

Identity-Based Encryption with Conventional Public-Key Infrastructure



Jon Callas
PKI '05
April 2005

©2005 PGP Corporation

Identity-Based Encryption

- First proposed by Adi Shamir in 1984
- Certificates are data structures that can be hard to manage
- What if public keys could be a function of an identity?
 - An identity is an arbitrary bit string.
 - Could directly convert the string into a public key pair
- Can this lower or eliminate the need for certificates?
- At the very least, does this ease enrollment issues?

IBE Construction

- IBE systems consist of four components:
 1. System Setup
 - Private Key Generator (PKG) uses master secret to generate key pairs
 - Establishes all keygen parameters
 2. Encryption
 3. Key Extraction
 - Owner of identity (ID) authenticates to PKG to get private key of key pair
 4. Decryption

Progress in IBE

- Pairing-based and other systems provide mechanism for producing public key pairs from strings
- This has renewed interest and research in IBE
- However,
 - New IBE systems use their own public-key cryptosystems
 - Incompatible with existing systems
 - Requires new software
 - Requires new standards
- Can we answer Shamir's IBE question with a conventional PKI (using RSA, Elgamal, DSS, EC Discrete Logs, etc.)?

Issues with IBE

- Requires a central server
 - Server holds a master key that all public key pairs are derived from
 - Thus, IBE systems are key escrow systems
 - Requires extra computer security on this server; one secret determines all keys
 - No worse than systems like Kerberos, but no better, either
 - Difficulties arise from the *expectation* that users may truly possess keys

Issues with IBE (cont'd_

- Naming issues become paramount
 - All PKIs have naming issues (Ellison, etc. have discussed)
 - Key management becomes name management
 - jon, jcallas, jon.callas, jon_callas, じよなさん 鳥, cto, cso
 - Misspellings are also keys
 - john, calls, callis, etc.
 - The John Wilson problem can be solved with appropriate qualifiers
 - Key management doesn't go away, it becomes slightly different

Issues with IBE (cont'd)

- Revocation is difficult

- If a name is equivalent to a key, does revoking the key require revoking the name?
- Solutions include appending qualifying information to the name
 - “jon@pgp.com || 20-Apr-2005”
- Can lead to explosion of names and qualifiers
- Still need to distribute updated information

- Legal Complications

- EU data signature laws require private keys to be held *only* in signer's hand
- Some believe this applies to encryption keys as well

On-Line and Off-Line Key Generation

- An interesting feature of some IBE systems is off-line key generation
 - Client contacts the server, gets generation parameters, then can create keys
- In an on-line world, off-line generation is nice but superfluous.
- Off-line generation may be a security flaw
 - This is a directory harvest attack for all possible keys
 - Makes changing parameters more difficult
 - On-line access still needed for revocation
 - Makes agglomerating equivalent names impossible as it is distributed
 - Exacerbates naming issues and naming errors
- Giving up off-line generation gives us flexibility in creating a system

Conventional IBE Keys

- Architecture

- Select Master Key
- Select Identity Digest Function
- Seed PRNG with result of IDF
- Generate public key with PRNG
- Wrap in appropriate data structures

- Example implementation

- 512 bit HMAC key, 512 bit salt
- HMAC-SHA512(k, salt||id)
- PGPsdk PRNG
- PGPsdk keygen functions for RSA
- OpenPGP certificate, X.509 cert, SPKI, etc.

- Other selections possible for key components, depending on desired qualities of system, multiple systems can be used within a single domain



Is This Really IBE?

- It satisfies Shamir's question, but with software engineering, not mathematics.
- It is an IBE keygen function
- Also has been called *Attribute Based Enrollment*
- It isn't an IBE cryptosystem
- Gives up interesting mathematical property of off-line key gen, which is operationally of mixed security

Advantages of this system

- Interoperates with existing systems
 - These keys wrapped as X.509, OpenPGP can be used with *any* other cryptosystem
- Can mix IBE and non-IBE components
 - Directory software can constrain name explosion
 - Can authenticate or limit discovery of public keys
 - Can work with existing revocation systems
 - Existing cryptosystems need not know they're working with IBE
 - Can work within political / legal constraints on key handling

Advantages (cont'd)

- Can rely on security strength of underlying components
 - HMAC has security proofs as pseudo-random function
 - Crypto strength relies on cryptosystem security proofs
- Works with any existing public key cryptosystem
 - RSA, DSS, DH, EC, etc.
 - All that is needed is PRNG-seeded key generator
- Allows speculative encryption before ownership is even known
 - This is the true need that IBE fills, replicable data sealing to a given identity

Unsolved Problems

- Key Escrow
 - Existence of master secret is operationally most risky form of escrow
 - Secure hardware is the best, possibly only solution
- Does this remove certificates?
 - If you're going to append a qualifier to an identity, that's a simple certificate
 - Could be mixed with lightweight cert systems like SPKI for interesting systems
- Proving ownership of a name
 - Easy to do in some cases -- proving a user name to a RADIUS, Kerberos server
 - Hard to do in abstract
 - Proof of unique ownership may require unique naming

Summary

- Software Engineering solution to Shamir's IBE question
 - Security limitations and concerns are same as other IBE systems
- Creates a function that uniquely, deterministically creates public key pairs from identities
- Trades off interesting mathematical properties of other systems for interesting operational properties
- Gives existing, well-used PKIs benefits of Identity-Based Enrollment
- Does not require new software infrastructure
- Hybrid approach suited to real-world political and legal constraints

Questions?

