Authenticating airline passengers

Jason Lee Rogers

By applying the minrank problem to construct a zero-knowledge protocol

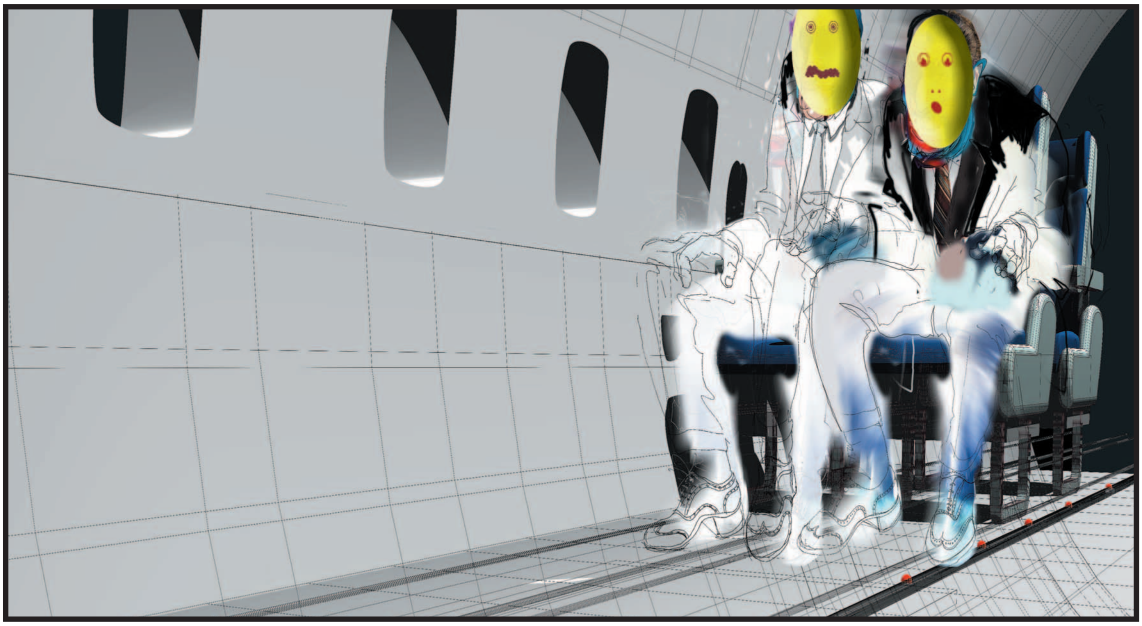
The events of 11 September 2001 sent a ripple of fear across the United States. The US government sought to alleviate the increased concerns by enforcing strict security in airports, government buildings and athletic stadiums; however, with any security there is a price. For airline passengers that price was long lines caused by multiple security checks placed throughout the nation's airport terminals. At each stop, passengers were required to present their plane tickets and personal identification and to allow their bags, garments and even shoes to be checked. Even so, reports filled the papers of what was able to get aboard some airplanes. Also, security was infringing upon privacy and coming up with very few results. A secure method for authenticating airline passengers while allowing a certain level of privacy would need to be found.

Zero-knowledge

To provide security while maintaining privacy is the primary goal of Zero-knowledge. Zero-knowledge, as its name suggests, is an area of mathematics and computer science where the existence of a solution to a problem can be *proved* without giving away the solution.

So if one posed the question, "Is this person a valid passenger on this plane?" The answer could be found, in a manner of speaking, without having to reveal the person's name, social security number, address, or any other piece of information that citizens would rather keep secret. In this environment, the user becomes the Prover who will attempt to *prove* to the airline, or Verifier, that he or she is a valid passenger.

The goal is to create an application that could be run from any airport terminal to quickly and securely verify a passenger while preserving the passenger's privacy. The Zero-knowledge authentication protocol will rely on a category of mathematics problems called NP-Hard, where NP means non-probabilistic.



©DIGITALVISION-DANIEL MACKIE, COMPOSITE: MKC

NP-hard is a classification of problems that, at this time, cannot be solved in polynomial time. This unique nature makes them a desirable choice when constructing a Zero-knowledge situation. The specific NP-Hard problem used in this implementation is called the MinRank Problem.

