CPSC 367: Cryptography and Security

Michael J. Fischer

Lecture 17
March 28, 2019

Lamport One-Time Signatures

Merkle Signatures

Authentication Using Passwords
    Authentication problem
    Passwords authentication schemes

Lamport One-Time Signatures

Overview of Lamport signatures

Leslie Lamport devised a digital signature scheme based on hash functions rather than on public key cryptography.

Its drawback is that each key pair can be used to sign only one message.

We describe how Alice uses it to sign a 256-bit message. As wtih other signature schemes, it suffices to sign the hash of the actual message.

How signing works

The private *signing key* consists of a sequence $r = (r^1, \ldots, r^{256})$ of pairs (r_0^k, r_1^k) of random numbers, $1 \leq k \leq 256$.

Let m be a 256-bit message. Denote by m_k the kth bit of m.

The *signature* of m is the sequence of numbers $s = (s^1, \ldots, s^{256})$, where

$$s^k = r_{m_k}^k.$$

Thus, one element from the pair (r_0^k, r_1^k) is used to sign m_k, so $s^k = r_0^k$ if $m_k = 0$ and $s^k = r_1^k$ if $m_k = 1$.

How verification works

The public *verification key* consists of the sequence
$v = (v^1, \ldots, v^{256})$ of pairs (v_0^k, v_1^k), where $v_j^k = H(r_j^k)$, and H is
any one-way function (such as a cryptographically strong hash
function). Thus, v comprises the hashes of all random numbers in
the private signing key r.

To verify a signed message (m, s), Bob checks that $H(s^k) = v_{m_k}^k$
for each k.

Assuming all is well, we have

$$H(s^k) = H(r_{m_k}^k) = v_{m_k}^k$$

for all k, so the verification succeeds.

Forgery resistance

Lamport signatures are one-time because half of the private key is released when it is used.

The revealed r_j^k would allow anyone to construct a valid signature of the same message m, but that signature is already known and valid.

However, if other numbers of the private key are ever disclosed, then Eve could in general produce valid signatures of messages that have never been signed by Alice.

How to forge if the key is reused

Suppose Alice signs m and m' using the same private key.

Suppose further that m and m' differ in two bits j and ℓ. Then Eve can create a signature for the new message m'', where

$$m''_k = \begin{cases} m'_j & \text{if } k = j \\ m_k & \text{if } k \neq j \end{cases}$$

Example: $m = 01101$, $m' = 00100$. They differ in bit positions $j = 2$ and $\ell = 5$. Then Eve can produce a valid signature for $m'' = 00101$ even though Alice never signed m''.

Merkle Signatures

Merkle signature scheme

The Merkle signature scheme applies hash trees to Lamport signatures to allow a large number $N = 2^n$ of messages to be signed, yet each can be verified using the same short trusted verification key.

It works by creating N Lamport signing keys $R_1, \ldots, R_N$ and N corresponding verification keys $V_1, \ldots V_N$. The ith message is signed using key pair (R_i, V_i).

Reducing the size of the trusted verification key

The problem here is the need for Bob to obtain and store the N verification keys.

Rather than distributing all V_i in advance, V_i is sent to Bob along with the ith signed message since it is only needed to verify that one message.

But this creates the problem of validating V_i, since how does Bob know that V_i is genuine rather than a fake designed to make a forged signature look good?

Hash trees to the rescue

Validating received data is what hash trees are good at.

When Alice creates the N Lamport key pairs (R_i, V_i), she puts all V_i into a hash tree, computes top hash, and makes it public.

When she signs the ith message m, she creates a Lamport signature s using the pair (R_i, V_i) as before.

She then sends Bob (m, s) along with V_i and B_i, where B_i is the validation block for V_i.

Verifying the Merkle signature

We assume Bob knows top hash. He validates the signed message $(m, (s, i, V_i, B_i))$ in two steps:

1. He uses the Lamport signature verification algorithm with verification key V_i to check that (m, s) is valid.

2. He checks that V_i is consistent with top hash.

Step 1 is done exactly the same as before, i.e., Bob checks that $H(s^k) = v_{m_k}^k$ for each k.

For step 2, Bob validates V_i against the hash tree by walking up the tree from leaf i to root, computing the hash of each node in turn. For this, he uses the hash values stored in B_i. He checks that the final hash value equals the stored top hash.

