Issues in Cryptography

Ronald L. Rivest

MIT Laboratory for Computer Science

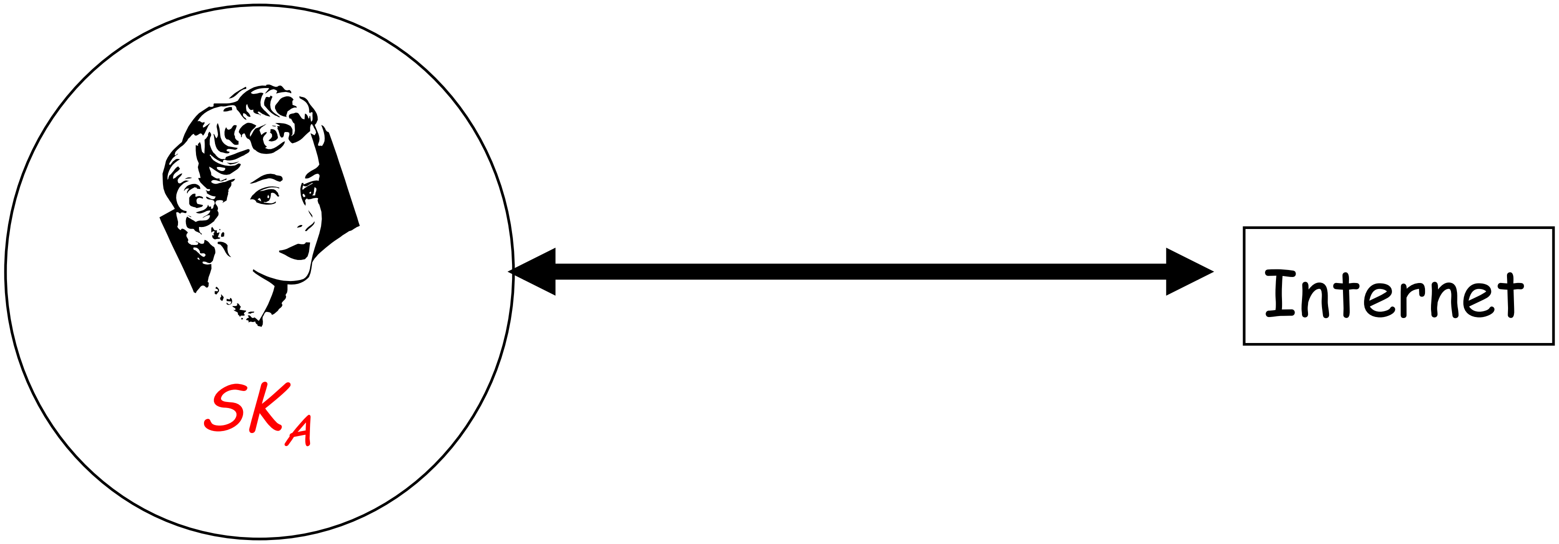
Outline

- “Where’s Alice?”
  ---The Secure Platform Problem
- Digital Signatures
- Repudiation

The "Alice abstraction"

- Assumes Alice can generate and use her secret key SK_A, while keeping it secret.
- Alice's secret key SK_A is her "cyber-soul", her "electronic identity" (or pseudonym), her way of identifying herself. SK_A ***is*** Alice!

