University of Bristol

DEPARTMENT OF COMPUTER SCIENCE

A wearable public key infrastructure (WPKI)

H. Muller N. P. Smart

June 2000 | CSTR-00-006

A wearable public key infrastructure (WPKI)

N.P. Smart H. Muller
Computer Science Department, Woodland Road, University of Bristol, BS8 1UB, UK
nigel@cs.bris.ac.uk, Henk.Muller@bristol.ac.uk

Abstract

We describe the design and implementation of public key infrastructure for the Bristol University Cyberjacket and three initial applications. The first one removes the need for the user to remember passwords, the second application provides an authentic record that a meeting took place, the third provides an authentic record of a conversation. The wearable public key infrastructure (WPKI) we develop uses very small communication, whilst also providing a balanced and low computational overhead on both the 'client' and 'server' sides.

1 Introduction

Many security systems require some form of authenticating token such as a magnetic swipe card, a smart card, a key fob or some other device. The idea being that the possession of the token acts as a proxy for your identity. Hence you *are* the piece of plastic in your back pocket and loosing this token means loosing your identity, or the ability to prove your identity to various applications around you. Looking to the future it would appear that a mobile phone could start to take on a similar functionality since it could subsume all of the different tokens and embed them into one single device.

These tokens and devices, such as phones, have a number of drawbacks for example they provide something extra for you to carry round, they are easily lost or stolen and they are easily used by someone else. It makes more sense to use something that people always have with them to act as a digital proxy (or authentication token), for example your clothing. This has a number of advantages. Whilst it is easy for a pick-pocket to steal your wallet or purse containing your smart cards, or to take your mobile phone when you put it down for five minutes, it is much harder for the thief to steal the clothes you are wearing.

The University of Bristol has for a number of years had an experimental Cyberjacket where ideas for mobile computing have been tested, see [1], [2]. We have designed and implemented a public key infrastructure (PKI) on this jacket. The PKI has been designed so to minimise communication and computational requirements (because of limited power and bandwidth). Using this PKI we have identified three applications: authentication for networked workstations, authenticated records of meetings, and authenticated scripts of communications.

In this paper we first briefly describe the hardware, followed by a detailed discussion of the security requirements in Section 3. The applications are presented in Section 4

The basic motto of the work is that "You are what you wear".

2 Hardware

The Cyberjacket is an ordinary biking jacket with built-in wearable. The hardware that are relevant in thee context of this paper are a processor, GPS, audio I/O interface, and footbridge network. The processor is at this moment an antiquated 486 running Linux. The GPS receiver gives us precise timings and positioning. The audio I/O acts both as speech interface to the wearable and as an interface that allow application programs to store and retrieve audio notes. The footbridge network [1] is an ultra short range network that acts both as a proximity sensing device (it detects that a footbridge sender is present within approximately one foot), and it acts as a communication device allowing two footbridges that are in each others proximity to exchange data.

The hardware has limited specifications, and we expect those to be limited in the future; future wearables will have faster communication and processing devices, but there will always be a trade-off between processing power and bandwidth on the one hand and power consumption on the other hand. The security protocols that we have used are intrinsically small, and will therefore not place high requirements on either processing or communications.

3 Security Requirements

Due to the low bandwidth of communications and storage on a wearable computer it is crucial to keep message sizes to a minimum. We also need to keep computing costs

down to a minimum. It was decided that the system chosen should be extensible in that it should allow Jackets to authenticate themselves to each other and to fixed equipment, such as a workstation.

To obtain authentic public keys one needs to use a public key infrastructure in which each entity can authenticate the public keys in its possession via the use of a certificate.

3.1 Traditional solution using RSA

A traditional public key infrastructure would have used RSA public and private key pairs, see [3] and [4]. With a standard security level of 1024 bits (which is believed to be very hard to crack) this would mean that each signature and each public key would take 1024 bits of information. A certificate would then consist of the following data

(ID, PK, S)

where ID is the data being bound to the public key, PK is the public key being bound to ID and S is the certificate authority's (CA) signature on the pair (ID,PK). Hence this would take at least 2048 bits to store and transmit.

