# CPSC 467: Cryptography and Computer Security

Michael J. Fischer

Lecture 18
November 6, 2017

## Authentication While Preventing Impersonation

Challenge-response authentication protocols

Authentication using zero knowledge interactive proofs

## Quadratic Residues, Squares, and Square Roots

Modular square roots

Square roots modulo *n*

Square roots modulo an odd prime *p*

Square roots modulo the product of two odd primes

## Zero Knowledge Protocols

Feige-Fiat-Shamir Authentication Protocol

Secret cave protocol

# Authentication While Preventing Impersonation

# Preventing impersonation

A fundamental problem with all of the password authentication schemes discussed so far is that Alice reveals her secret to Bob every time she authenticates herself.

This is fine when Alice trusts Bob but not otherwise.

After authenticating herself once to Bob, then Bob can masquerade as Alice and impersonate her to others.

## Authentication requirement

When neither Alice nor Bob trust each other, there are two requirements that must be met:

1. Bob wants to make sure that an impostor cannot successfully masquerade as Alice.
2. Alice wants to make sure that her secret remains secure.

At first sight these seem contradictory, but there are ways for Alice to prove her identity to Bob without compromising her secret.

# Challenge-Response Authentication Protocols

# Challenge-response authentication protocols

In a challenge-response protocol, Bob presents Alice with a challenge that only the true Alice (or someone knowing Alice's secret) can answer.

Alice answers the challenge and sends her answer to Bob, who verifies that it is correct.

Bob learns the response to his challenge but Alice never reveals her secret.

If the protocol is properly designed, it will be hard for Bob to determine Alice's secret, even if he chooses the challenges with that end in mind.

# Challenge-response protocol from a signature scheme

A challenge-response protocol can be built from a digital signature scheme $(S_A, V_A)$.

(The same protocol can also be implemented using a symmetric cryptosystem with shared key $k$.)

| | Alice | | Bob |
|---|---|---|---|
| 1. | | $\xleftarrow{r}$ | Choose random string $r$. |
| 2. | Compute $s = S_A(r)$ | $\xrightarrow{s}$ | Check $V_A(r, s)$. |

# Requirements on underlying signature scheme

This protocol exposes Alice's signature scheme to a chosen plaintext attack.

A malicious Bob can get Alice to sign any message of his choosing.

Alice had better have a different signing key for use with this protocol than she uses to sign contracts.

While we hope our cryptosystems are resistant to chosen plaintext attacks, such attacks are very powerful and are not easy to defend against.

Anything we can do to limit exposure to such attacks can only improve the security of the system.

# Limiting exposure to chosen plaintext attack: try 1

We explore some ways that Alice might limit Bob's ability to carry out a chosen plaintext attack.

Instead of letting Bob choose the string $r$ for Alice to sign, $r$ is constructed from two parts, $r_1$ and $r_2$.

$r_1$ is chosen by Alice; $r_2$ is chosen by Bob. Alice chooses first.

| | Alice | | Bob |
|---|---|---|---|
| 1. | Choose random string $r_1$ | $\xrightarrow{r_1}$ | |
| 2. | | $\xleftarrow{r_2}$ | Choose random string $r_2$. |
| 3. | Compute $r = r_1 \oplus r_2$ | | Compute $r = r_1 \oplus r_2$ |
| 4. | Compute $s = S_A(r)$ | $\xrightarrow{s}$ | Check $V_A(r, s)$. |

# Problem with try 1

The idea is that neither party should be able to control $r$.

Unfortunately, that idea does not work here because Bob gets $r_1$ before choosing $r_2$.

Instead of choosing $r_2$ randomly, a cheating Bob can choose $r_2 = r \oplus r_1$, where $r$ is the string that he wants Alice to sign.

Thus, try 1 is no more secure against chosen plaintext attack than the original protocol.

# Limiting exposure to chosen plaintext attack: try 2

Another possibility is to choose the random strings in the other order—Bob chooses first.

| | Alice | | Bob |
|---|---|---|---|
| 1. | | $\xleftarrow{r_2}$ | Choose random string $r_2$. |
| 2. | Choose random string $r_1$ | $\xrightarrow{r_1}$ | |
| 3. | Compute $r = r_1 \oplus r_2$ | | Compute $r = r_1 \oplus r_2$ |
| 4. | Compute $s = S_A(r)$ | $\xrightarrow{s}$ | Check $V_A(r, s)$. |

# Try 2 stops chosen plaintext attack

Now Alice has complete control over $r$.