Authentication Using Passwords

Outline	Lamport Signatures	Merkle Signatures	**Passwords**
○	○○○○○○	○○○○○	○●○○○○○○○○○

Authentication problem

The authentication problem

The *authentication problem* is to identify with whom one is communicating.

For example, if Alice and Bob are communicating over a network, then Bob would like to know that he is talking to Alice and not to someone else on the network.

Knowing the IP address or URL is not adequate since Mallory might be in control of intermediate routers and name servers.

As with signature schemes, we need some way to differentiate the real Alice from other users of the network.

Outline	Lamport Signatures	Merkle Signatures	Passwords
○	○○○○○○	○○○○○	○○●○○○○○○○

Authentication problem

Possible authentication factors

Alice can be authenticated in one of three ways:

1. By something she knows;
2. By something she possesses;
3. By something she is.

Examples:

1. A secret password;
2. A smart card;
3. Biometric data such as a fingerprint.

Passwords

Assume that Alice possess some secret that is not known to anyone else. She authenticates herself by proving that she knows the secret.

Password mechanisms are widely used for authentication.

In the usual form, Alice authenticates herself by sending her password to Bob.

Bob checks that it matches Alice's password and grants access.

This is the scheme that is used for local logins to a computer and is also used for remote authentication on many web sites.

Outline	Lamport Signatures	Merkle Signatures	**Passwords**
○	○○○○○○	○○○○○	○○○○●○○○○○

Passwords authentication schemes

Weaknesses of password schemes

Password schemes have two major security weaknesses.

1. Passwords may be exposed to Eve when being used.
2. After Alice authenticates herself to Bob, Bob can use Alice's password to impersonate Alice.

Password exposure

Passwords sent over the network in the clear are exposed to various kinds of eavesdropping, ranging from ethernet packet sniffers on the LAN to corrupt ISP's and routers along the way.

The threat of password capture in this way is so great that one should *never* send a password over the internet in the clear.

Outline
○

Lamport Signatures
○○○○○○

Merkle Signatures
○○○○○

Passwords
○○○○○○○●○○○

Passwords authentication schemes

Some precautions

Users of the old insecure Unix tools should switch to secure
replacements such as ssh, slogin, and scp, or kerberized versions of
telnet and ftp.
Do any of you even know what the insecure tools were called?

Web sites requiring user logins generally use the TSL/SSL
(Transport Layer Security/Secure Socket Layer) protocol to
encrypt the connection, making it relatively safe to transmit
passwords to the site, but some do not.

Depending on how your browser is configured, it will warn you
whenever you attempt to send unencrypted data back to the server.

Outline	Lamport Signatures	Merkle Signatures	Passwords
○	○○○○○○	○○○○○	○○○○○○○○●○○

Passwords authentication schemes

Password propagation

After Alice's password reaches the server, it is no longer the case that only she knows her password.

Now the server knows it, too!

This is no problem if Alice only uses her password to log into that *that particular* server.

However, if she uses the same password for other web sites, the first server can impersonate Alice to any other web site where Alice uses the same password.

Outline	Lamport Signatures	Merkle Signatures	Passwords
○	○○○○○○	○○○○○	○○○○○○○○○●○

Passwords authentication schemes

Multiple web sites

Users these days typically have accounts with dozens or hundreds of different web sites. (I personally have over 300.)

The temptation is strong to use the same username-password pairs on all sites so that they can be easily remembered.

But then anyone with access to the password database on one site can log into Alice's account on any of the other sites.

Typically different sites protect data of very differing sensitivity.

An on-line shopping site may only be protecting a customer's shopping cart, whereas a banking site allows access to a customer's bank account.

Outline	Lamport Signatures	Merkle Signatures	Passwords
○	○○○○○○	○○○○○	○○○○○○○○○●

Passwords authentication schemes

Password policy advice

My advice is, use a different password for each account.

Of course, nobody can keep dozens of different passwords straight, so the downside of my suggestion is that the passwords must be written down and kept safe, or stored in a properly-protected password vault.

If the primary copy gets lost or compromised, then one should have a backup copy so that one can go to all of the sites ASAP and change the passwords (and learn if the site has been compromised).

The real problem with simple password schemes is that Alice is required to send her secrets to other parties in order to use them. We will later explore authentication schemes that avoid this.