NDSS’98 Trust Management Panel:
LE NOZZE DI NOMEN

Bob Blakley
blakley@us.ibm.com

“You may say to yourself, ‘this is not my beautiful wife!’
You may ask yourself, ‘my God, what have I done?’”
--David Byrne

What follows is a slightly embellished transcript of my panel presentation at ISOC NDSS‘98 in San Diego, California.

At the conference, the part of Nomen was played by Li Gong of Javasoft; the part of Clavis was played by Barbara Fox of Microsoft; the part of Pastor was played by Brian Dillaway of Microsoft; and I played the part of Chorus myself. Li, Barbara, and Brian participated in this session without the benefit of rehearsal (or even advance knowledge of the script!) and I owe all of them a big “thank you” for their smooth performances and good humor.

==

The title of today’s panel is “Trust Management”. People often say they’re going to talk about trust management, but then dive straight into certificates, and signing code, and policy for verifying signatures. So I wanted to spend a little time today talking about what exactly any of that has to do with “trust”, and what it might mean to manage trust.

It’s important to recognize that trust is not keys, nor is it in keys. It’s also not certificates, nor is it in certificates. Finally, it’s not policy, though policy may authorize actions based on explicit or implicit claims of trust.

Still, I think it is hiding in among these artifacts, and I think people are managing it. But I think it’s hiding in a place many people may find surprising. I’ll tell you where I think trust is hiding and who I think is managing it at the very end of the session. But first, I want to give you a way to think about trust which isn’t technical and which is familiar enough that it will make some of the issues with managing trust obvious.

I’m going to do this using a little skit.

LE NOZZE DI NOMEN

A farce in one scene

DRAMATIS PERSONAE

Clavis: a woman in a long white dress

Nomen: a young man in a coat and tie

Pastor: a distinguished-looking older gentleman in a plain robe

Chorus: as usual, explains the pieces of the action which are too expensive or difficult to work into the dramatic action.

SCENE ONE

(A long room with a vaulted ceiling, with rows of seats facing the front.)

(Nomen and Clavis stand at the front of the room with their backs to the audience, Nomen on the right, Clavis on the left. Between them, facing the audience, stands Pastor. Chorus lurks in the wings.)

Pastor: By the power vested in me by Verisign, I now pronounce you Name and Key. What RSA hath joined, let no man put asunder.

Chorus: That's what a digital certificate does. It binds a name to a key, according to the authority of some CA.

Pastor: I now present Mr. Nomen and Mrs. Clavis. Welcome them among you.

Chorus: Well, it looks like a wedding, doesn't it? This is what happens when we accept a digital certificate - we're asked to bless the nuptials. Seems pretty simple, doesn't it? What could go wrong? On the other hand, we're away from home, and this isn't our congregation. Before we say anything we might regret, let's ask a few questions.

Mrs. Clavis, did you promise to honor and obey Mr. Nomen here?

Clavis: Obey?? Isn't that a little old-fashioned?

Chorus: It may be old-fashioned, but we'd like to know that the key always expresses the name's intent! We can't have you sneaking around behind Mr. Nomen's back signing things he might repudiate later.

(to the audience) This means that the owner of a key must not allow the key to be applied without his knowledge and consent. Does your software support this? Does it always prompt for a password before allowing a signature operation, for example?

(to Nomen) Did you promise to honor and cherish Mrs. Clavis?

Nomen: Of course! Doesn't everyone?

Chorus: Indeed they do! Everyone knows that users need to protect their keys and keep them private.

Did you both swear "until death do you part"?

Nomen: No! There's a repudiation procedure. I have to find three witnesses and say "I divorce you" three times.

Clavis: Wait a minute! That's not right! We have to go back to Mr. Pastor and prove that no valid marriage ever existed. And it's called "annulment", not "divorce"!

Chorus: Let me guess - you two didn't grow up in the same church.

Clavis: That's right...

Chorus: (to the audience) You need to be clear about the rules for repudiation and revocation.

(to Nomen and Clavis) Do your parents approve of the wedding?

Nomen: Actually, we didn't ask them. We eloped!

Chorus: (to the audience) Oh, this could be bad. There are important things which the "parents" of a name and key need to agree to, like whether the Pastor has the authority to marry the name, and whether they key can be used for the purposes to which the certificate will be put.

While we're on the subject, the Pastor has to approve also. His "certification practice statement" describes the rules and regulations for the newlyweds. Let's ask him a question.

(to Pastor) What do you think about this repudiation issue?

Pastor: We will grant divorces, but there has to be a good cause. We can also declare marriages invalid if we find out that something was wrong that we didn't know about.

Chorus: Like if you find out that Mr. Nomen was already married to another key?

Pastor: Oh, no! We allow polygamy! But of course, Mrs. Clavis can't be married to another name. And if he can prove she's been fooling around with another name on the sly, that's grounds for revocation!

Chorus: How traditional. If that happens -- if Mr. Nomen here finds another name with his key, what happens to their property after the revocation?

Pastor: Oh, Mr. Name is responsible for all obligations Mrs. Clavis entered into up to the moment he discovered the infidelity.

Chorus: That's in the eyes of the Church, right? What do the civil courts say?

Pastor: It kind of depends. If Mr. Nomen can find hard evidence that some other name has been running around with his key, some courts may let him off the hook.

