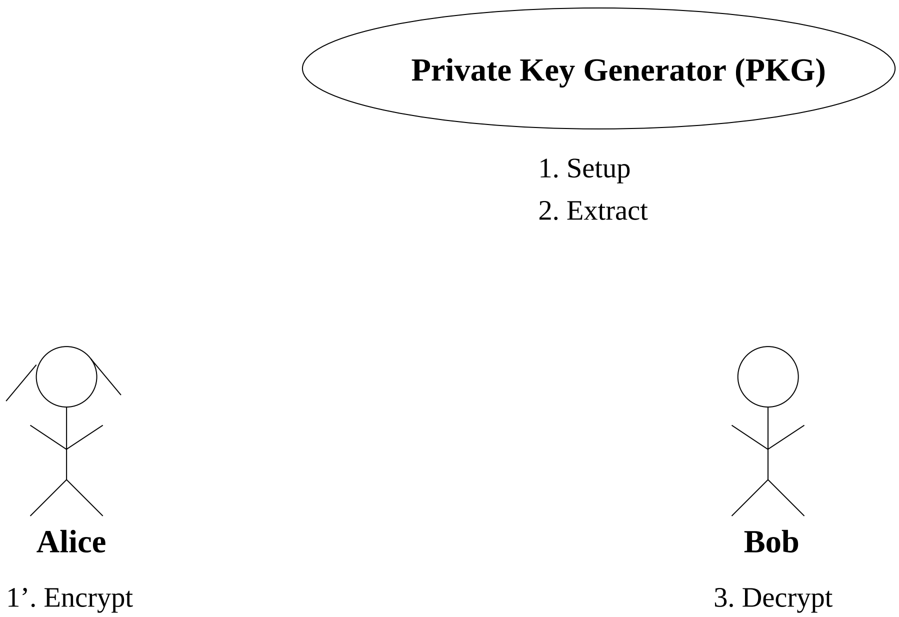
Notes on *Identity-Based Encryption from the Weil Pairing*

declarke@mit.edu



- **1. Setup**: $f(k, \mathcal{M}, \mathcal{C}) \rightarrow$ `params, master-key`
  k is a security parameter
  $\mathcal{M}$ is a description of a finite message space
  $\mathcal{C}$ is a description of a finite ciphertext space
  `params` is publicly known
  `master-key` is known only to the "Private Key Generator" (PKG)
- **2. Extract**: $f($`params, master-key, ID`$) \rightarrow d$
  The Extract algorithm extracts a private key from the given public key.
  `ID` is the string that will be used as a public key.
  d is the corresponding private decryption key.
- **1'. Encrypt**: $f(\texttt{params, ID}, M) \rightarrow C$
  $M \in \mathcal{M}$
  $C \in \mathcal{C}$
- **3. Decrypt**: $f(\texttt{params, ID}, C, d) \rightarrow M$
  $M \in \mathcal{M}$
  $C \in \mathcal{C}$

$\forall M \in \mathcal{M} : \texttt{Decrypt}(\texttt{params, ID}, C, d) = M$ where $C = \texttt{Encrypt}(\texttt{params, ID}, M)$

Advantages

- Public keys already known and do not need to be distributed. Alice wants to send encrypted mail to Bob at `bob@hotmail.com`, she simply encrypts her message using the public key string "`bob@hotmail.com`".
- Certificate management easy cos do not have to be creating many certificates per public key.
  (dec: Alice still needs to use the correct params, though if she uses the wrong params, you do not have a man-in-the-middle attack ...)
- Alice can send encrypted email to Bob even if Bob has not yet setup his public key certificate.
- Key escrow is inherent because PKG has private key.
- Key revocation is easy if Alice uses "bob@hotmail.com || current-year"as the public key. Bob can use his private key during the current year only. Once a year Bob needs to obtain a new private key from the PKG.
  Section 1.1.1.
  (dec: compare w/ SPKI/SDSI short validation periods.)
- Delegation
  Section 1.1.2
  Bob plays the role of PKG.
  Bob has to get a certificate from a CA binding his `params` to his name.

  1. Alice uses the `params` in Bob's certificate, and the current date, to encrypt her message.
  2. Since Bob has the `master-key`, he can extract the private key corresponding the current date, and decrypt the message.
  3. Bob simply installs on his laptop the seven private keys corresponding to the seven days of his trip. If laptop is stolen, the `master-key` is unharmed.

  1. Use the assistant's responsibility as public key, and to create a specific private key. Assistant cannot decrypt messages intended for other assistants. Alice uses the same `params`, though different `IDs` (for each responsibility) to encrypt messages.
- "Using standard techniques from threshold cryptography, the PKG can be distributed so that the master-key is never available in a single location" [Page 2][Page 11]
- Their scheme is semantically secure against an adaptive chosen ciphertext attack in the random oracle model. [Page 4]
- IBE scheme described is for encryption. Could also use it for digital signatures.
  Private key for the signature is the `master-key` for the IBE scheme.
  Public key is `params`.
  Signature on M is the IBE decryption key, d, for $\texttt{ID} = M$.
  To verify a signature, choose a random message M', encrypt M' using $\texttt{ID} = M$, then attempt to decrypt using d. If you decrypt successfully, it is shown that the sender has possession of `master-key`.
  (dec: seems like params still need to be distributed in a CA's certificate, because master-key is not under the control of some PKG, unless you want to trust a PKG even when you are doing digital signatures.)
- Instead of distributing copies of one public key to many Alices, one private key is sent from PKG to Bob.

Others

- Bob needs to authenticate himself to the PKG in the same way he would authenticate himself to a CA to obtain his private key from the PKG.
- Private keys need to be distributed, though they just have to be distributed to one principal. Still goes against what Diffie-Hellman described in '76: that a private key is generated by one principal, known only to that one principal, and never distributed.
- If just using one PKG, that PKG has the private keys of everyone. Also, that PKG is a single point of failure.