The MinRank Problem touches upon an area of linear algebra where certain unknown variables of a Matrix are chosen to give the Matrix a desired rank. For its portability and flexibility, the JAVA programming language has been chosen for implementing the protocol. This MinRank Authentication Scheme should not be considered a cure-all solution for airline security. But, rather, it is a small step towards securing the world's airports.

The ground zero goal

The goal of a Zero-knowledge protocol is to provide an algorithm that is capable of proving "a statement without yielding anything beyond its validity," writes Oded Goldreich in *Foundations of Cryptography*. In other words, a Prover could convince a Verifier of a solution to a publicly known problem without ever revealing the solution.

Implementation of a Zero-knowledge requires interaction between the Prover and Verifier. After the initial setup of defining the problem and deriving a solution by the Prover, the interaction occurs in two distinct phases. The *commit* phase, which begins the process, forces the Prover to commit to a unique value. This phase is constructed to yield no knowledge to the Verifier. The interaction between Prover and Verifier finishes with the *reveal* phase where the Prover reveals a response to the Verifier that should coincide with the original unique value to which the Prover was committed. The Verifier will validate this assertion. If so, the protocol is considered *viable* according to Goldreich.

Two properties central to the development of Zero-knowledge are soundness and completeness. The *soundness* property relates to the assertion that the Verifier cannot be "tricked" into accepting false statements from the Prover. *Completeness*, most appropriately, refers to the Prover's capacity to convince the Verifier of true statements, says Goldreich. The probabilities of soundness and completeness determine a protocol's strength.

For a typical protocol the probability for each to occur is about fifty percent. Therefore, the Prover can trick the Verifier one out of every two interactions. Such a revelation would appear to offset any security benefits. However, "if one executes one Zero-knowledge proof after another, then the composed execution must be Zero-knowledge," according to Goldreich.

This allows the Prover to develop *accreditation*, or "the building of confidence through each iteration of the protocol," points out Professor Aronsson, at Helsinki University of Technology (Finland). So if the initial probability is 50 percent, the probability of tricking the Verifier is two-fifths of a percent, or 1 out of 256, after ten consecutive interactions. This assumes that any false interaction will invalidate the entire interaction.

Zero-knowledge seeks to help validate solutions of publicly known problems. Therefore, it is necessary to investigate certain sets of problems that might be used to construct a sufficient protocol.

NP

NP is a class of problems that, by their nature, proceed nicely into Zero-knowledge protocols. Goldreich demonstrates in *Foundations of Cryptography* "a method for constructing Zero-knowledge proofs for every language in NP." So if one considers a problem within NP, a Zero-knowledge protocol implementing the problem is not far off.

Foremost, a NP problem must be a decision problem. A *decision* problem is any problem whose input requires a yes or no answer. Furthermore, it must be a decision problem of a certain complexity. Complexity is determined by the number of steps that would be required to find or verify a solution.

All NP problems can be verified in polynomial time, or n^k where n is problem specific and k is some constant. The question is whether or not a solution can be discovered in polynomial-time for a given problem. Or, rather, whether or not P=NP where P is the set of all problems that can be solved in polynomial time.

This unique characteristic is what gives NP problems the ability to construct a Zero-knowledge protocol. Within NP are those problems considered NP-Complete. NP-Complete problems are the most difficult of the NP problems. Any problems not considered NP-Complete are just NP-Hard. From this assertion, one looks to linear algebra and the operation of constructing matrices of desired rank.

The MinRank problem

The MinRank problem concerns itself with creating a matrix of minimum rank. The rank of a matrix is the maximum number of linearly independent rows. The MinRank problem is a problem of finding a linear combination of some given matrices that has a small rank according to Nicolas Curtois. That is mathematically speaking,

Let E, S be subsets of a commutative ring R.

Let $x_1, x_2, \ldots, x_t$ be variables.

Given M – M(x_1, x_2, …, x_t) with entries chosen from

the union of E and { $x_1, x_2, \ldots, x_t$ }.

Then,

MinRank$_S$(M) = min{ rank M(_$_1$, _$_2$, …, _$_t$) where (_$_1$, _$_2$, …, _$_t$) are elements of S^t (Buss, *et al*).

