# Connect Now to MitM

Kaisa Nyberg
Helsinki University of Technology
Nokia Research Center
Finland

Crypto 06 Rump Session  22 August 2006

# Microsoft Windows Rally

- Allows easy and secure setup of networks of devices particularly for home users
- Windows Vista Connect Now-NET implements a snapshot of

**Simple Config** by WiFi Alliance

(work still in progress)

- Allows different configuration technologies, one method is

**Diffie-Hellman Key Agreement**

authenticated by a device password entered by the user and

**Device Password Proof-of-Possession (DPwPoP)**

protocol by Simple Config

- For more information, see

http://www.microsoft.com/whdc/rally/

# DPwPoP Protocol

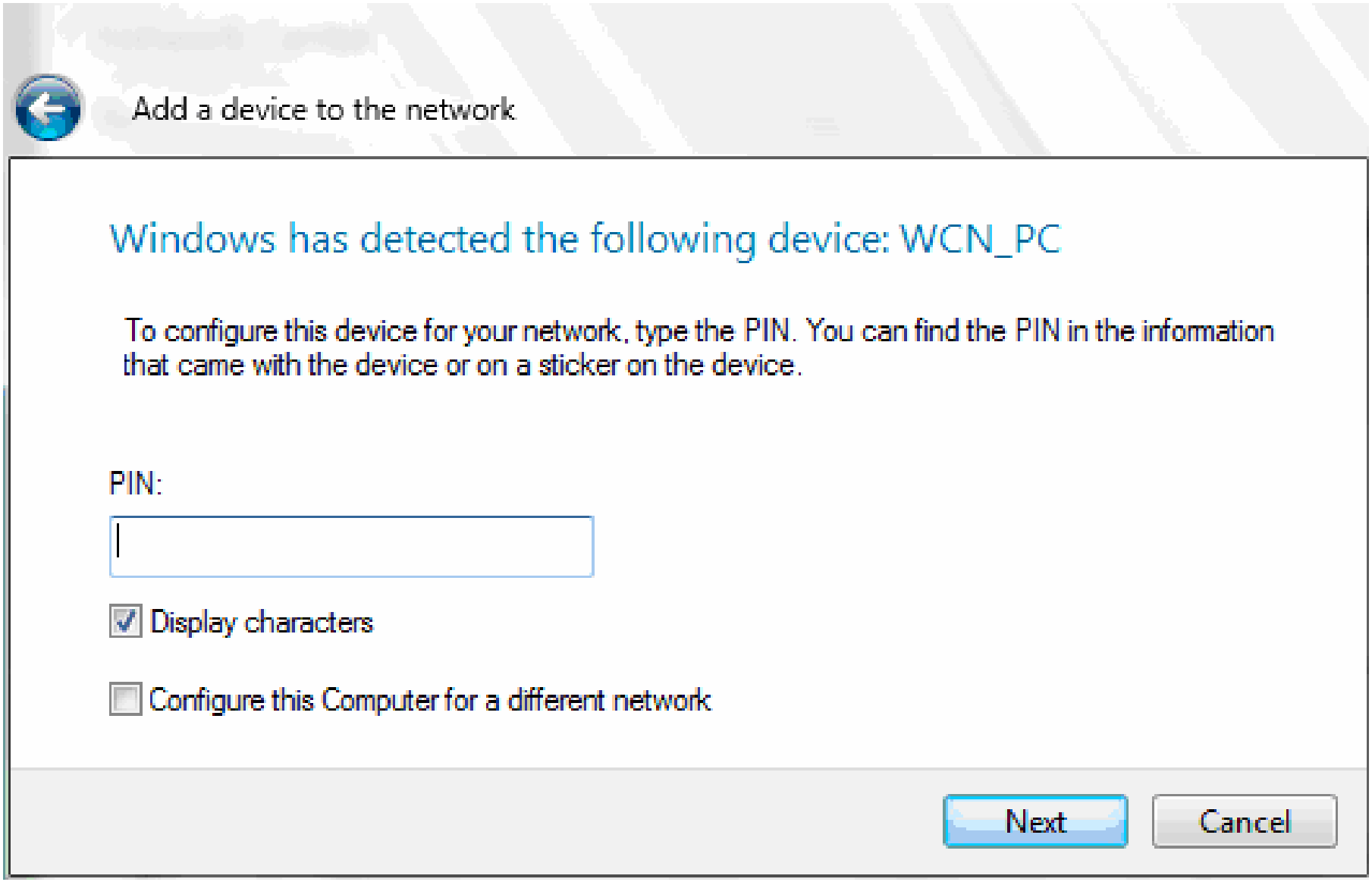
Enrollee → Registrar:
*M1* = Version || N1 || Description || PKE
Enrollee ← Registrar:
*M2* = Version || N1|| N2 || Description || PKR || HMACAuthKey(*M1* || *M2**)
Enrollee → Registrar:
*M3* = Version || N2 || E-Hash1 || E-Hash2 || HMACAuthKey(*M2* || *M3**)
Enrollee ← Registrar:
*M4* = Version || N1 || R-Hash1 || R-Hash2 || ENCKeyWrapKey(R-S1) || HMACAuthKey (*M3* || *M4**)
Enrollee → Registrar:
*M5* = Version || N2 || ENCKeyWrapKey(E-S1) || HMACAuthKey (*M4* || *M5**)
Enrollee ← Registrar:
*M6* = Version || N1 || ENCKeyWrapKey(R-S2) || HMACAuthKey (*M5* || *M6**)
Enrollee → Registrar:
*M7* = Version || N2|| ENCKeyWrapKey(E-S2 ) || HMACAuthKey (*M6* || *M7**)
Enrollee ← Registrar:
*M8* = Version || N1 || HMACAuthKey (*M7* || *M8**)