Once having obtained authentic public keys we need to perform various secure communications and challenge response mechanisms which require the agreement of a session key. We would like, for future applications, to have the security property of 'forward secrecy'. A protocol is said to have 'forward secrecy' if finding a session key in the future does not imply that the session keys of the past have then been compromised. Hence, the best way of obtaining session keys is via the use of an authenticated Diffie-Hellman protocol, see [4]. Again using standard technology this would require the exchange of 1024 bit Diffie-Hellman group elements, which would need to be signed so as to achieve authentication. Each party would need to send and receive data of the form

(DH,S)

where S is the signature on the Diffie-Hellman group element DH. Hence, again at least 2048 bits would need to be exchanged. We will show that we can design a protocol that exchanges far fewer bits with the same strength as RSA.

There is another problem with using RSA. RSA is a highly unbalanced algorithm in that the private key operation is much slower than the public key one. Whilst this is fine for verification of certificates this is not efficient for signed Diffie-Hellman, where each party needs to perform one signature and one verification.

3.2 Light weight security using Elliptic Curves

It is for these reasons we decided to adopt Elliptic Curve Cryptography [5] (ECC) as our main public key technology. This allows us to obtain smaller message sizes for the same level of security. In addition ECC provides a balanced signature algorithm so even though the verification is slower than RSA, the extra speed obtained in the signing operation more than compensates for this.

To further reduce both computing time and message sizes we used Implicit Certificates instead of standard certificates, and we used the MQV protocol instead of a signed Diffie-Hellman protocol. This meant that the maximum size of the cryptographic part of any message sent is at most 392 bits, a reduction of 80% in comparison to the RSA protocol shown earlier.

3.2.1 Details of the Elliptic Curve

We use the elliptic curve $P-192$ as recommended by NIST [6]. This curve is based on a field of 192 bits defined by a Generalised Mersenne Prime [7], given by $p = 2^{192} - 2^{64} - 1$. Such a field is advantageous since it allows for efficient field operations. Each field element fits in six 32-bit words and so one obtains 192-bits of security for almost the same computing cost of a field size of 163-bits.

The elliptic curve $P-192$ is also particularly efficient for elliptic curve cryptography since it is of prime order n, hence one does not need to worry about small subgroup attacks. The curve is given by

$$Y^2 = X^3 - 3X + b$$

where, $b = 0x64210519\ e59c80e7\ 0fa7e9ab\ 72243049\ feb8deec\ c146b9b1$. The use of a curve with minus three as the coefficient of X also provides an advantage when performing the basic curve operations. The curve $P-192$ comes with a fixed base point which in the following discussion we shall denote by P. Like in almost all elliptic curve systems we shall use SHA-1 as the hash function, which we shall denote by H.

The points on an elliptic curve form an (additive) finite Abelian group, with the group law being given by the formulae, with zero $\mathcal{O}$: Suppose $P = (x, y)$ then

$$-P = (x, -y).$$

If $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, with $x_1 \neq x_2$, we set

$$\begin{aligned} \lambda &= \frac{y_2 - y_1}{x_2 - x_1}, \\ x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= (x_1 - x_3)\lambda - y_1. \end{aligned}$$

and then

$$P_3 = (x_3, y_3) = P_1 + P_2.$$

In the case where $x_1 = x_2$ and $y_1 = -y_2$ the two points add up to the zero, i.e.

$$P_1 + P_2 = \mathcal{O}.$$

The final case is when $x_1 = x_2 = x$ and $y_1 = y_2 = y \neq 0$, in which case we are said to be doubling a point. In this case we set

$$\lambda = \frac{3x^2 - 3}{2y}.$$

and then

$$(x_2, y_2) = [2]P = P + P$$

where x_2 and y_2 are given by the formulae

$$\begin{aligned} x_2 &= \lambda^2 - 2x, \\ y_2 &= (x - x_2)\lambda - y. \end{aligned}$$

The basic cryptographic operation on elliptic curves is to compute $[k]P$, where k is some random number and P is a point on the curve. By $[k]P$ we mean the result of adding the point P to itself k times, this can be done in $O(\log k)$ curve additions using the formulae above using the binary method. Various optimisations of the calculation of $[k]P$ given k and P are given in [5].

3.3 Implicit Certificates

The certificate authority, CA, has a public/private key pair given by $([c]P, c)$. The public key $[c]P$ is embedded into every Jacket. When a jacket wishes to register with the CA, the jacket and the CA will agree on an implicit certificate, based on some user ID such as the name of the jackets owner. This is analogous to registration in standard PKI systems.