No matter how Bob chooses $r_2$, Alice's choice of a random string $r_1$ ensures that $r$ is also random.

This thwarts Bob's chosen plaintext attack since $r$ is completely random.

Thus, Alice only signs random messages.

# Problem with try 2

Unfortunately, try 2 is totally insecure against active eavesdroppers.
Why?

Suppose Mallory listens to a legitimate execution of the protocol between Alice and Bob.

From this, he easily acquires a valid signed message $(r_0, s_0)$.
How does this help Mallory?

Mallory sends $r_1 = r_0 \oplus r_2$ in step 2 and $s = s_0$ in step 4.

Bob computes $r = r_1 \oplus r_2 = r_0$ in step 3, so his verification in step 4 succeeds.

Thus, Mallory can successfully impersonate Alice to Bob.

# Further improvements

Possible improvements to both protocols.

1. Let $r = r_1 \cdot r_2$ (concatenation).
2. Let $r = h(r_1 \cdot r_2)$, where $h$ is a cryptographic hash function.

In both cases, neither party now has full control over $r$.

This weakens Bob's ability to launch a chosen plaintext attack if Alice chooses first.

This weakens Mallory's ability to impersonate Alice if Bob chooses first.

# Concept of zero knowledge

In all of the challenge-response protocols above, Alice releases some partial information about her secret by producing signatures that Bob could not compute by himself.

*Zero knowledge* protocols allows Alice to prove knowledge of her secret without revealing any information about the secret itself.

# Authentication using zero knowledge

Alice authenticates herself by successfully completing several rounds of a protocol that requires knowledge of a secret $s$.

In a single round of the protocol, Bob has at least a 50% chance of catching an impostor Mallory.

By repeating the protocol $t$ times, the error probability (that is, the probability that Bob fails to catch Mallory) drops to $1/2^t$.

This can be made acceptably low by choosing $t$ to be large enough.

For example, if $t = 20$, then Mallory has only one chance in a million of successfully impersonating Alice.

# Feige-Fiat-Shamir authentication protocol

The Feige-Fiat-Shamir authentication protocol is a zero knowledge protocol based on the difficulty of computing square roots modulo composite numbers.

We will present it later in some detail.

But first, we need to look more closely at squares and square roots in $\mathbf{Z}_n$.

# Quadratic Residues, Squares, and Square Roots

# One-way functions

Cryptography is built on the notion of *one-way function*, that is, a function that is easy to compute but hard to invert.

Can't prove that inversion is hard.

Instead, postulate it to be hard for particular well-studied functions that have no known feasible inversion algorithms.

Some presumed one-way functions and associated hard problems:

| | |
|---|---|
| $(p, q) \mapsto p \cdot q$ | Factoring problem |
| $x \mapsto g^x \bmod p$ | Discrete log problem |
| $P \mapsto k \times P$ | Ellpitic curve discrete log problem |
| $x \mapsto H(x)$ | Collision-finding problem |
| $x \mapsto x^2 \bmod n$ | Quadratic residuosity problem |

# The squaring function modulo $n$

Today we look at the squaring function $x \mapsto x^2 \bmod n$ and its inverse function, $y \mapsto \sqrt{x} \bmod n$.

We've already seen square roots modulo a prime $p$ in lecture 13, where finding points on an elliptic curve requires solving the equation

$$y^2 \equiv x^3 + ax + b \pmod{p}$$

for $y$, and that in turn requires computing square roots in $\mathbf{Z}_p^*$.

# Perfect squares over integers and $\mathbf{Z}_n^*$

Squares and square roots have several other cryptographic applications as well.

A *quadratic residue* is the analog in $\mathbf{Z}_n^*$ of a perfect square over the integers.

In both cases, it is an element for which one or more square roots exist.

For example, 4 is a perfect square over the integers.
It has two square roots, 2 and $-2$.

