CPSC 467: Cryptography and Computer Security

Michael J. Fischer

Lecture 17
October 29, 2014
Authentication While Preventing Impersonation
  Challenge-response authentication protocols
  Authentication using zero knowledge interactive proofs

Quadratic Residues, Squares, and Square Roots
  Square roots modulo general \( n \)
  Square roots modulo an odd prime \( p \)
  Square roots modulo the product of two odd primes

Chinese Remainder Theorem

Feige-Fiat-Shamir Authentication 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 \(k\).)

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(\leftarrow^r) Choose random string (r).</td>
</tr>
<tr>
<td>2.</td>
<td>Compute (s = S_A(r)) (\rightarrow^s) Check (V_A(r, s)).</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose random string $r_1$</td>
<td>$\rightarrow r_1$</td>
</tr>
<tr>
<td>2.</td>
<td>$\leftarrow r_2$ Choose random string $r_2$.</td>
</tr>
<tr>
<td>3. Compute $r = r_1 \oplus r_2$</td>
<td>Compute $r = r_1 \oplus r_2$</td>
</tr>
<tr>
<td>4. Compute $s = S_A(r)$</td>
<td>$\rightarrow s$ Check $V_A(r, s)$.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Choose random string $r_2$.</td>
</tr>
<tr>
<td>2. Choose random string $r_1$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>3. Compute $r = r_1 \oplus r_2$</td>
<td>Compute $r = r_1 \oplus r_2$</td>
</tr>
<tr>
<td>4. Compute $s = S_A(r)$</td>
<td>$s$ Check $V_A(r, s)$.</td>
</tr>
</tbody>
</table>
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.
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 interactive proofs

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, 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 $\mathbb{Z}_n$. 

Quadratic Residues, Squares, and Square Roots
Square roots in $\mathbb{Z}_n^*$

Recall from lecture 12 that to find points on an elliptic curve requires solving the equation

$$y^2 = x^3 + ax + b$$

for $y \pmod{p}$, and that requires computing square roots in $\mathbb{Z}_p^*$. Squares and square roots have several other cryptographic applications as well.

Today, we take a brief tour of the theory of quadratic resides.
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 $\mathbb{Z}_n^*$

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

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

Why? Because

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

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

$$p | b \iff p | a.$$  

Assume that all quadratic residues and square roots are in $\mathbb{Z}_n^*$ unless stated otherwise.
\section*{\textbf{QR}_n \text{ and } \textbf{QNR}_n}

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

\begin{align*}
\text{QR}_n &= \{ a \in \mathbb{Z}_n^* \mid a \text{ is a quadratic residue modulo } n \}.
\text{QNR}_n &= \mathbb{Z}_n^* - \text{QR}_n.
\end{align*}

\text{QR}_n \text{ is the } \textit{set of quadratic residues} \text{ modulo } n.

\text{QNR}_n \text{ is the } \textit{set of quadratic non-residues} \text{ modulo } n.

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

\[ \sqrt{a} = \{ b \in \mathbb{Z}_n^* \mid b^2 \equiv a \ (\text{mod } n) \}, \]

the \textit{set of square roots} of $a$ modulo $n$.  

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

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

<table>
<thead>
<tr>
<th>$b$</th>
<th>$b^2 \mod 15$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>8 $= -7$</td>
<td>4</td>
</tr>
<tr>
<td>11 $= -4$</td>
<td>1</td>
</tr>
<tr>
<td>13 $= -2$</td>
<td>4</td>
</tr>
<tr>
<td>14 $= -1$</td>
<td>1</td>
</tr>
</tbody>
</table>

Thus, $\text{QR}_{15} = \{1, 4\}$ and $\text{QNR}_{15} = \{2, 7, 8, 11, 13, 14\}$. 
Quadratic residues modulo an odd prime $p$

**Fact**

For an odd prime $p$,

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

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

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

$$|\mathbb{QR}_n| = \frac{|\mathbb{Z}_p^*|}{2} = \frac{p - 1}{2}.$$
Quadratic residues in $\mathbb{Z}_{11}^*$

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

<table>
<thead>
<tr>
<th>$b$</th>
<th>$b^2 \mod 11$</th>
<th>$b$</th>
<th>$-b$</th>
<th>$b^2 \mod 11$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>8</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>9</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Thus, $\text{QR}_{11} = \{1, 3, 4, 5, 9\}$ and $\text{QNR}_{11} = \{2, 6, 7, 8, 10\}$. 
Proof that $|\sqrt{a}| = 2$ modulo an odd prime $p$

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

- It must have a square root $b \in \mathbb{Z}_p^*$. 
  
