Authentication protocols based on human comparison of short strings in pervasive computing

Long H. Nguyen and Andrew W. Roscoe

Oxford University Computing Laboratory

University College

{Long.Nguyen,Bill.Roscoe}@comlab.ox.ac.uk

Authentication protocols

- In authentication protocols, parties want to obtain the authentic information such as IDs and public keys of other parties.

- There are some well established methods to achieve this goal based on a PKI or passwords.

- However, the nature of pervasive computing devices introduces a number of new challenges in authentication.

Public key infrastructure

- Authentication is provided by a trusted third party, a Public Key Infrastructure (PKI).

- However, a PKI is expensive to maintain, especially in the environment that has many light weight (wireless) devices whose identities and public keys change very frequently.

- Examples of the devices are credit cards, (mobile) phones, and PDAs that are severely limited in storage and computation power.

Bootstrapping security in pervasive computing

- We do not intend to use a PKI or passwords. However, it is well known that we cannot to bootstrap security from nothing.

- An approach studied by many researchers is to use the Dolev-Yao network in combination with the *authentic/empirical channel* to bootstrap security from scratch.

- The normal Dolev-Yao network (e.g. wireless or Internet, denoted $\longrightarrow_N$) is high-bandwidth, but is controlled by the attacker.

Authentic/empirical channel ($\longrightarrow_E$)

- This is the local, or human mediated, way of identifying the people whom we want to talk to (authenticity property).

- This provides stronger security properties, for example: it cannot be faked, blocked and replayed. (Sometimes *un-delayable* in the *strong* authentic channel: $\longrightarrow_{SE}$).

- Examples of the channel are physical contact first proposed by Stajano and Anderson, human/telephone conversation, and special radio technology which are all very *low-bandwidth*.

Example of application I: Telephone Banking

- In a telephone banking protocol, a customer has to confirm some authentic information over the phone to make a transaction.

- Telephone conversation provides authenticity, but on the other hand is time-consuming and inconvenient.

- We aim to minimise the amount of data required to be confirmed over the phone, and so optimising the human work.

Example of application II: Group meeting

- A group of unknown people in a room want to obtain the public keys of one another to communicate securely via their laptops.

- They can talk to each other their (1024-bit) public keys or copying them by exchanging memory sticks.

- But this is too much human work when the group gets large.

- Either human conversation or visual aid can be employed as the authentic channel in our protocols.

Existing work in this area

- Most researchers concentrate on the case of one-way and pair-wise authentication in a peer-to-peer network.

- Some of them have been discovered not to be optimal in the human work as we are going to discuss in this talk.

- Our main contributions to this area are the group protocol and a new cryptographic primitive termed Digest function.

Protocol notation

- Each party A wants to authenticate its information $INFO_A$ to all other nodes at the end of a successful run of the protocol.

- Each $INFO_A$ might include its identity, an uncertificated public key, a Diffie-Hellman token (g^{x_A}) or its position.

- We denote $INFOS$ as the concatenation of all the $INFO_A$'s.

- Dolev-Yao and the authentic channels are denoted $\longrightarrow_N$ and $\longrightarrow_E$.

Cryptographic hash and Digest functions

- A cryptographic hash $Hash(m)$ is like a normal hash function but also is hard to invert (one-way function) and search for a collision.

- $Digest(k, m)$ is a b-bit output function ($b = 16$ or 20 bits). It has 2 inputs: a public message m and a private key k.

- $Digest(k, m)$ is like a $family$ of short hash functions where each of them is indexed by a key k.

V-MANA I: one-way authentication

(Gehrmann-Mitchell-Nyberg and Vaudenay)

1. $A \longrightarrow_N \quad B : INFO_A$
   A picks a b-bit random number K
2. $A \longrightarrow_{SE} \quad B : K \parallel Digest_K(INFO_A)$

- A wants to authenticate $INFO_A$ to B.
- Both digest output and key are b-bit, 16 for example.
- The authentication string must be both unspoofable and *undelayable*. And therefore we require a strong empirical channel ($\longrightarrow_{SE}$) to transmit – $2b$ – bits.