# Squares and square roots

An integer $b$ is a *square root* of $a$ modulo $n$ if

$$b^2 \equiv a \pmod{n}.$$

An integer $a$ is a *quadratic residue (or perfect square)* modulo $n$ if it has a square root modulo $n$.

# Quadratic residues in $\mathbf{Z}_n^*$

If $a, b \in \mathbf{Z}_n$ and $b^2 \equiv a \pmod{n}$, then

$$b \in \mathbf{Z}_n^* \text{ iff } a \in \mathbf{Z}_n^*.$$

Why? Because

$$\gcd(b, n) = 1 \text{ iff } \gcd(a, n) = 1$$

This follows from the fact that $b^2 = a + un$ for some $u$, so if $p$ is a prime divisor of $n$, then

$$p \mid b \text{ iff } p \mid a.$$

Assume that all quadratic residues and square roots are in $\mathbf{Z}_n^*$ unless stated otherwise.

# $\mathrm{QR}_n$ and $\mathrm{QNR}_n$

We partition $\mathbf{Z}_n^*$ into two parts.

$$\mathrm{QR}_n = \{a \in \mathbf{Z}_n^* \mid a \text{ is a quadratic residue modulo } n\}.$$
$$\mathrm{QNR}_n = \mathbf{Z}_n^* - \mathrm{QR}_n.$$

$\mathrm{QR}_n$ is the *set of quadratic residues* modulo $n$.

$\mathrm{QNR}_n$ is the *set of quadratic non-residues* modulo $n$.

For $a \in \mathrm{QR}_n$, we sometimes write

$$\sqrt{a} = \{b \in \mathbf{Z}_n^* \mid b^2 \equiv a \pmod{n}\},$$

the *set of square roots* of $a$ modulo $n$.

# Quadratic residues in $\mathbf{Z}_{15}^*$

The following table shows all elements of $\mathbf{Z}_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$ and their squares.

| $b$ | | $b^2 \bmod 15$ |
|---|---|---|
| 1 | | 1 |
| 2 | | 4 |
| 4 | | 1 |
| 7 | | 4 |
| 8 | $= -7$ | 4 |
| 11 | $= -4$ | 1 |
| 13 | $= -2$ | 4 |
| 14 | $= -1$ | 1 |

Thus, $\mathrm{QR}_{15} = \{1, 4\}$ and $\mathrm{QNR}_{15} = \{2, 7, 8, 11, 13, 14\}$.

# Quadratic residues modulo an odd prime $p$

## Fact

*For an odd prime p,*

- *Every $a \in QR_p$ has exactly two square roots in $\mathbf{Z}_p^*$;*
- *Exactly 1/2 of the elements of $\mathbf{Z}_p^*$ are quadratic residues.*

In other words, if $a \in \mathrm{QR}_p$,

$$|\sqrt{a}| = 2.$$

$$|\mathrm{QR}_n| = \frac{|\mathbf{Z}_p^*|}{2} = \frac{p-1}{2}.$$

# Quadratic residues in $\mathbf{Z}_{11}^*$

The following table shows all elements $b \in \mathbf{Z}_{11}^*$ and their squares.

| $b$ | $b^2 \bmod 11$ |
|---|---|
| 1 | 1 |
| 2 | 4 |
| 3 | 9 |
| 4 | 5 |
| 5 | 3 |

| $b$ | $-b$ | $b^2 \bmod 11$ |
|---|---|---|
| 6 | $-5$ | 3 |
| 7 | $-4$ | 5 |
| 8 | $-3$ | 9 |
| 9 | $-2$ | 4 |
| 10 | $-1$ | 1 |

Thus, $\mathrm{QR}_{11} = \{1, 3, 4, 5, 9\}$ and $\mathrm{QNR}_{11} = \{2, 6, 7, 8, 10\}$.

# Proof that $|\sqrt{a}| = 2$ modulo an odd prime $p$

Let $a \in \mathrm{QR}_p$.

