no-fly list. The government wants to keep it secret, while airlines want to protect passenger privacy. Malkin has developed a fast way to exchange critical information without showing compromising data. Other applications of secure computation include online voting, sharing national intelligence, and bidding on projects.

In today’s increasingly interconnected world, Malkin’s work on provably secure cryptographic protocols could help protect some very important secrets.

B.S., Bar-Ilan University (Israel), 1993; M.S., Weizmann Institute of Science (Israel), 1995; Ph.D., Massachusetts Institute of Technology, 2000