To obtain an implicit certificate the jacket sends a certificate request to the CA which consists of an ephemeral public key $[t]P$, where the jacket keeps t secret. Using point compression techniques this requires a single 200 bit message.

The CA then computes another random integer k and computes

$$\gamma = [t]P + [k]P$$

and the element

$$s = cH(\mathrm{ID}||\gamma) + k \pmod{n}$$

where the hash is over the concatenation of the string ID and the compressed representation of γ. The CA then sends the triple (ID,s,γ) to the jacket and stores the pair

$$(\mathrm{ID}, \gamma)$$

in a world readable database. This transfer to the jacket requires around 400 bits whilst the storage of γ requires 200 bits, both assuming point compression is used.

The jacket then computes its private key

$$a = t + s \pmod{n}$$

and stores (ID,a,γ).

The implicit certificate is the pair (ID,γ), since given this pair we can recover the public key of the jacket via the equation

$$[a]P = [H(\mathrm{ID}, \gamma)]([c]P) + \gamma.$$

As soon as the jacket uses its private key by performing some private key operation, any other entity can determine that the computed public key is correct since otherwise the associated public key operation would not make sense. Hence, from the jacket using its private key correctly one obtains implicit assurance that the public key is bound with its ID.

Since every jacket has a copy of the CA's public key we now have the ability for each jacket to authenticate each other on the basis of the owners ID.

3.4 MQV Key Agreement

We now explain how the jacket and another entity, be it a PC or another jacket, can agree on a secret shared session key by only exchanging around 200 bits and still providing forward secrecy. This is done via the MQV protocol [8]. We assume that the jacket already has a public/private key pair embedded into it as does the party it is communicating with.

Both parties first need to exchange authentic public keys. This is either done by exchanging implicit certificates or by looking up such implicit certificates in a public directory. The latter course being more suited to a PC than a jacket. The two authentic public/private key pairs we shall denote by $([a]P, a)$ and $([c]P, c)$.

The jacket generates an ephemeral public/private key pair $([b]P, b)$ and sends the pair (ID,$[b]P$) to the other party. The other party generates a public/private key pair $([d]P, d)$ and sends $([d]P)$ to the jacket.

We give the MQV protocol from the point of view of the jacket. To obtain the view from the other parties perspective one needs to swap every occurrence of (a, b, c, d) for (c, d, a, b).

- Compute $x = (x([b]P) \pmod{2^{96}}) + 2^{96}$.
- Compute $s = b + xa \pmod{n}$.
- Compute $x = (x([d]P) \pmod{2^{96}}) + 2^{96}$.
- Compute the elliptic curve point

$$Q = [s]([d]P + [x]([c]P)).$$

- If Q is not the point at infinity then one can take the hash of the x-coordinate of Q as the shared secret value.

4 Applications

Using the lightweight Public Key Infrastructure described above we have implemented three applications. We first describe how we load an identity in a jacket, using a network of workstations as a Certificate Authority (CA). We then described three applications: one to allow the user to login automatically to a workstation based on the identity of the jacket and the proximity of the wearable to the workstation. Then an application for creating an authentic record of two jackets meeting each other. Finally, an application to create an authentic record of a conversation between two users of wearables.

4.1 Storing an identity in the jacket

In our first application we make use of the login process to a workstation to act as the registration for the CA. The workstation is the certificate authority and hence has a public/private key pair given by $([c]P, c)$. The public key $[c]P$ is embedded into the Jacket, as stated above, whilst the private key is held on the workstation in a file which can only be read by the super user. The private key should clearly be held on a central server in a more trusted environment, but for the moment this is how the architecture is.

The first time a user wearing his Cyberjacket logs onto the workstation the jacket and the workstation will agree an implicit certificate, as described above. Since the user must be present at the workstation to log onto it, this provides a means of linking the jacket with the users system ID. Since our workstation is using Linux, the ID is nothing more than the username, which is a string of eight characters.

4.2 Automatic logon

This application allows a user wearing the Cyberjacket to be automatically logged onto a workstation. This application closely follows the badges of Olivetti/Cambridge/ATT [9], but our protocol is bomb-proof in that on cannot break the authentication. Even if someone manages to snoop the communication between the Cyberjacket and the workstation, then this does not give the snooper sufficient information to log into the workstation.

