Balancing Security and Ease-of-Use on the Sun SPOTs

Vipul Gupta
Distinguished Engineer
Sun Microsystems Laboratories

2007 Sun Labs Open House

Agenda

- Sun SPOT overview
- Goals
- Secure over-the-air reprogramming
- Secure communication
- Future work

Agenda

- **Sun SPOT overview**
- Goals
- Secure over-the-air reprogramming
- Secure communication
- Future work

Sun SPOT

Sun Small Programmable Object Technology

- Explores the next wave in the evolution of computing devices
- Basic device has three layers
  - > Rechargeable battery
  - > Processor board w/ radio
  - > Application specific board w/ sensors/actuators
- Processor board alone acts as a base station
- User programmable in Java™ using standard tools
- Available for purchase in the US
  www.sunspotworld.com

Initial Target Markets

- Typical sensor applications:
  - Structural/environmental monitoring, Proactive health care, Asset tracking, Military surveillance ...
- Sun SPOTs more capable, flexible, easier to use – best suited to:
  - **Education:** embedded systems, robotics, design classes
  - **Research:** Flexible, easy-to-deploy platform for research topics ranging from gesture interfaces to volcanic activity
  - **Hobbyist:** Powerful platform, easy to program, easy to interface
- Beyond sensors: Program the World!

Guess what this is ...

http://news.com.com/When+art+meets+wireless+sensors/2100-11392_3-6105881.html

Agenda

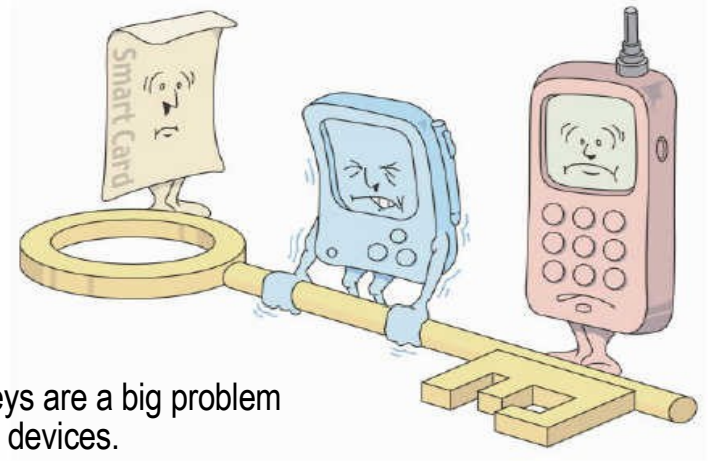
- Sun SPOT overview
- **Goals**
- Secure over-the-air reprogramming
- Secure communication
- Future work

Goals ...

Security mechanisms must be simple to use, user-friendly

- "Over-the-air" is way of life, requires security
- Convenience frequently trumps security
- Most target users are not crypto-savvy
- Security mechanisms need to be efficient
- The hard part, trust management, should "just work" in the simplest, most common scenarios

Elliptic Curve Cryptography



Large keys are a big problem for small devices.

Sym.	RSA	ECC	Ratio	MIPS yrs
80	1,024	160	6:1	10^{12}
112	2,048	224	9:1	10^{24}
128	3,072	256	12:1	10^{28}
192	7,680	384	20:1	10^{47}
256	15,360	521	30:1	10^{66}

Best before 2010

- Highly resource efficient, NSA-endorsed, next-gen public-key cryptosystem
  - > Smaller keys than RSA for equivalent security
  - > Faster computations: memory, bandwidth, energy savings
  - > Advantage improves as security needs increase
- Standardized by NIST, ANSI, IEEE, IETF, gaining broad vendor support
- Good match for AES

