Royal Holloway University of London

Information Security Group

Using passwords for club membership

Chris Mitchell
Royal Holloway, University of London
http://www.isg.rhul.ac.uk/~cjm

Royal Holloway University of London

Information Security Group

Agenda

- The problem
- Using strong secrets
- An optimistic solution
- Amplifying security of a weak secret
- Bit-wise fair exchange
- Conclusions

Using a password

- Suppose a secret password *p* is used to prove membership of a club.
- When two members (*A* and *B*) meet and wish to verify each other's membership, *A* shows *p* to *B.*
- If *B* is a member, *B* will recognise *p* and hence know that *A* is a member.

Problems

- There are two problems with this procedure.
  1. *A* will now have no way of checking whether or not *B* is a member of the club, since *B* will know *p*.
  2. If *B* is not a legitimate member of the club, then the group password will have been compromised, since *B* will now know it.

Royal Holloway University of London

Information Security Group

Requirement

- How can someone prove knowledge of a password without revealing it?
- There is much work (inc. ISO/IEC 11770-4) on password-based authentication.
- Covers only client-to-server authentication.
- All such protocols assume the client has a trusted copy of the server public key, and so the client can check that the peer entity knows the password.

5

Royal Holloway University of London

Information Security Group

Agenda

- The problem
- Using strong secrets
- An optimistic solution
- Amplifying security of a weak secret
- Bit-wise fair exchange
- Conclusions

6

Strong passwords

- Consider the case where the password is a ‘strong secret’.
- I.e., where the password is chosen from a space sufficiently large that it cannot feasibly be guessed.
- For example, suppose the password is chosen from a set of size 2^{80}.

A simple solution

- In this case the password can be used as a secret key in a shared secret key based mutual entity authentication protocol.
- Such protocols are well-studied, and MAC-based protocols of this type have been standardised (ISO/IEC 9798-4).
- For more on authentication protocols see book by Boyd and Mathuria (*Protocols for key establishment and authentication*).

Another simple solution

- Could also use public key cryptography.
- Group could be assigned a signature key pair, with private key given to every group member.
- A member could then prove membership of the group simply by engaging in a signature-based authentication protocol.

Problems with real-life passwords

- Such protocols are inappropriate if the password is a weak secret, i.e. if it is chosen so that all possibilities can be tested in a modest time.
- An interceptor (and the protocol parties) can exhaustively search for the password.
- Devices used to perform the protocol may be physically insecure and easily lost, i.e. not trusted to store a long-term secret key.
- This forces the use of human-memorable passwords, which tend to be weak.

Agenda

- The problem
- Using strong secrets
- An optimistic solution
- Amplifying security of a weak secret
- Bit-wise fair exchange
- Conclusions

A two-stage approach

- By changing the requirements slightly we can devise a partial solution to the problem.
- This solution has two stages, designed to address threats from two different quarters.

First stage

- Set up a secure channel between the two involved parties.
- This is designed to prevent eavesdropping on communications between pairs of entities to try to learn passwords.
- This could be done using Diffie-Hellman to agree a shared secret key used to encrypt and integrity protect exchanged data.
- Alternatively, if a public key certificate in place, then use an SSL/TLS session.

13

An important note

- This secure channel will be subject to man-in-the-middle attacks.
- A third party could set up secure channels with both parties, and relay data from one channel to the other, and read all data.
- Such an attack may be difficult if parties meet physically and use physical proximity to set up the secure channel.
- Could connect the devices using a cable, providing a physically secure channel.

14

Second stage

- Second stage involves a conventional MAC-based mutual authentication protocol via the secure channel, using the password *p* as the shared secret key.
- Both parties expect responses to messages within a short space of time, e.g. one second.
- Even if *p* has relatively low entropy, it will be infeasible to use a received MAC to exhaustively search for *p* before sending a response.
- So a party that does not know *p* will not be able to complete the protocol in a timely way.

Dealing with attacks

- If an authentication exchange fails because no response is received in the allowed time period (or an invalid response is received) then must assume that this represents an attempted attack.
- If this authentication failure is an attack, then the attacker will potentially be able to work out the password from the MAC sent.
- Hence must assume that the password has been compromised, so a new password must be generated and distributed.

Issues

- I.e., every authentication failure causes a new password to be generated/distributed.
- In practice this may place an unacceptable burden on password administration.
- However, if attacks are rare, then perhaps such a burden can be accommodated.
- We thus refer to this as an optimistic solution, since it works effectively if liars are rare.