Cryptography in Theory



SK_A

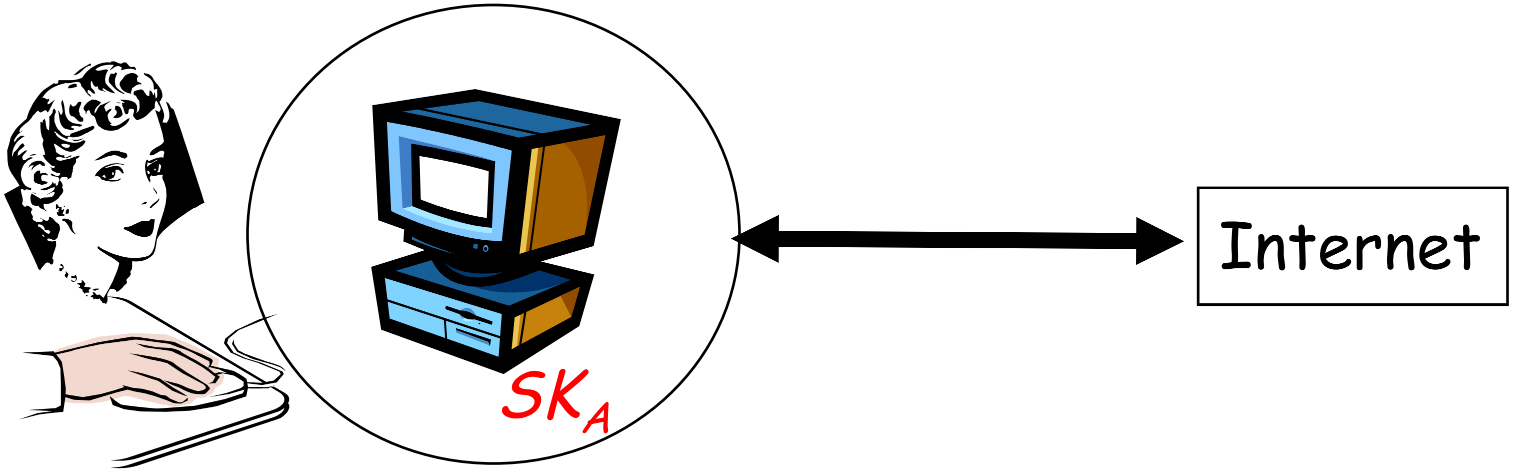
Internet

Alice

But Alice is not a computer!

- Alice needs a computer (or at least a processor) to store her secret key SK_A and perform cryptographic computations on her behalf.
- In particular, her processor should produce Alice's digital signature when appropriately authorized…

Cryptography in Practice



SK_A

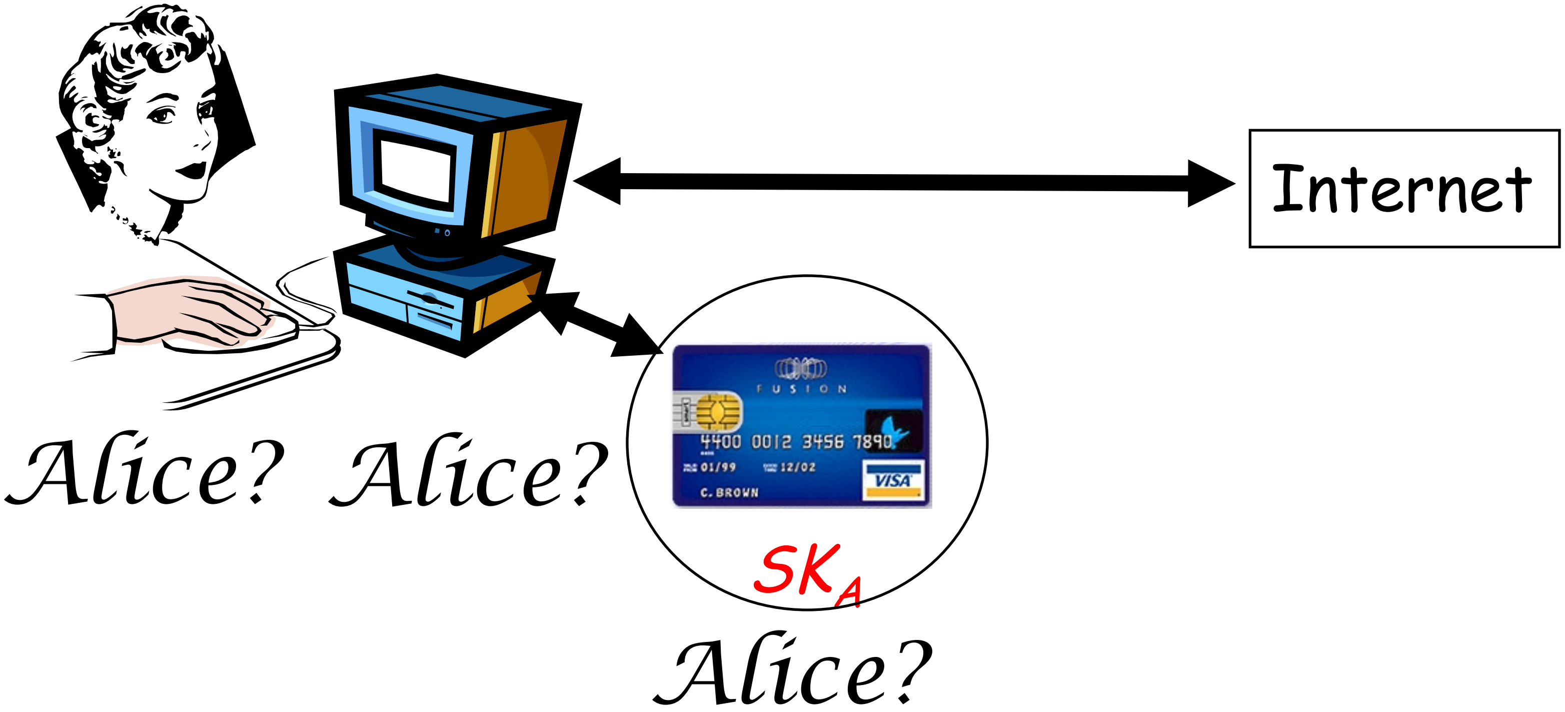
Internet

Alice? *Alice!*

But her OS is not secure!