SPOT Cryptographic Library

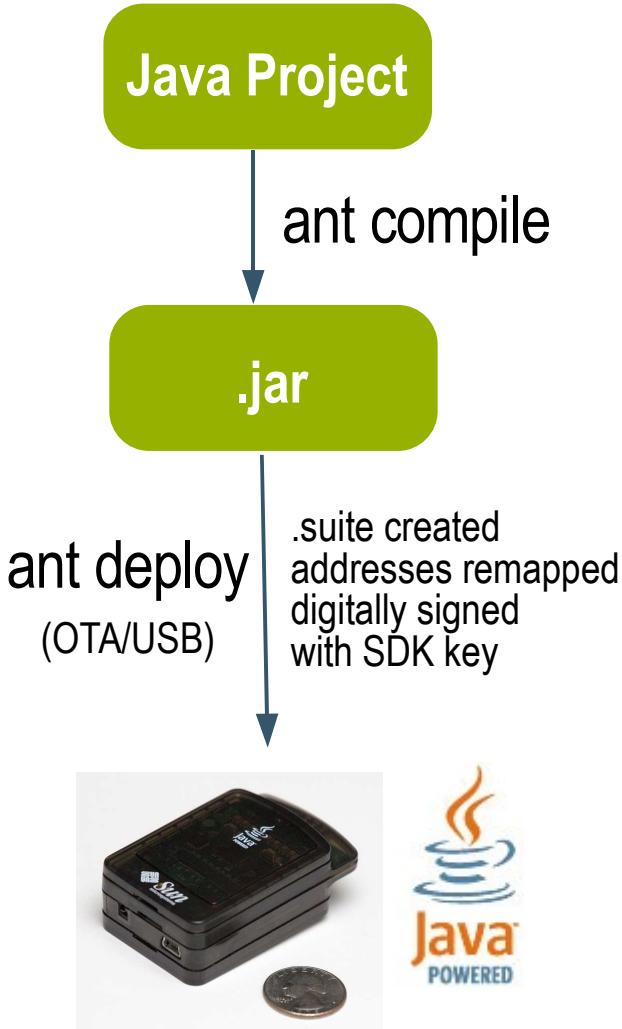
- Highly efficient, pure Java code
- Implements:
  - Elliptic Curve Cryptography for digital signatures, key agreement
    - More than 60x faster than bouncycastle.org code
  - Message Digests for data integrity, authentication
  - Ciphers for data confidentiality
- API modelled after Java SE

Agenda

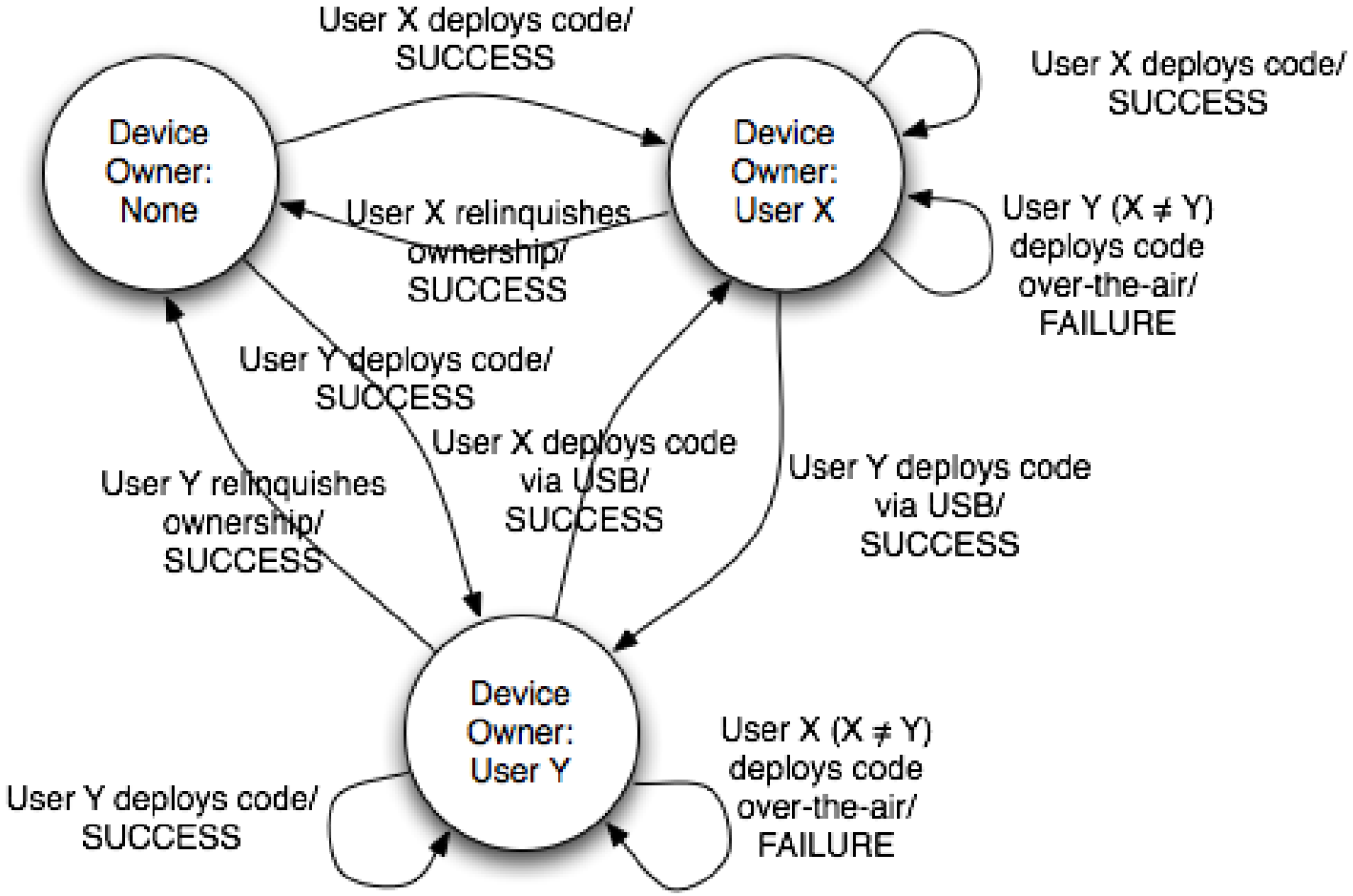
- Sun SPOT overview
- Goals
- **Secure over-the-air reprogramming**
- Secure communication
- Future work