Since solving multivariate quadratic equations over a field is NP-Hard, which is what MinRank attempts to do, MinRank is NP-Hard. Buss, Frandsen, and Jeffrey extend MinRank to NP-Complete when S is a Gaussian Field. A Gaussian Field, GF(q), is simply a set of integers from 1 to q where q is a prime number. Since the MinRank Problem is NP-Complete over GF(q), a Zero-knowledge protocol can be constructed.

MinRank authentication system

The MinRank Authentication System seeks to apply the MinRank problem to construct a Zero-knowledge Authentication protocol. (Note: This protocol is adapted from Nicolas Curtois, "Efficient Zero-knowledge Authentication Based on a Linear Algebra Problem MinRank," CP8 Crypto Team, SchlumbergerSema. For more information, visit <http://www.MinRank.org.>.)

Before any interaction between the Prover and the Verifier takes place, the public and private keys must be created. Since finding a solution to the MinRank problem is NP-Complete, the problem must be created from a possible solution.

To begin, a Gaussian Field, $GF(q)$, is created over q, a prime integer. Using this field, m random n x n matrices, M_0; M_1, …, M_{m-1}, are created. Next, a m-tuple, _, is created from $GF(q)^m$. This m-tuple becomes the private key or solution to the MinRank problem. The final step is to create the final matrix M_m, such that _ _$_i M_i$ for all $i = m$ is equal to some rank, r. To reach this conclusion a matrix, M, of rank r is used to compute M_m as follows,

M_m = (M + M_0 - _ _$_i M_i$)/_$_m$
for all $i<m$

Now the linear combination of all m matrices will have rank r as desired. From this setup of the public and private keys, subsequent interactions can occur.

The interaction between the Prover and the Verifier must be developed carefully in order to preserve the principles of Zero-knowledge, which in this case is the private m-tuple. The Prover begins by creating three random n x n matrices S and T, which are invertible, and X. Choosing a random _$_1$ from $GF(q)^m$, the Prover creates,

N_1 = _ _$_{1i} M_i$ for all $i = m$

and,

N_2 = _ _$_{2i} M_i$ for all $i = m$

where _$_2$ - _$_1$ = _.

Next, the Prover selects a collision-intractable one-way hash function, H. A hash function is simply an algorithm that takes each matrix and sends it to a unique value. As defined by Phillip Rogaway of University of California-Davis, for a hash function, H, to be collision-intractable it must not be possible to find M, M' such that H(M) = H(M'). Also, for the hash function to be one-way there must be no way to find M given H(M). One round of the Prover-Verifier interaction goes as follows:

1. The Prover sends to the Verifier:

$H(S, T, X)$, $H(TN_1S + X)$, $H(TN_2S + X - TM_0S)$

This would be considered the commitment phase of the interaction.

2. The Verifier chooses a query Q of {0,1,2} and sends Q to the Prover:

3. If Q=0 the Prover gives the following values:

$(TN_1S + X)$, $(TN_2S + X - TM_0S)$

Verification Q=0: The Verifier

accepts if

$H(TN_1S + X)$ and $H(TN_2S + X – TM_0S)$ are correct and if

$(TN_2S + X – TM_0S)$ - $(TN_1S + X)$ = TMS is indeed a matrix of rank *r*.

3'. If *Q* = 1, 2 the Prover reveals: *S, T, X* and $\_{}_Q$

Verification Q = 1, 2: The Verifier checks if *S* and *T* are invertible and that $H(S, T, X)$ is correct. The he computes

TN_QS

and verifies $H(TN_1S + X)$ *or* $H(TN_2S + X – TM_0S)$.

For a legitimate Prover, _ always succeeds in making this protocol complete. By this proof, states Curtois, the algorithm has one-third soundness. To achieve high probability, many rounds of interaction can be performed, which preserves the Zero-knowledge.

Privacy advantage

The advantage of using Zero-knowledge authentication is the privacy it ensures. In most modern systems the user is asked to give up some information to the system in order to authenticate, such as credit card numbers, mother's maiden name, or fingerprints. Once the system has the information users must trust that the system will protect it. With Zero-knowledge authentication the information exchanged doesn't have to be protected.