- Modern OS's (Windows, Unix) are too complex to be adequately secure for many applications (viruses, Trojan horses).
- Would *you* base the security of an Internet presidential election on the security of Linux?
- Alice's key SK_A may be vulnerable to abuse or theft…

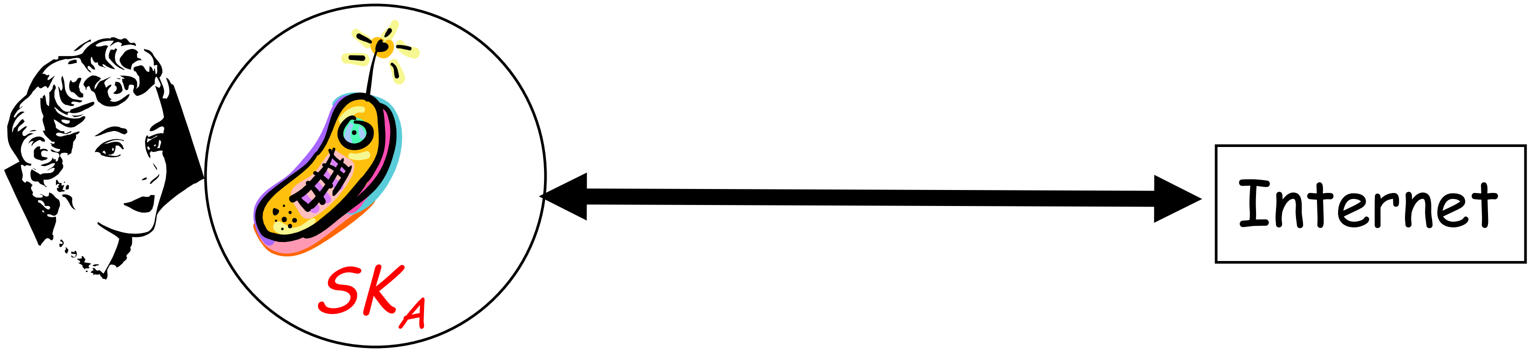
Can SK_A go on a smart card?



But her OS is still not secure!

- Smart card has no direct I/O to Alice.
- When Alice authorizes a digital signature, she must trust OS to present correct message to smart card for signing.

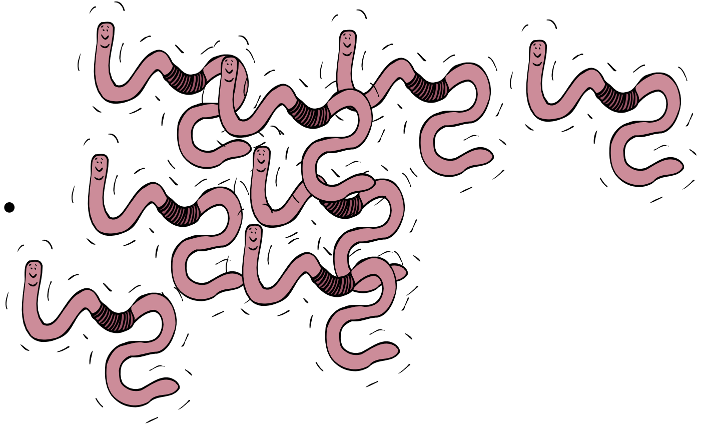
Can SK_A go on a phone or PDA?



Alice? Alice?

But this looks very familiar!

- Same story as for PC, but smaller!
- PC smart card → Phone SIM card.
- Phones now have complicated OS's, downloadable apps, the whole can of worms.
- Little has changed.