Agenda

- The problem
- Using strong secrets
- An optimistic solution
- Amplifying security of a weak secret
- Bit-wise fair exchange
- Conclusions

Background

- We want a password-based authentication to take place without revealing the password to any party, including the entity performing the authentication.
- Suppose party *B* wishes to verify whether party *A* knows the password *p* to assert membership of a particular club, and that *A* is happy to cooperate, but does not wish *B* to learn *p* if *B* does not know it already.

Assumptions I

- *A* and *B* must agree on two security parameters: *r*, *s*.
  - *r* must be large enough that, if a password is used at most *m* times, then *m* randomly chosen *r*-bit strings are likely to be all distinct. E.g., if a password will only ever be used at most 2^{30} times, then $r = 80$ will do.
  - *s* must be as large as possible given that *B* is prepared to perform 2^s MAC function computations to authenticate *A*.

Assumptions II

- *A* and *B* must also agree on a MAC function *f*, where $f_k(X)$ denotes the MAC computed on data string *X* using key *k*.
- If the output of *f* is a *t*-bit string, then *t* must be significantly larger than *s*.
- For example, if *f* is chosen to have a 128-bit output, then this is likely to be sufficient for all conceivable application scenarios.

21

Protocol A

1. *B* randomly chooses an *r*-bit string r_B.
2. *B* sends r_B to *A*.
3. *A* randomly chooses *s*-bit string r_A, and computes $f_p(r_B||r_A)$, (|| denotes concatenation).
4. *A* sends $f_p(r_B||r_A)$ to *B*.
5. Assuming *B* knows *p*, *B* computes $f_p(r_B||r)$ for every possible *s*-bit string *r*, and compares each value to the value received from *A*. If $f_p(r_B||r)$ equals the value received from *A* for any value of *r*, then *B* stops its computations and deems *A* authenticated. If the search ends with no matches then *B* rejects *A*.

22

Properties of protocol A

- *B* performs at most 2^s MAC computations in order to verify *A*.
- If *B* doesn't know *p*, performing an exhaustive search for a password with *u* bits of entropy will require up to 2^{u+s} computations.
- I.e., to determine *p* (assuming *f* is secure) *B* must search through all possible values for *p* combined with all possible values for r_A.
- Since *p* has *u* bits of entropy, such a search will involve up to 2^{u+s} computations of *f*.

23

Numerical example I

- Suppose $u = 30$, i.e. password space has size around 10^9.
- Can be achieved by using as password two randomly chosen English words, or a randomly chosen word and a 5-digit PIN.
- Suppose $s = 20$, i.e. a legitimate user must conduct 2^{20} MAC computations every time (takes one second if can do 10^6 MAC computations per second).

24

Numerical example II

- This means an attacker must perform 2^{50} computations to discover the password.
- This is not totally infeasible, but would impose a significant cost on a casual attacker.
- If attacker has 10 times the resources of a legitimate user (and so can compute 10^7 MACs per second) it would take up to 3 years to find the password.

An issue

- If *B* is a determined attacker, then *B* could pick a fixed value for r_B, and then compute a table of $f_p(r_B||r_A)$ for all possible values of r_A and p.
- Such a table would have size 2^{u+s}.
- *B* could then immediately learn p from any instance of the protocol, simply by looking up the value of $f_p(r_B||r_A)$ received from *A*.

Fixes

- Such a table-based attack can be prevented by introducing an additional nonce into the protocol.
- Also, as in the 'optimistic' scheme, there are advantages if *A* and *B* set up a secure channel before using the protocol. This means that only *B* can perform a search for the password.
- Again as in the optimistic scheme, *B* could be required to prove that it knows the password *p* by providing a response to the message from *A* within a relatively short period of time.

Protocol B (part 1)

1. *B* randomly chooses an *r*-bit string r_B.
2. *B* sends r_B to *A*.
3. *A* randomly chooses an *s*-bit string r_A and an *r*-bit string r'_A, and computes $f_p(r_B||r_A||r'_A)$. [This second random value stops an attack by *B* based on a precomputed table].
4. *A* sends $r'_A \,||\, f_p(r_B||r_A||r'_A)$ to *B*.

Royal Holloway University of London

Information Security Group

Protocol B (part 2)

5. *B* generates $f_p(r_B||r||r_A)$ for every *s*-bit string *r*, and compares each to the value received from *A*.
   If $f_p(r_B||r||\ r'_A)$ equals the value received from *A* for any value of *r*, then *B* deems *A* authenticated and computes $f_p(r||r_B)$ for the successful value of *r*.
   If the search ends with no matches then *B* rejects *A* and terminates the protocol.