- It must have a square root $b \in \mathbf{Z}_p^*$.
- $(-b)^2 \equiv b^2 \equiv a \pmod{p}$, so $-b \in \sqrt{a}$.
- Moreover, $b \not\equiv -b \pmod{p}$ since $p \nmid 2b$, so $|\sqrt{a}| \geq 2$.
- Now suppose $c \in \sqrt{a}$. Then $c^2 \equiv a \equiv b^2 \pmod{p}$.
- Hence, $p \,|\, c^2 - b^2 = (c-b)(c+b)$.
- Since $p$ is prime, then either $p \,|\, (c-b)$ or $p \,|\, (c+b)$ (or both).
- If $p \,|\, (c-b)$, then $c \equiv b \pmod{p}$.
- If $p \,|\, (c+b)$, then $c \equiv -b \pmod{p}$.
- Hence, $c \equiv \pm b \pmod{p}$, so $\sqrt{a} = \{b, -b\}$, and $|\sqrt{a}| = 2$.

# Proof that half the elements of $\mathbf{Z}_p^*$ are in $\mathrm{QR}_p$

- Each $b \in \mathbf{Z}_p^*$ is the square root of exactly one element of $\mathrm{QR}_p$, namely, $b^2 \bmod p$.
- The mapping $b \mapsto b^2 \bmod p$ is a 2-to-1 mapping from $\mathbf{Z}_p^*$ to $\mathrm{QR}_p$.
- Therefore, $|\mathrm{QR}_p| = \frac{1}{2}|\mathbf{Z}_p^*|$ as desired.

# Quadratic residues modulo $pq$

We now turn to the case where $n = pq$ is the product of two distinct odd primes.

## Fact

*Let $n = pq$ for $p$, $q$ distinct odd primes.*

- *Every $a \in QR_n$ has exactly four square roots in $\mathbf{Z}_n^*$;*
- *Exactly 1/4 of the elements of $\mathbf{Z}_n^*$ are quadratic residues.*

In other words, if $a \in \mathrm{QR}_n$ then $|\sqrt{a}| = 4$, so

$$|\mathrm{QR}_n| = \frac{|\mathbf{Z}_n^*|}{4} = \frac{(p-1)(q-1)}{4}.$$

# Proof sketch

- Let $a \in \mathrm{QR}_n$. Then $a \in \mathrm{QR}_p$ and $a \in \mathrm{QR}_q$.
- There are numbers $b_p \in \mathrm{QR}_p$ and $b_q \in \mathrm{QR}_q$ such that
  - $\sqrt{a} \pmod{p} = \{\pm b_p\}$, and
  - $\sqrt{a} \pmod{q} = \{\pm b_q\}$.
- Each pair $(x, y)$ with $x \in \{\pm b_p\}$ and $y \in \{\pm b_q\}$ can be combined to yield a distinct element $b_{x,y}$ in $\sqrt{a} \pmod{n}$.[1]
- Hence, $|\sqrt{a}| = 4$, and $|\mathrm{QR}_n| = \frac{1}{4}|\mathbf{Z}_n^*|$.

[1] To find $b_{x,y}$ from $x$ and $y$ requires use of the Chinese Remainder theorem (see Appendix).

# Zero Knowledge Protocols

# Zero knowledge interactive proofs (ZKIP)

A protocol where Bob provably learns nothing about Alice’s secret is called a *zero-knowledge interactive proof*.

Here, “learns” means computational knowledge: Anything that Bob could have computed with the help of Alice he could have computed by himself without Alice’s help.

We start with a simplified version of the Feige-Fiat-Shamir authentication protocol.

# Feige-Fiat-Shamir protocol: preparation

The Feige-Fiat-Shamir protocol is based on the difficulty of computing square roots modulo composite numbers.

- Alice chooses $n = pq$, where $p$ and $q$ are distinct large primes.
- Next she picks a quadratic residue $v \in \mathrm{QR}_n$ (which she can easily do by choosing a random element $u \in \mathbf{Z}_n^*$ and letting $v = u^2 \bmod n$).
- Finally, she chooses $s$ to be the smallest square root of $v^{-1} \pmod{n}$.[2] She can do this since she knows the factorization of $n$.

She makes $n$ and $v$ public and keeps $s$ private.

[2] Note that if $v$ is a quadratic residue, then so is $v^{-1} \pmod{n}$.