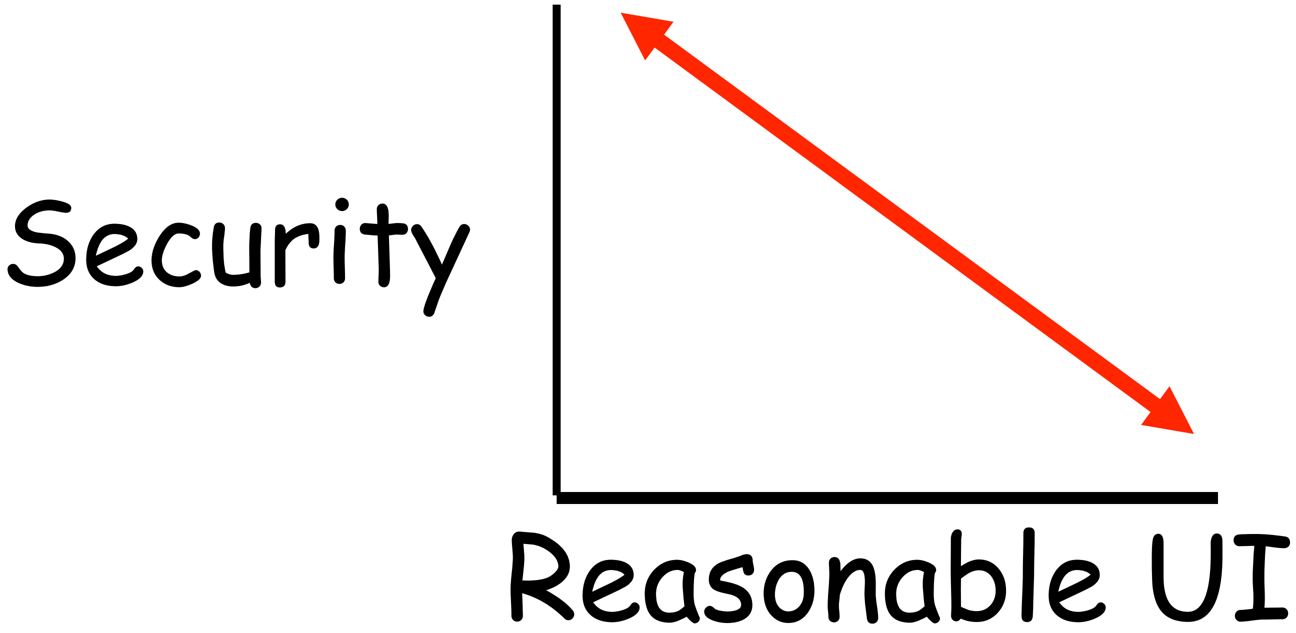


Why can't we solve problem?

- There is a *fundamental conflict*!
- Downloadable apps and complexity are:
  - *Necessary* for reasonable UI
  - *Incompatible* with security

The Sad Truth?

- *The following are incompatible:*
  - A reasonable UI
  - Security

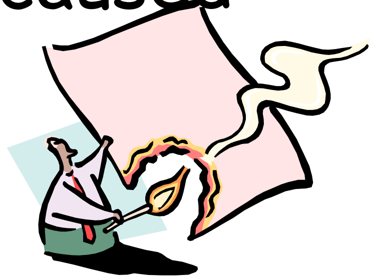


But Digital Sigs Need Both!

- *Security*
  to protect secret key and securely show user what is being signed.
- *Reasonable UI*
  to support complex and variable transactions.

Are Digital Signatures Dead?

- *As usually conceived, perhaps…*
- We should change our mind-set:
  - A digital signature is not *nonrepudiable proof* of user's intent, but merely *plausible evidence*.
  - We should build in *repudiation mechanisms* to handle the damage that can be caused by malicious apps.
  - Repudiate *signatures*, not *keys*.



Use a Co-Signing Registry

- Signature not OK until saved and co-signed by user's *co-signing registry* (e.g. at home or bank).
- User can easily review all messages signed with his key.
- Registry can follow user-defined policy on co-signing.
- Registry can notify user whenever his key is used to sign something.

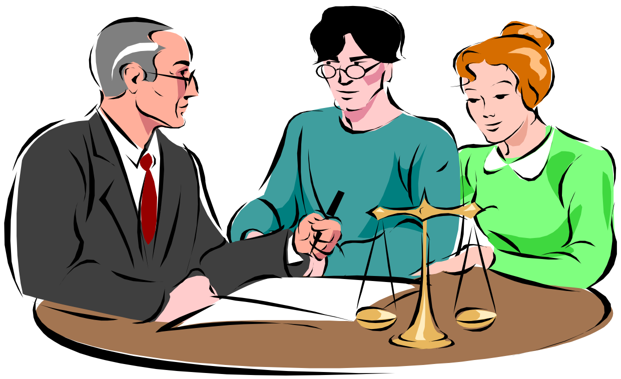
Use One-Time Signing Keys

- Registry can give user a set of *one-time* signing keys, so damage from key compromise is limited. Registry won't co-sign if key was used before.

In this case, registry really holds user's secret signing key, and signs for him when authorized by one-time key.

Repudiation

- May not be so hard to live with, once we accept that it is necessary.
- Consistent with legal status of handwritten signatures (can be repudiated, need witnesses for higher security).



Conclusions

- Cryptography works great, but insecure OS's make digital signatures problematic, because of conflict between security and reasonable UI's.
- Design systems that are robust in face of some key abuse (Alice may not always know what is being signed by her key!)

(THE END)