V-MANA I: one-way authentication

1. $A \longrightarrow_N \quad B : INFO_A$
   A picks a b-bit random number K
2. $A \longrightarrow_{SE} \quad B : K \parallel Digest_K(INFO_A)$

- The authentication string must be both unspoofable and *un-delayable*. And therefore we require a strong empirical channel ($\longrightarrow_{SE}$) to transmit – $2b$ – bits.

- This is clearly not optimal in the human work since – $2b$ – empirical bits only can guarantee at best 2^b security level.

- There is another problem due to the short bit-length of the key.

Digest function

- This relies on a b-bit function $Digest_k(m)$, here m is controlled by the intruder, whereas k is constructed secretly and randomly.

- For all pairs of distinct values (m_1, m_2) and θ, as k varies randomly

$$\textbf{Pr}\left[\mathbf{Digest_k(m_1) = Digest_{k\oplus\theta}(m_2)}\right] \leq \mathbf{2^{-b}}$$

- This has been shown to be satisfied if the key bit-length is greater than some theoretical bound proved by Stinson:

$$\textbf{bit-length}(\mathbf{k}) \geq |\mathbf{m}| - \mathbf{b}$$

An improved protocol

- The bound implies the chance of a successful one-shot attack (or digest/hash collision) is strictly greater than 2^{-b}.
- This leads us to propose an improved version of the scheme. In the below description k_A is a long random key of A.
- The protocol requires manual comparison of a b-bit digest, this is optimal in the human work (2^b security level).

$$
\begin{array}{ll}
1. & A \longrightarrow_N \quad B : INFO_A, Hash(k_A) \\
2. & A \longrightarrow_{SE} \ B : Digest_{k_A}(INFO_A) \\
3. & A \longrightarrow_N \quad B : k_A
\end{array}
$$

Interactive authentication protocols

- Protocols of Hoepman, Wong and Stajano achieve mutual authentication, but require human comparison of multiple short strings.

- This is not optimal when we generalise them to group-version.

- We can reduce multiple into a single b-bit string by clever use of either *indirect* or *direct* information binding strategies.

Multiple-string protocol of Wong-Stajano

1. $A \longrightarrow_N B : A, g^{x_A}, Hash(A, g^{x_A}, R_A, K_A)$
1'. $B \longrightarrow_N A : B, g^{x_B}, Hash(B, g^{x_B}, R_B, K_B)$

R_Y and K_Y are short (16-bit) and long random nonces of Y

2. $\mathbf{A} \longrightarrow_{\mathbf{E}} \mathbf{B} : \mathbf{R_A}$
2'. $\mathbf{B} \longrightarrow_{\mathbf{E}} \mathbf{A} : \mathbf{R_B}$

3. $A \longrightarrow_N B : K_A$
3'. $B \longrightarrow_N A : K_B$

A and B then share the key $k = g^{x_A x_B}$

- Parties compare 2 different short strings/nonces (R_A and R_B).

Improving human work in Wong-Stajano

1. $A \longrightarrow_N B : A, g^{x_A}, Hash(A, g^{x_A}, R_A, K_A)$
1′. $B \longrightarrow_N A : B, g^{x_B}, Hash(B, g^{x_B}, R_B, K_B)$

R_Y and K_Y are short (16-bit) and long random nonces of Y

2. $A \longrightarrow_N B : K_A || R_A$
2′. $B \longrightarrow_N A : K_B || R_B$

3. $\mathbf{A} \longleftrightarrow_{\mathbf{E}} \mathbf{B} : \mathbf{R_A} \oplus \mathbf{R_B}$

- We swap Messages 2 and 3 in the original protocol.
- The humans manually compare a *single* short string: $\mathbf{R_A} \oplus \mathbf{R_B}$.

Improving computation cost in Wong-Stajano

1. $A \longrightarrow_N B : A, Hash(A, g^{x_A}, R_A)$
1′. $B \longrightarrow_N A : B, Hash(B, g^{x_B}, R_B)$

R_Y and g^{x_Y} are short (16-bit) and long random nonces of Y

2. $A \longrightarrow_N B : g^{x_A} || R_A$
2′. $B \longrightarrow_N A : g^{x_B} || R_B$

3. $\mathbf{A} \longleftrightarrow_{\mathbf{E}} \mathbf{B} : \mathbf{R_A} \oplus \mathbf{R_B}$

- We can eliminate long random nonces $K_{A/B}$ because Diffie-Hellman tokens $g^{x_{A/B}}$ can play the role of fresh nonces.