Secure Code Deployment

- Digital signature used to “seal” verified byte codes on desktop
- Code sent over-the-air (OTA) or USB
- Signature verified on SPOT before execution
- Addresses both
  - > Java branding issues
  - > general code authentication



Device Ownership Model

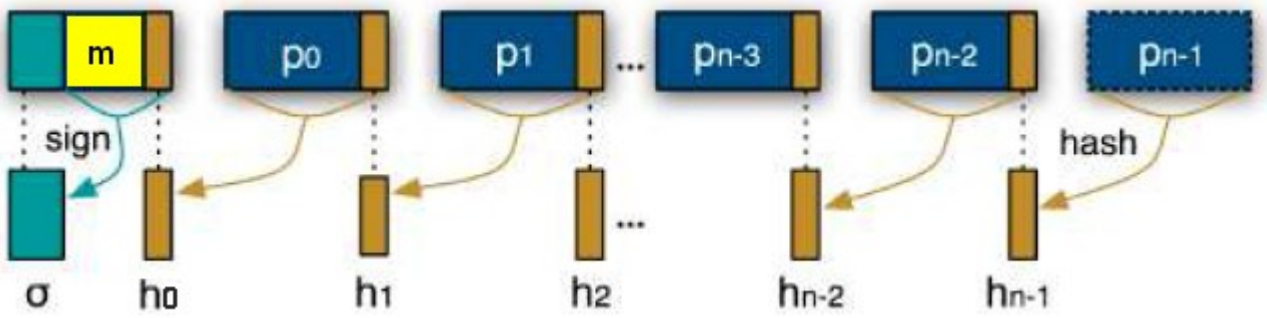


Key Management for Code Deployment

- Balances security, convenience
- Mostly user-transparent; simple “ownership” model
- Each user's SDK has a public/private key-pair
- SPOT stores trusted public-key of “owner”
- First deployer becomes owner
- Owner has special privileges (can deploy new code or restore SPOT to ownerless state)
- Policy permits ownership change with physical access

Epidemic Secure Code Deployment

- User updates version info in MANIFEST.MF, deploys to one SPOT, code spreads epidemically
- Dissemination is pipelined across multiple hops, tolerates node failures and packet loss (via SNACK)
- Node compromise does not permit unauthorized code dissemination



Agenda

- Sun SPOT overview
- Goals
- Secure over-the-air reprogramming
- **Secure communication**
- Future work

Secure Communication

- Inspired by the simplicity of “https” user experience
- Applications require one simple change

```
conn = Connector.open(“radiostream://“ + addr + “:” + port);
conn = Connector.open(“sradiostream://“ + addr + “:” + port);
```

- Reuses SSL protocol underneath with ECC
- Certificate management is user-transparent in typical usage, extends key management from ownership model

Key Management for Secure Communication

SDK requests SPOT's Public Key

SPOT generates a key pair Pub_{SPOT} and $Priv_{SPOT}$, sends Pub_{SPOT}

SDK creates an X.509 certificate $Cert_{SPOT}$ with Pub_{SPOT} and signs it using its private key $Priv_{SDK}$. SDK sends $Cert_{SPOT}$, its own certificate $Cert_{SDK}$

$Cert_{SPOT}$ stored in key store as the SPOT's personal certificate, $Cert_{SDK}$ stored in key store as the owner's certificate

Key Management for Secure Communication (contd)

- SPOT uses $Cert_{SPOT}$ to identify itself in SSL handshake
- Secure communication between SPOTs belonging to the same owner just works!
- Additional scenarios also supported, *e.g.*, trusting web CAs and SPOTs belonging to other owners

Agenda

- Sun SPOT overview
- Goals
- Secure over-the-air reprogramming
- Secure communication
- **Future work**

Future Work, Conclusion

- Efficiency optimizations
- Integration into Sun SPOT SDK
- Integration with SPOT World GUI

Sun SPOTs ... making embedded programming accessible

secure

Questions?

Vipul Gupta
vipul.gupta@sun.com