Blaming the cryptographic user

Daniel J. Bernstein

University of Illinois at Chicago, Technische Universiteit Eindhoven

Unrelated advertisement:
http://nametags.cr.yp.to

Cryptography promises to provide confidentiality, integrity, availability against network attackers.
Oops, is cryptography failing to meet this promise?
Oops, is cryptography failing to meet this promise?

Traditional response: Blame the user.
Oops, is cryptography failing to meet this promise?

Traditional response:
Blame the user.

e.g. Padding-oracle attacks broke RSA SecurID 800 tokens?

“RSA reminds all of its customers to apply the latest OS security patches . . . An end user should remove the RSA SecurID 800 device from its USB port when not in use.”
e.g. Timing attacks extracted kernel’s AES keys? “Don’t allow untrusted code to run alongside your crypto code.”
e.g. Timing attacks extracted kernel’s AES keys? “Don’t allow untrusted code to run alongside your crypto code.”

e.g. ECDSA nonce reuse leaked PlayStation 3 signing key? Sony “made a critical mistake.”
e.g. Timing attacks extracted kernel’s AES keys? “Don’t allow untrusted code to run alongside your crypto code.”

e.g. ECDSA nonce reuse leaked PlayStation 3 signing key? Sony “made a critical mistake.”

e.g. Breaks in RSA-512? MD5? RSA-1024? User should have selected different primitives.
e.g. Timing attacks extracted kernel’s AES keys? “Don’t allow untrusted code to run alongside your crypto code.”

e.g. ECDSA nonce reuse leaked PlayStation 3 signing key? Sony “made a critical mistake.”

e.g. Breaks in RSA-512? MD5? RSA-1024? User should have selected different primitives.

e.g. RSA-2048 painfully slow? User should have bought more hardware to handle the load.
A different response:
Build a cryptographic library that eliminates the failures.
A different response:
Build a cryptographic library
that eliminates the failures.

http://nacl.cr.yp.to

Joint work with
Tanja Lange (Eindhoven),
Peter Schwabe (Academia Sinica);
various code contributions from
Matthew Dempsky (Mochi Media), Niels Duif (Eindhoven),
Emilia Käsper (Leuven),
Adam Langley (Google),
Bo-Yin Yang (Academia Sinica).