- Input of Hash function in Messages 1 is shortened.

Direct binding authentication protocol

- The short string (digest/shorthash output) depends functionally on the information parties want to authenticate. This can be formalised as follows:

P1 Make all parties who are intended to be part of a protocol run empirically agree a digest of a complete description of the run.

- All parties also need to commit to the final digest before any of them knows what it is: *commitment before knowledge*.

Symmetrised Hash Commitment Before Knowledge

1. $\forall A \longrightarrow_N \forall A'$: $INFO_A, Hash(A \| k_A)$

2. $\forall A \longrightarrow_N \forall A'$: k_A

3. $\forall A \longrightarrow_E \forall A'$: Users compare $Digest(k^*, INFOS)$

   k^* is the XOR of all the k_A's for $A \in \mathbf{G}$

- Each node A creates its own sub-key k_A.

- Each node takes responsibility separately for influencing the final key k^*, and therefore the final digest value $Digest(k^*, INFOS)$.

- Neither any one nor any proper subset of $\mathbf{G}$ can determine the final digest until all the sub-keys are revealed in Messages 2.

ϵ-*almost* **Digest function**

- For all pairs of distinct values (m_1, m_2) and θ, as k varies randomly

$$\text{Pr}\left[Digest_k(m_1) = Digest_{k \oplus \theta}(m_2)\right] \leq \epsilon$$

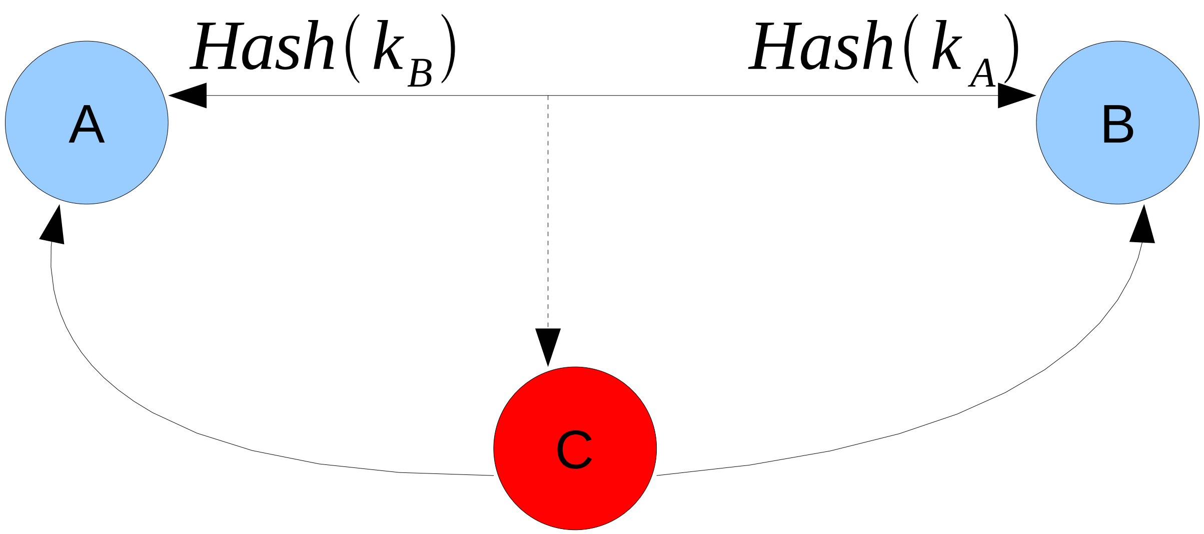
- This is more restrictive than Universal Hash Functions because of the presence of θ. Two definitions are the same when $\theta = 0$.

- In SHCBK, keys vary dynamically/randomly at run time, and can be manipulated to be relatively shifted by θ known to an attacker.