We assume that the jacket already has a public/private key pair embedded into it. Since the user logged onto the system in the past and agreed an implicit certificate with the system, the jacket needs only to send its ID to the system (via the footbridge radio link embedded into the cuff of the jacket, and a footbridge link embedded in the mouse-pad). Using this ID the workstation looks up the implicit certificate of the jacket in a public directory and hence recovers the public key of the jacket. The jacket already has the public key of the workstation since it has embedded into it the public key of the CA, which in our case is the workstation.

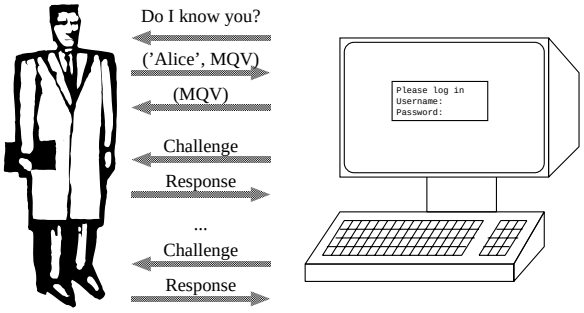

Figure 1. Logging onto a workstation

Then the jacket and the workstation can perform the MQV protocol and agree on a secret key. Using this secret key the jacket and the workstation proceed with a challenge response protocol based on a symmetric cipher, for which we used RC4 [10]. As long as the jacket responds correctly the workstation automatically logs the user in, without the need for a password, as outlined in Figure 1. However if the jacket should respond incorrectly, either due to the jacket not being genuine or the user walking away from the workstation, then the user is logged out. We have implemented this application embedding a transmitter in the cuff of our jacket and the receiver in the mouse pad of a workstation.

4.3 Meeting application

Our second application makes use of the fact that the jacket has a GPS system embedded into it. The GPS system gives an accurate (outdoor) position, and an accurate time. This allows for the recording of dates, times, and places when people met. This has applications in legal and commercial arguments, where evidence of a meeting at a specific time between two people is of great commercial value.

We assume we have two jackets, having obtained implicit certificates in some way such as that described above. We assume that the jackets are owned by Alice and Bob, who want to achieve an authentic record of their meeting.

Alice's jacket first sends Bob's jacket the tuple,

(Alice's Implicit Certificate, Alice's ID, Time, Location).

Bob's jacket can then verify that he agrees to the Time and Location (within certain limitations of accuracy) and returns the tuple

(Bob's Implicit Certificate, Bob's ID, Sign(Time, Location, Alice's ID)).

Finally Alice responds with

(Sign(Time, Location, Bob's ID)).

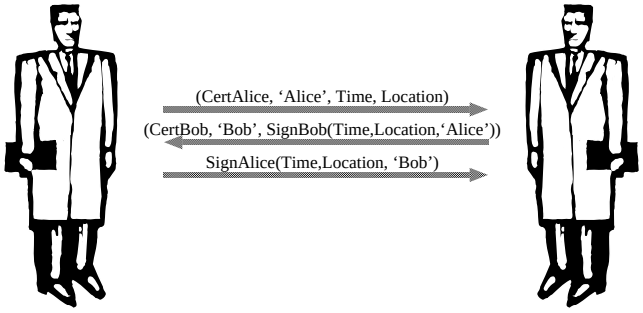

Figure 2. Protocol for obtaining an authentic record of the time and location of a meeting

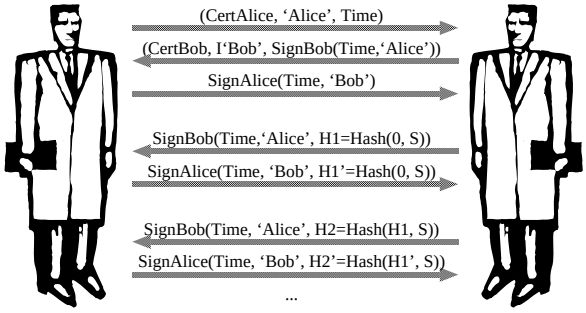

Figure 3. Protocol for obtaining an authentic record of a conversation

This process is sketched in Figure 2. Now each jacket has a copy of an authentic record that Alice and Bob met at a certain time and in a certain place. One problem with the above protocol is that Alice may refuse to send the final signature. This can be because of two problems

- Alice is honest and Bob is cheating. In which case Alice has decided not to pursue the protocol, since something was wrong with Bob's response, and the only advantage which Bob obtains is a copy of Alice's Implicit Certificate, but this is public knowledge. In this case the jacket that will inform the wearer, Alice, that Bob is not trustworthy.
- Bob is honest and Alice is cheating. Now Bob has committed to having met Alice at a given time and place without similar evidence being obtained from Alice. But since Bob is honest why should he worry about anyone knowing who he met and when, he clearly has no privacy concerns since he willingly gave up his signature on this information.