# A simplified one-round FFS protocol

Here's a simplified one-round version.

| | Alice | | Bob |
|---|---|---|---|
| 1. | Choose random $r \in \mathbf{Z}_n^*$. | | |
| | Compute $x = r^2 \bmod n$. | $\xrightarrow{x}$ | |
| 2. | | $\xleftarrow{b}$ | Choose random $b \in \{0, 1\}$. |
| 3. | Compute $y = rs^b \bmod n$. | $\xrightarrow{y}$ | If $b = 0$, check $x = y^2 \bmod n$. |
| | | | If $b = 1$, check $x = y^2 v \bmod n$. |

When both parties are honest, Bob accepts Alice because

$$x = y^2 v^b \bmod n.$$

This holds because

$$y^2 v^b \equiv (rs^b)^2 v^b \equiv r^2 (s^2 v)^b \equiv x(v^{-1} v)^b \equiv x \pmod{n}.$$

# The Secret Cave Protocol

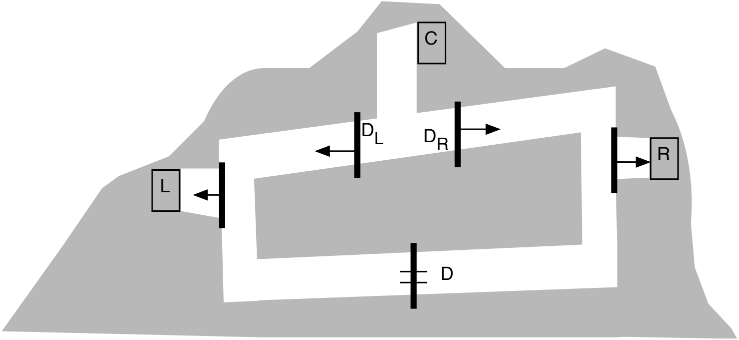
# Zero knowledge without number theory

While it might seem that zero knowledge interactive proofs are intimately tied up with number theory, the Secret Cave Protocol is a purely physical illustration of zero knowledge, devoid of mathematics or number theory.

# The secret cave protocol

The secret cave protocol illustrates the fundamental ideas behind zero knowledge without any reference to number theory or hardness of computation.

Image a cave with tunnels and doors as shown below.



# Secret cave protocol (cont.)

There are three openings to the cave: $L$, $C$, and $R$.

$L$ and $R$ are blocked by exit doors, like at a movie theater, which can be opened from the inside but are locked from the outside. The only way into the cave is through passage $C$.

The cave itself consists of a U-shaped tunnel that runs between $L$ and $R$. There is a locked door $D$ in the middle of this tunnel, dividing it into a left part and a right part.

A short tunnel from $C$ leads to a pair of doors $D_L$ and $D_R$ through which one can enter left and right parts of the cave, respectively. These doors are also one-way doors that allow passage from $C$ into either the left or right parts of the cave, but once one passes through, the door locks behind and one cannot return to $C$.

# Alice's proposition

Alice approaches Bob, tells him that she has a key that opens door $D$, and offers to sell it to him.

Bob would really like such a key, as he often goes into the cave to collect mushrooms and would like easy access to both sides of the cave without having to return to the surface to get into the other side.

However, he doesn't trust Alice that the key really works, and Alice doesn't trust him with her key until she gets paid.

# Their conversation

Bob tells Alice.

> *"Give me the key so I can go down into the cave and try it to make sure that it really works."*

Alice retorts,

> *"I'm not that dumb. If I give you the key and you disappear into the cave, I'll probably never see either you or my key again. Pay me first and then try the key."*

Bob answers,

> *"If I do that, then you'll disappear with my money, and I'm likely to be stuck with a non-working key."*

# How do they resolve their dilemma?

They think about this problem for awhile, and then Alice suggests,

> *"Here's an idea: I'll enter the cave through door $C$, go into the left part of the cave, open $D$ with my key, go through it into the right part of the cave, and then come out door $R$. When you see me come out $R$, you'll know I've succeeded in opening the door."*

Bob thinks about this and then asks,

> *"How do I know you'll go into the left part of the cave? Maybe you'll just go into the right part and come out door $R$ and never go through $D$."*

# Alice's plan

Alice says,

> *"OK. I'll go into either the left or right side of the cave. You'll know I'm there because you'll hear a door clank when it closes behind me. You won't know whether I went through $D_L$ or $D_R$, but that doesn't matter. I'll be stuck in one side of the cave or the other."*
>
> *"You then yell down into the cave which door you want me to come out—L or R—and I'll do so. If I'm on the opposite side from what you request, then I'll have no choice but to unlock $D$ in order to pass through to the other side."*

# Bob's hesitation

Bob is beginning to be satisfied, but he hesitates.