- Whereas in the calculation of MACs they are fixed for all time.

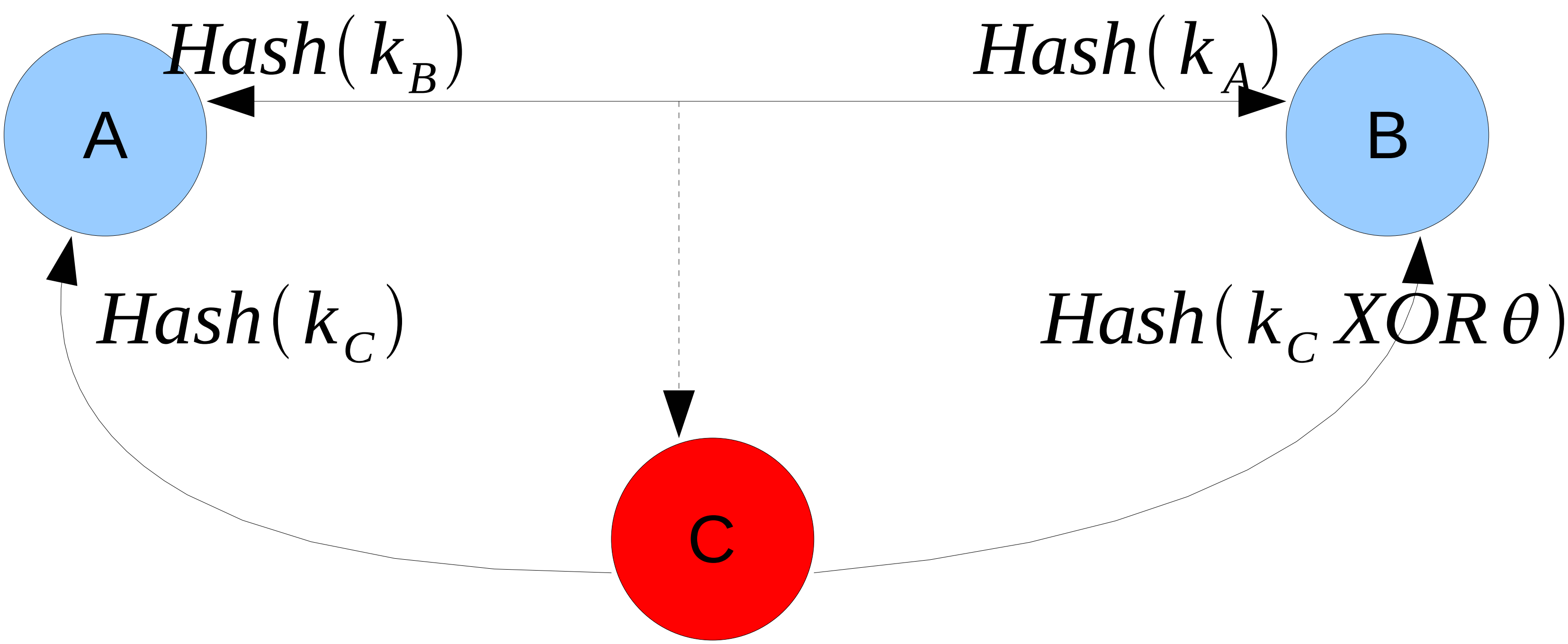
Key manipulation in SHCBK

3 parties A, B, and C run the SHCBK protocol.



Key manipulation in SHCBK

3 parties A, B, and C run the SHCBK protocol.



Party A: $k_A^* = k_A \oplus k_B \oplus k_C$

Party B: $k_B^* = k_A \oplus k_B \oplus k_C \oplus \theta$

Efficiency of the *direct* binding approach

- This is optimal in human work because a b-bit human comparison corresponds to a 2^{-b} chance of a successful one-shot attack.

- As regards computation cost, we use the following cost model:

$$\text{Cost(Hash/Digest)} \approx \text{input-length} \times \text{output-length}$$

- We only need to bind the large data $INFOS$ to the short string (digest output) thanks to the principle **P1**.