- $(-b)^2 \equiv b^2 \equiv a \pmod{p}$, 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, $c = \pm b$, so $\sqrt{a} = \{b, -b\}$, and $|\sqrt{a}| = 2$. 

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

- Each $b \in \mathbb{Z}_p^*$ is the square root of exactly one element of $\mathbb{QR}_p$.
- The mapping $b \mapsto b^2 \mod p$ is a 2-to-1 mapping from $\mathbb{Z}_p^*$ to $\mathbb{QR}_p$.
- Therefore, $|\mathbb{QR}_p| = \frac{1}{2}|\mathbb{Z}_p^*|$ 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 $a \in \text{QR}_n$ has exactly four square roots in $\mathbb{Z}_n^*$;
- Exactly $1/4$ of the elements of $\mathbb{Z}_n^*$ are quadratic residues.

In other words, if $a \in \text{QR}_n$,

$$|\sqrt{a}| = 4.$$ 

$$|\text{QR}_n| = \frac{|\mathbb{Z}_n^*|}{4} = \frac{(p - 1)(q - 1)}{4}.$$
Proof sketch

- Let \( a \in \text{QR}_n \). Then \( a \in \text{QR}_p \) and \( a \in \text{QR}_q \).
- There are numbers \( b_p \in \text{QR}_p \) and \( b_q \in \text{QR}_q \) 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, \( |\sqrt{a} \pmod{n}| = 4 \), and \( |\text{QR}_n| = \frac{1}{4} |\mathbb{Z}_n^*| \).

\(^1\)To find \( b_{x,y} \) from \( x \) and \( y \) requires use of the Chinese Remainder theorem.
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\(^2\), let $n = \prod_{i=1}^{k} n_i$, and let $a_i \in \mathbb{Z}_{n_i}$ for $i = 1, \ldots, k$. Consider the system of congruence equations with unknown $x$:

\[
\begin{align*}
  x &\equiv a_1 \pmod{n_1} \\
  x &\equiv a_2 \pmod{n_2} \\
  &\vdots \\
  x &\equiv a_k \pmod{n_k}
\end{align*}
\]

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

\(^2\)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

$$N_i = \frac{n}{n_i} = \frac{n_1 n_2 \ldots n_{i-1} \cdot n_{i+1} \ldots n_k}{n_i},$$

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

$N_i^{-1} \mod n_i$ 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 13.

The solution to (1) is

$$x = \left( \sum_{i=1}^{k} a_i M_i N_i \right) \mod n \quad \text{(2)}$$
Correctness

Lemma

\[ M_j N_j \equiv \begin{cases} 
1 \pmod{n_i} & \text{if } j = i; \\
0 \pmod{n_i} & \text{if } j \neq i.
\end{cases} \]

Proof.

\[ M_i N_i \equiv 1 \pmod{n_i} \text{ since } M_i = N_i^{-1} \text{ mod } n_i. \]

If \( j \neq i \), then \( M_j N_j \equiv 0 \pmod{n_i} \text{ since } n_i|N_j. \)

It follows from the lemma and the fact that \( n_i|n \) that

\[ x \equiv \sum_{i=1}^{k} a_i M_i N_i \equiv a_i \pmod{n_i} \text{ (3)} \]

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

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

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

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

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

---

$^3$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 $\mathbb{Z}_n$. 
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 $v \in \mathbb{QR}_n$ (which she can easily do by choosing a random element $u \in \mathbb{Z}_n^*$ and letting $v = u^2 \mod n$).
- Finally, she chooses $s$ to be the smallest square root of $v^{-1} \mod n$.\(^4\) She can do this since she knows the factorization of $n$.

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

\(^4\)Note that if $v$ is a quadratic residue, then so is $v^{-1} \mod n$.\)
A simplified one-round FFS protocol

Here’s a simplified one-round version.

<table>
<thead>
<tr>
<th>Alice</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose random $r \in \mathbb{Z}_n$.</td>
<td></td>
</tr>
<tr>
<td>Compute $x = r^2 \mod n$.</td>
<td>$\xrightarrow{x}$</td>
</tr>
<tr>
<td>2.</td>
<td>$\leftrightarrow b$</td>
</tr>
<tr>
<td>3. Compute $y = rs^b \mod n$.</td>
<td>$\xrightarrow{y}$</td>
</tr>
<tr>
<td>$\xrightarrow{y}$ Check $x = y^2 v^b \mod n$.</td>
<td></td>
</tr>
</tbody>
</table>

When both parties are honest, Bob accepts Alice because

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

This holds because

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