The record can only be use as proof against each other that a meeting has taken place; it cannot be used as an alibi because both people can cheat and can generate a fake time/location pair.

4.4 Authentic record of conversations

The jacket can also be used to create an authenticated record of the conversation during the meeting. This functionality has applications in evidence gathering by police forces and in the negotiation of commercial contracts where paper contracts are hard to draw up, such as on a trading floor of a financial market. The protocol is similar to the one above, except that we are encoding a digitised version of what has been said.

In the first instance we assume that the Jackets are communicating with each other via some wireless link, using microphones and earpieces; for example a mobile phone. Each jacket is going to record the digital conversation, and sign each others record. So Alice's jacket will record the conversation and send a signed version of it (discussed below) to Bob's jacket. Likewise Bob's jacket sends signed versions of his recording to Alice's jacket.

To achieve these signatures on a continuous audio stream we perform the following: At time zero, set $H_0 = 0$, then for time interval i (every thirty seconds say) perform the following two steps (as visualised in Figure 3).

1. $H_i = Hash(H_{i-1}, S)$ where S is the digital signal received from Bob over the last time interval.
2. Sign H_i and transmit it to Bob.

Whilst Alice's jacket is performing this operation, Bob's jacket is doing likewise and so both parties obtain a signed version of the conversation attached to their own recording of the conversation. Both versions are signing the same conversation, because we assumed a digital exchange of data, hence both Alice's and Bob's jacket can check that the signatures are genuine.

An extension of this application which is non trivial would be to sign conversations between two people that are in each others proximity. In this case each jacket will make its own digital recording of the conversation, and sign it accordingly for the other party. However, in order for Alice to check whether the signed conversation is real, the only solution would be for Bob to transmit his version of the recorded signal to Alice in conjunction with the signature, which will allow Alice to listen to this recording via an earpiece. Of course there will be a very small lag in the signal, causing Alice to hear the conversation with one ear at real-time, and via the other ear with say, 0.05 seconds delay. Clearly Alice can at any time break off the communication if Bob's jacket was cheating and sending and signing corrupted data.

5 Conclusions

We have described a lightweight Public Key Infrastructure which is suitable for wearable computers. The choice of ECC technology has allowed us to do this will small message sizes and low computing costs. The additional choice of using implicit certificates and the MQV protocol has further reduced the size of the transferred messages. We contend that placing digital authentication in users clothing rather than in some external device is far more intuitive and acceptable.

We have also described three applications which make use of our wearable Public Key Infrastructure. Each of these makes use of the additional hardware that the Bristol Cyberjacket already has on it, namely a GPS receiver and a footbridge network.

References

[1] P. Neaves and J. Bedford-Roberts. Dynamic connection of wearable computers to companion devices using near-field radio. In *Proceedings of The Second International Symposium on Wearable Computers*, pages 156–157, October 1998.

[2] N. Cambell, H. Muller, and C. Randell. Combining positional information with visual media. In *Proceedings of The Third International Symposium on Wearable Computers*, pages 203–205, October 1999.

[3] R.L. Rivest, Shamir A., and L.M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM*, 21:120–126, 1978.

[4] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[5] I.F. Blake, G. Seroussi, and N.P. Smart. *Elliptic Curves in Cryptography*. Cambridge University Press, 1999.

[6] FIPS PUB 186-2 : DIGITAL SIGNATURE STANDARD (DSS). Nist, 2000.

[7] J.A. Solinas. *Generalized Mersenne Numbers*. preprint, 1999.

[8] A. Menezes, M. Qu, and S. Vanstone. Some new key agreement protocols providing mutual implicit authentication. In *Workshop on selected areas in cryptography (SAC '95)*, pages 22–32, 1995.

[9] Roy Want, Andy Hopper, Veronica Falcao, and Jonathon Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.

[10] B. Schneier. *Applied Cryptography*. John Wiley and Sons, 1996.