29

Royal Holloway University of London

Information Security Group

Protocol B (part 3)

6. *B* sends $f_p(r||r_B)$ to *A*.
7. *A* checks received value by recomputing it. If check fails, or no reply received within specific time interval, *A* assumes that *B* is an impostor, and notifies the password manager of the attack. The password can then be replaced (this need not be immediately, given the work necessary to find the password).

30

Mutual authentication

- The unilateral authentication protocols can easily be extended to provide a mutual authentication protocol.
- In such a scheme both parties are able to verify each other's club membership.
- Indeed, Protocol B is just such a protocol.

An asymmetry

- The only issue that remains is the fact that Protocol B is asymmetric – *B* is required to perform a significant number of computations, whereas *A* is not.
- This could be fixed in two obvious ways.
  - *A* and *B* could perform two instances of the described protocols, one in each direction.
  - Secondly, a protocol can be devised which requires both parties to perform computations.

Agenda

- The problem
- Using strong secrets
- An optimistic solution
- Amplifying security of a weak secret
- Bit-wise fair exchange
- Conclusions

A different approach

- Previous approach makes genuine party perform moderately complex calculations (and attacker perform a very complex calculation to discover the weak secret)
- Now consider an approach involving mutual, verifiable transfer of the password.
- Such protocols have long been proposed in a variety of contexts (e,g. Luby et al. 1983, Tedrick 1984).

One bit at a time

- We first propose a simple iterative protocol which enables the two parties to prove knowledge of (and hence reveal to each other) one bit of the secret password *p* at a time.
- [We second describe a protocol which enables the amount of information per iteration to be reduced to less than a bit.]

Requirements I

- Parties must agree on a parameter *v*, chosen so the probability of an impersonator successfully completing the protocol is 2^{-v}.
- No point in choosing *v* to exceed the entropy of the password in use.
- Probably safe to use $v = 20$ for most applications (since large *v* affects complexity of the protocol).

Requirements II

- Parties must choose a MAC function *f* giving a 1-bit output (e.g. by truncating the output of any sound MAC function)
- Choose a further parameter *r*, so the chance of any party generating the same *r*-bit string twice during the lifetime of the protocol shall be very small.
- For discussion here we assume $r = 128$.

Preparation I

- Every time protocol is used:
  1. *A* chooses a sequence of $v+1$ random numbers $R_{A,i}$ $(0 \le i \le v)$, each *r* bits long. *A* sends $R_{A,0}$ to *B*.
  2. *B* chooses a sequence of $v+1$ random numbers $R_{B,i}$ $(0 \le i \le v)$, each *r* bits long. *B* sends $R_{B,0}$ to *A*.

Preparation II

3. Once A has received $r_{B,0}$, A computes v 1-bit hash values $h_{A,i} = f_p(r_{B,0}||r_{A,i})$, $(1 \le i \le v)$, and sends the v values $h_{A,i}$ to B.
4. Once B has received $r_{A,0}$, B computes v 1-bit hash values $h_{B,i} = f_p(r_{A,0}||r_{B,i})$, $(1 \le i \le v)$, and sends the v values $h_{B,i}$ to A.

Protocol I

- The authentication protocol can then be performed.
- This consists of a sequence of v iterations.
- If any iteration fails, then the protocol is immediately aborted.

Protocol II

- The *i*th round ($1 \le i \le v$) operates as follows:
  1. *A* sends $r_{A,i}$ to *B*. *B* uses this with its stored values of *p* and $r_{B,0}$ to check $h_{A,i}$. If the check succeeds, then *B* continues; otherwise *B* aborts the protocol and rejects *A*.
  2. *B* sends $r_{B,i}$ to *A*. *A* uses this with its stored values of *p* and $r_{A,0}$ to check $h_{B,i}$. If the check succeeds, then *A* continues; otherwise *A* aborts the protocol and rejects *B*.

Remark

- Note that *B* has a '1-bit advantage' over *A*, in that *B* gets to learn *A*'s value $r_{A,i}$ before *B* reveals its own value $r_{B,i}$ to *A*.

Less than one bit per round

- The above scheme can be generalised to enable less information to be exchanged in each iteration.
- This has the advantage that *B*'s advantage over *A* is reduced, at the cost of increasing the number of rounds that it is necessary to perform.