> *"Well, yes, that's true, but if you're lucky and happen to be on the side I call out, then you don't have to use your key at all, and I still won't know that it works."*

Alice answers,

> *"Well, I might be lucky once, but I surely won't be lucky 20 times in a row, so I'll agree to do this 20 times. If I succeed in coming out the side you request all 20 times, do you agree to buy my key?"*

# Agreement finally

Bob agrees, and they spend the rest of the afternoon climbing in and out of the cave and shouting.

# Chinese Remainder Theorem

## Systems of congruence equations

### Theorem (Chinese remainder theorem)

*Let $n_1, n_2, \ldots, n_k$ be positive pairwise relatively-prime integers*[3]*, let $n = \prod_{i=1}^{k} n_i$, and let $a_i \in \mathbf{Z}_{n_i}$ for $i = 1, \ldots, k$. Consider the system of congruence equations with unknown $x$:*

$$\begin{array}{rcl} x & \equiv & a_1 \ (mod\ n_1) \\ x & \equiv & a_2 \ (mod\ n_2) \\ & \vdots & \\ x & \equiv & a_k \ (mod\ n_k) \end{array} \tag{1}$$

*(1) has a unique solution $x \in \mathbf{Z}_n$.*

[3] This means that $\gcd(n_i, n_j) = 1$ for all $1 \leq i < j \leq k$.

## How to solve congruence equations

To solve for $x$, let

$$N_i = n/n_i = \underbrace{n_1 n_2 \dots n_{i-1}}\cdot\underbrace{n_{i+1} \dots n_k},$$

and compute $M_i = N_i^{-1} \bmod n_i$, for $1 \le i \le k$.

$N_i^{-1} \pmod{n_i}$ exists since $\gcd(N_i, n_i) = 1$. (Why?)

We can compute $N_i^{-1}$ by solving the associated Diophantine equation as described in lecture 14.

The solution to (1) is

$$x = \left(\sum_{i=1}^{k} a_i M_i N_i\right) \bmod n \tag{2}$$

# Correctness

### Lemma

$$M_j N_j \equiv \begin{cases} 1 \ (mod\ n_i) & if\ j = i; \\ 0 \ (mod\ n_i) & if\ j \neq i. \end{cases}$$

### Proof.

$M_i N_i \equiv 1 \pmod{n_i}$ since $M_i = N_i^{-1} \bmod n_i$.
If $j \neq i$, then $M_j N_j \equiv 0 \pmod{n_i}$ since $n_i | N_j$. □

It follows from the lemma and the fact that $n_i \mid n$ that

$$x \equiv \sum_{i=1}^{k} a_i M_i N_i \equiv a_i \pmod{n_i} \tag{3}$$

for all $1 \leq i \leq k$, establishing that (2) is a solution of (1).

## Uniqueness

To see that the solution is unique in $\mathbf{Z}_n$, let $\chi : \mathbf{Z}_n \to \mathbf{Z}_{n_1} \times \ldots \times \mathbf{Z}_{n_k}$ be the mapping

$$x \mapsto (x \bmod n_1, \ldots, x \bmod n_k).$$

$\chi$ is a surjection[4] since $\chi(x) = (a_1, \ldots, a_k)$ iff $x$ satisfies (1).

Since also $|\mathbf{Z}_n| = |\mathbf{Z}_{n_1} \times \ldots \times \mathbf{Z}_{n_k}|$, $\chi$ is a bijection, and there is only one solution to (1) in $\mathbf{Z}_n$.

[4] A *surjection* is an onto function.

# An alternative proof of uniqueness

A less slick but more direct way of seeing uniqueness is to suppose that $x = u$ and $x = v$ are both solutions to (1).

Then $u \equiv v \pmod{n_i}$, so $n_i | (u - v)$ for all $i$.

By the pairwise relatively prime condition on the $n_i$, it follows that $n | (u - v)$, so $u \equiv v \pmod{n}$. Hence, the solution is unique in $\mathbf{Z}_n$.