Computing disadvantage

MinRank Authentication relies heavily on the strength of the hash algorithm used. So any attack on the hash algorithm is a possible attack on MinRank. A temporal solution would be to choose a stronger algorithm if such an attack is found. But, the strength of the algorithm is still limited by the processing power of the smart card.

The application

The purpose of any sound theory is to move toward a practical application. The application here has been developed using JAVA. JAVA was chosen for its flexibility and its portability. The documentation and the source code for the application can be found at <http://csc.noctrl.edu/f/kwt>.

The main class is called MinRankAuthentication and it calls the MinRank class, the true engine of the protocol. Within the MinRank class are two separate classes, the Prover and the Verifier, that act as the separate parties within the interaction.

Both classes also utilize a *GFqMatrix* class that provides matrix creation and computation over a Gaussian Field *GF(q)*. This class was derived from the matrix class within the JAMA package. (This package was not included within the standard JAVA release.) All classes have been wrapped within a MinRank package.

SHA-512, as defined in Secure Hash Standard, NIST FIPS 180-2, has been selected as the hash algorithm. Any other hash algorithm that preserves the properties of being collision-intractable and one-way could be substituted.

Implementation

Though the package is not implementable by itself, a possible implementation could proceed as follows. Passengers would apply for their Travel Card, the smartcard used for authentication, from a secure authority either provided by the government for use on all airlines or by the airlines themselves. Once the passenger had identified themselves with the secure authority a private key would be stored upon the travel card, while the public key would be bound to the passenger. When booking a flight, an airline would request the passengers public from the secure authority. Upon arriving at the airport the passenger would proceed to the Ticketing Terminal where they would use their Travel Card to authenticate themselves and receive another private key corresponding to their reservation on the plane. Since MinRank allows multiple private keys for any given public key, only one public key would need to be stored for any flight. Before boarding the airline, the passenger would use their Travel Card to authenticate both themselves and their reservation on the flight. Any implementation would still require secure methods for transferring and applying public and private keys.

Conclusion

The MinRank Authentication System attempts to provide a partial solution toward securing airports. Implementable in a smart card environment, airlines could quickly but securely validate passengers boarding a plane. Security measures, such as random bag checks and shoe searches, have recently discouraged many citizens from flying. The privacy the MinRank Authentication System could insure would be comforting, attracting business back to the world's struggling airlines.

No one wants to board an airplane afraid of fellow travelers. As the repercussions of 11 September continue to play out, citizens will notice the restrictions caused by new policy and procedure. Easing those restrictions, through methods that preserve privacy, will be instrumental.

Read more about it

- Aronsson, Hannu. “Zero Knowledge Protocols and Small Systems.” Department of Computer Science, Helsinki University of Technology. 13 Feb. 2002.
- Buss, Jonathan, Gudmund Frandsen, and Jeffrey Shallit. “The Complexity of Some Problems of Linear Algebra.” 30 Sept. 1996.
- “Complexity classes P and NO.” www.wikipedia.com. 12 Feb. 2002.
- Cormen, Thomas, *Introduction to Algorithms*, Boston: MIT Press, 2001.
- Curtois, Nicolas, “Efficient Zero-knowledge Authentication Based on a Linear Algebra Problem MinRank.” CP8 Crypto Team, SchlumbergerSema. 1999.
- Goldreich, Oded, *Foundations of Cryptography*, 1995.
- Horstmann, Cay and Gary Cornell, JAVA 2: *Volume 1-Fundamentals*, Palo Alto, CA: Sun Microsystems, 2001.
- Rogaway, Phillip. “Reducibility and the Modeling of Block Ciphers.” ECS 227: Modern Cryptography. University of California. Davis, 16 Jan. 1996
- “Secure Hash Standard”. Federal Information Processing Standards Publication 180-2. 7 May 2001.

About the author

Jason Lee Rogers obtained his B.S. degree from North Central College in Naperville, IL. He is currently a graduate student at the Naval Postgraduate School under the Scholarship for Service program funded by the National Science Foundation. His current research interests are in formal method analysis and computer security.