- Since the digest-output bitlength is much shorter than a hash output, the digest should be computed very efficiently in practice.

Efficiency of the *indirect* binding approach

- This is also optimal in human work.

- However, it might not be very efficient in computation cost.

- We need to bind large authenticated information by long-output hash function that is more expensive than short-output digest.

Indirect binding group protocol

1. $\forall A \longrightarrow_N \forall A' \;:\; INFO_A, Hash(INFO_A, R_A, K_A)$
2. $\forall A \longrightarrow_N \forall A' \;:\; R_A \parallel K_A$
3. $\forall A \longrightarrow_E \forall A' \;:\; \bigoplus_{A \in \mathbf{G}} R_A$

- Each node has to compute long hash of $INFO_A$ for all $A \in \mathbf{G}$.

- This is more expensive than a short output digest of $INFOS$.

Example of application II: Group meeting

- A group of unknown people in a room want to obtain the public keys of one another to communicate securely via their laptops.

- They can run our group protocol to bootstrap security from scratch.

- This requries the human comparison of a single short 16-bit string.

- Alternatively, the 16-bit string can be used to construct a picture.

Theoretical bounds of Almost-Universal Hashes

- We have discovered Stinson bound: $|k| \geq \log \frac{2^{|m|}(2^b-1)}{2^{|m|}(\epsilon 2^b-1)+2^{2b}(1-\epsilon)}$ is accurate in a very short range of values of ϵ.

- We introduce our new combinatorial bound: $|k| \geq \log \frac{|m|}{\epsilon b}$

- When $\epsilon = 2^{-b}$, Stinson bound gives $|k| \geq |m| - b$ which is much tighter than ours that is $|k| \geq b + \log \frac{|m|}{b}$.

- However, our bound produces a better result when $\epsilon \geq 2^{-b}\left(1 + \frac{b}{|m|-b}\right)$

Implementing the digest function

- We can construct (b-bit output) Digest function based on some well established methods invented for universal hash functions.

- Toeplitz matrix multiplication and pseudo-random number generation proposed by Wegman,Carter,Krawczyk,Mansour and others.

- Error correcting code (Reed-Solomon) by Bierbrauer and others.

Toeplitz Matrix multiplication and PRNG

- We need to derive $b + |m| - 1$ random bits from the key k to construct the Toeplitz matrix R. Using matrix multiplication, we define:

  $$Digest_k(m) = m \odot R \pmod 2$$

  This is equivalent to

  $$d_t = \bigoplus_{j=1}^{|m|} (R_{t,j} \wedge m_j)$$

  and

  $$Digest_k(m) = \langle d_1 \ldots d_b \rangle$$

Efficient implementation of Digest function

- The above algorithm has been shown to satisfy our specification exactly by using a perfect random bit stream.

- In practice, we recommend to use a linear pseudo-random number generator such as shift registers to produce pseudo-random bits, or several seeded with parts of k.

Human interaction: future research

- Efficient ways to present data that can be easily handled by human.

- For example: instead of displaying a string on a screen with –Agree– and –Disagree– buttons.

- We can display the string with a couple of other $random$ ones, and then ask the human to select the correct value.

- Displaying the distorted image of the string.

References

- L. H. Nguyen and A. W. Roscoe. Authenticating ad-hoc networks by comparison of short digests. To appear in *Journal of Information and Computation* in Dec 2007.

- L. H. Nguyen and A. W. Roscoe. Efficient group authentication protocol based on human interaction. *Proceedings of Workshop on Foundation of Computer Security and Automated Reasoning Protocol Security Analysis*, FSC-ARSPA'06. Seattle, Aug 2006.

- L. H. Nguyen and A. W. Roscoe. Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey. Submitted to *Journal of Computer Security*.

Conclusion

- We have analysed a variety of protocols that use the low-bandwidth empirical (authentication) channel to bootstrap security from scratch.

- We have proposed some new protocols both for one-way, two-way authentication and group version that optimise the human work as well as the computation cost.

- A more restrictive version of the Universal hash functions has been introduced, and is termed the Digest function.

- We hope that the family of protocols will find use in a wide variety of applications.