Chorus: On a related topic, Mrs. Clavis, were you chaste before the marriage?

Clavis: That's NONE of your business!!

Chorus: (to the audience) I'm afraid it is our business, though of course we hate to pry. We've just been told to let this key speak for Mr. name here as long as she's faithful. How can we trust Mrs. Clavis if we don't know whether anyone else has had his hands on her private part?

You need to pay attention to who generates keys and how they're handled before their owners get them.

(to Clavis) Mrs. Clavis, do you think anyone might have compromising pictures of you?

Clavis: Ummm, it's kind of embarrassing really, but the FBI might have some...

Chorus: (to the audience) Make sure signature keys don't get escrowed!!

By the way, this means that you're trusting your Public Key Infrastructure to know the difference between signature keys and other kinds of keys, and to treat signature keys right. Does your software do this?

(to Clavis) Now, Mrs. Clavis, I'm going to have to ask you another delicate question. Are you really female?

Clavis: (what Clavis actually said at this point in the presentation was "I'm not going to say that!!" Wouldn't you know that my directorial debut would coincide with an actors' strike? What the script says here is:)

Well, it was bound to come out sooner or later. Actually, this is a post-operative condition.

Chorus: (to the audience) Good little keys have to have the right mathematical properties. If the key in a certificate you receive has been carelessly chosen (using a weak random number generator, or an inadequate primality test, or whatever) it could be possible to forge signatures.

(to Nomen) And how about you, Mr. Name, are you a real man?

Nomen: You caught me too! In fact, I'm a virtual reality simulation.

Chorus: (to the audience) Does your CA ask for proof of identity before it seals a name into a certificate?

Could lara.croft@eidosinteractive.com get a certificate?

Could "The IBM Corporation" get a certificate? If so, what would a signature using that certificate be good for? Today, no one signs "The IBM Corporation" on a check or a contract; instead, they sign "John Q. Public, Treasurer, for the IBM Corporation". This has important legal implications.

(to Pastor) As long as we're on this topic, are you really a minister?

Pastor: Of course!

Chorus: Who ordained you?

Pastor: I ordained myself!

Chorus: (to the audience) Not exactly apostolic succession, is it? But if you look in the root key list in your browser, you'll find it's just the same story -- the certificates are self-signed!

(to Pastor) Do the civil authorities recognize the marriages you perform?

Pastor: Well, they do in Utah. I'm not sure about other places.

(exeunt Pastor, Nomen, et Clavis)

Chorus: Now our play is done; the time has come for the chorus to tie up the loose ends and bring down the curtain.

The morals of our little tale are:

- Don't assume it's a wedding, until you know what church you're in and what the vows were.
- The things that make the real world a messy place, and trust a risky commodity... do not change just because some clever guys invent a code that uses two keys instead of just one.

You have to trust a lot of people to use a certificate:

- The issuer

  * to stay in business
  * to do background checks and vet requesters.
  * to generate good keys and not to escrow signature keys
  * to revoke and to publish revocations
  * to abide by its Certification Practice Statement
  * to protect its own keys and processes.
- The acceptor

  * to keep his private key private
  * to abide by his obligations in the issuer's CPS
  * not to repudiate fraudulently
- The acceptor's software

  * not to reveal the private key
  * not to allow the use of the private key absent the acceptor's intent
- Your software

  * to verify signatures correctly and check for revocation, repudiation, and expiration
  * to protect your root key list
- The courts

  * to uphold obligations entered into using digital signatures

Many things are not in a certificate, including:

- an identity
- proof that the acceptor has the private key
- proof that others don't
- proof that the key is a good key

So if you want to be safe, you should take some precautions:

- Read your CA's Certification Practice Statement
- Find out how & by whom keys are created and distributed

- Read disclaimers before you sign
- Know the identity behind the name!
- Know what your software will authorize if you sign
- Look at the root CA list!
- Know the law in all relevant jurisdictions (or hire someone who does)
- Protect your keys (get a smartcard!)
- Sign contracts defining the terms and conditions for certificate and signature acceptance

If anyone knows a reason why this Name and this Key should not be joined in Holy Matrimony, let him speak now or forever hold his peace!

(exit Chorus; house lights down slowly)

FINIS

I promised that at the very end, I'd reveal where I think trust is hiding and who is managing it. So very briefly, here's my theory:

- A certificate ties a name to a key
- A key ties actions together into a sequence
- Humans -- acceptors of certificates -- evaluate sequences of actions and use them to construct their individual views of the identity represented by the sequence's name. Trustworthiness is one of the attributes of these individual views of identities.

So, certificates allow EACH INDIVIDUAL END USER (not the CA; not the Public Key Infrastructure; not a security administrator; not application software; not policy management software) to build an identity around a name, and use knowledge of that identity to make trust decisions.

Once a user has made a trust decision, he may configure his software to remember that decision and enforce it as policy for some period of time. But as new actions are added to sequences over time, the user's trust decisions may change...

Trust management is:

- The system's retention and presentation to the user of information related to sequences of actions performed by the same name, as verified by the key bound to that name by a certificate, plus
- The user's subjective evaluation of the significance of that information, plus
- The system's support for storage, enforcement, and evolution of the policy the user develops based on that information.