# DPwPoP Protocol

- Simple protocol using commitments with interleaving opening
- Commitments are hiding, but not for short data that can be searched exhaustively (offline dictionary attack)
- Security against MitM achieved ONLY IF fresh random passkey used for each execution of the protocol.

  BUT this does not always seem to be the case in Windows Rally… a recommended option is to use static password.

# Find 8-digit Password on Sticker



# DPwPoP Protocol & MitM (1)

**1st Attempt:**

Enrollee → MitM → Registrar:

*M1* = Version || N1 || Description || PKE

Enrollee ← MitM ← Registrar **(†)** :

*M2* = Version || N1|| N2 || Description || PKR || HMACAuthKey(*M1* || *M2**)

Enrollee → MitM → Registrar **(†)** :

*M3* = Version || N2 || E-Hash1 || E-Hash2 || HMACAuthKey(*M2* || *M3**)

Enrollee ← MitM← Registrar **(†)** :

*M4* = Version || N1 || R-Hash1 || R-Hash2 || ENCKeyWrapKey(R-S1) || HMACAuthKey (*M3* || *M4**)

Enrollee → Mitm → Registrar:

*M5* = Version || N2 || ENCKeyWrapKey(E-S1) || HMACAuthKey (*M4* || *M5**)

*Verification fails at the Registrar (with high probability )*

**Execution of the protocol is aborted.**
**MitM learns the first 4 digits of the password.**

**(†) MitM uses different AuthKeys and KeWrapKeys with Enrollee and Registrar**

# DPwPoP Protocol & MitM (2)

**2nd Attempt:**

Enrollee → MitM → Registrar:

*M1* = Version || N1 || Description || PKE

Enrollee ← MitM ← Registrar (†) :

*M2* = Version || N1|| N2 || Description || PKR || HMACAuthKey(*M1* || *M2**)

Enrollee → MitM → Registrar (†) :

*M3* = Version || N2 || E-Hash1 || E-Hash2 || HMACAuthKey(*M2* || *M3**)

Enrollee ← MitM ← Registrar (†) :

*M4* = Version || N1 || R-Hash1 || R-Hash2 || ENCKeyWrapKey(R-S1) || HMACAuthKey (*M3* || *M4**)

Enrollee → MitM → Registrar (†) :

*M5* = Version || N2 || ENCKeyWrapKey(E-S1) || HMACAuthKey (*M4* || *M5**)

Enrollee ← MitM ← Registrar:

*M6* = Version || N1 || ENCKeyWrapKey(R-S2) || HMACAuthKey (*M5* || *M6**)

*Verification fails at the Enrollee (with high probability)*

**Execution of the protocol is aborted.**

**MitM learns the second 4 digits of the password.**

# DPwPoP Protocol & MitM (3)

**Third attempt:**

M1 … M2 … M3 … M4… M5… M6 … M7 … M8 …

**Third time lucky!**

**User finally succeeds connecting the devices ... to the Man-in-the-Middle!**