Extra parameters

- *A* and *B* choose two parameters *w* and *v.*
  - The value of *w* determines the amount of information about the password revealed in each round (*w* = 1 yields a protocol essentially the same as just described).
  - As previously, *v* determines the number of rounds to be performed.
  - The probability of successfully impersonating a party in the protocol is $((2^w-1)/2^w)^v$, and *v* should be chosen accordingly.

Preparation I

- The two parties first perform the following preparatory steps.
  1. *A* chooses a set of $wv + 1$ random numbers $r_{A,0}$ and $r_{A,i,j}$ $(1 \le i \le v, 1 \le j \le w)$, each r bits long. *A* sends $r_{A,0}$ to *B*.
  2. *B* chooses a set of $wv + 1$ random numbers $r_{B,0}$ and $r_{B,i,j}$ $(1 \le i \le v, 1 \le j \le w)$, each r bits long. *B* sends $r_{B,0}$ to *A*.

45

Royal Holloway University of London
Information Security Group

Preparation II

3. Once *A* has received $r_{B,0}$, *A* computes wv 1-bit hash values $h_{A,i,j} = f_p(r_{B,0}||r_{A,i,j})$, $(1 \le i \le v, 1 \le j \le w)$.
   For every i, *A* chooses a random binary w-tuple $(g_{A,i,1}, g_{A,i,2}, \ldots, g_{A,i,w})$ subject only to the constraint that it differs from $(h_{A,i,1}, h_{A,i,2}, \ldots, h_{A,i,w})$ in at least one bit position, and sends the v w-tuples to *B*.

46

Royal Holloway University of London

Information Security Group

Preparation III

4. Once B has received $r_{A,0}$, B computes wv 1-bit hash values $h_{B,i,j} = f_p(r_{A,0}||r_{B,i,j})$, $(1 \le i \le v, 1 \le j \le w)$.
   For every i, B chooses a random binary w-tuple $(g_{B,i,1}, g_{B,i,2}, \ldots, g_{B,i,w})$ subject only to the constraint that it is distinct from $(h_{B,i,1}, h_{B,i,2}, \ldots, h_{B,i,w})$, and sends the v w-tuples to A.

47

Royal Holloway University of London

Information Security Group

Protocol I

- The ith round $(1 \le i \le v)$ of the authentication protocol operates as follows:
  1. A sends $(r_{A,i,1}, r_{A,i,2}, \ldots, r_{A,i,w})$ to B. B uses this with its stored values of p and $r_{B,0}$ to compute the w-tuple $(h_{A,i,1}, h_{A,i,2}, \ldots, h_{A,i,w})$.
     B then compares this with the tuple $(g_{A,i,1}, g_{A,i,2}, \ldots, g_{A,i,w})$ received from A during the preparation phase.
     If the two tuples differ in at least one bit position then the check has succeeded, and B continues; otherwise, if the two tuples are identical, B aborts the protocol and rejects A.

48

Protocol II

2. *B* then sends ($r_{B,i,1}$, $r_{B,i,2}$, …, $r_{B,i,w}$) to *A*. *A* uses this with its stored values of *p* and $r_{A,0}$ to compute the *w*-tuple ($h_{B,i,1}$, $h_{B,i,2}$, …, $h_{B,i,w}$). *A* then compares this with the tuple ($g_{B,i,1}$, $g_{B,i,2}$, …, $g_{B,i,w}$) received from *B* during the preparation phase.
If the two tuples differ in at least one bit position then the check has succeeded, and *A* continues; otherwise, if the two tuples are identical, *A* aborts the protocol and rejects *B*.

49

Remark

- Finally note that a further generalisation of the above protocol would involve *A* and *B* sending each other some fixed number (< 2^w) of distinct *w*-tuples in each round, where each of the sent tuples differs from the 'correct' tuple.
- This enables 'fine tuning' of the amount of information transferred between the two parties in a protocol round.

50

Agenda

- The problem
- Using strong secrets
- An optimistic solution
- Amplifying security of a weak secret
- Bit-wise fair exchange
- Conclusions

Goals of work described

- The protocols described are not meant as definitive examples for immediate use.
- Indeed, without a formal model, and proofs within such a model, adoption might be dangerous lest hidden flaws remain.
- The protocols are intended as examples of what might be achieved, to stimulate research on a genuine application issue.

Research questions

- Are the schemes described best possible, or could life for the attacker be made more difficult without imposing additional workloads on the legitimate parties?
- Could precomputations by legitimate parties using the password be used to make life more difficult for the attacker?
- What security model should be used to analyse protocols of the type described?
- Can formal security proofs be devised for the protocols (or for similar protocols)?

53