# Public CCA-Security

Cryptography and Protocols
Andrei Bulatov

## Login Problem Revisited

- The login problem

  Suppose that a server and a client share a secret PIN, $I$, that was chosen at random $0 \le I \le 10^4$ (13 bits)

  They also share a secret key $k$

  Protocol:

  the client sends encrypted $I$

  the server decrypts and checks if the PIN is correct

  if PIN is incorrect the server aborts the communication

- Public key crypto does not solve problems with this protocol
- Moreover, even digital signatures do not quite solve it, as they still require to remember private and public keys

# CCA Security

- Let $(K, E, D)$ be an asymmetric encryption scheme and $(T, \varepsilon)$ a superpolynomial pair. Consider the following game:

  (1) $K$ generates a pair of keys $(e, d)$

  (2) Eve gets input $e$

  (2) Eve gets access to the black box $D_d(\cdot)$

  (3) Eve chooses $P_1$ and $P_2$

  (4) Alice chooses $i \in \{0,1\}$ at random and gives Eve $C = E_e(P_i)$

  (5) Eve gets more access to the black box $D'_d(\cdot)$

$$D'_d(C') = \begin{cases} D_d(C'), & \text{if } C' \neq C \\ \perp, & \text{if } C' = C \end{cases}$$

  (6) Eve outputs $j \in \{0,1\}$

# CCA Security (cntd)

- Eve wins if $j = i$
- Scheme $(K, E, D)$ is $(T, \varepsilon)$-CCA-secure if for any Eve of time complexity at most $T$ $\Pr[\text{Eve wins}] < \frac{1}{2} + \varepsilon$

- Example:

  `Pure' Rabin or RSA schemes are not CCA-secure

# A CPA-Secure Scheme

- We will not be able to define a CCA-secure scheme in this course, although one exists (it uses zero-knowledge)
- Instead we define such a scheme in the random oracle model. That is we assume that we have access to a public truly random function
- In practice we then use a PRF instead of a random oracle. However, our usual proofs for PRFs do not work in this case

# A CPA-Secure Scheme (cntd)

- Scheme:

  - Let $G: \{0,1\}^n \to \{0,1\}^n$ be a random oracle, and $\{f, f^{-1}\}$ be a collection of trapdoor permutations.

  - The public key is $f$, the private key is $f^{-1}$

  - To encrypt $P \in \{0,1\}^n$ choose random $r \in \{0,1\}^n$ and compute $f(r)$ and $G(r) \oplus P$

  - To decrypt $C, C'$ compute $r = f^{-1}(C)$ and let $P = G(r) \oplus C'$

- **Theorem**

  This scheme is CPA-secure.

# A CPA-Secure Scheme: Proof

- **Proof**

  Note that Eve has access to $G(\cdot)$

  Let Eve as a challenge get $C^*, C'^*$ where $C^* = f(r^*)$ and $G'^* = G(r^*) \oplus P_i$

  **Claim**.

  The probability that Eve queries $r^*$ to $G$ is negligible

  Indeed, $G(r^*)$ is just a random string, and if Eve guesses $r^*$ using $f(r^*)$ she can invert a trapdoor permutation

  Thus, for Eve $G(r^*) \oplus P_i$ and $u \oplus P_i$, $u$ random, are indistinguishable. Moreover $u \oplus P_i$ is uniform

  Therefore $\Pr[\text{Eve wins}] < \frac{1}{2} + \varepsilon$

# A CCA-Secure Scheme

- Along with $G$ we need another random oracle
- **Scheme**

  - Let $G: \{0,1\}^n \rightarrow \{0,1\}^n$ and $H: \{0,1\}^{2n} \rightarrow \{0,1\}^n$ be two random oracles, and $\{f, f^{-1}\}$ be a collection of trapdoor permutations.

  - The public key is $f$, the private key is $f^{-1}$

  - To encrypt $P \in \{0,1\}^n$ choose random $r \in \{0,1\}^n$ and compute $f(r), G(r) \oplus P$ and $H(P,r)$

  - To decrypt $C, C', C''$ compute $r = f^{-1}(C)$ and let $P = G(r) \oplus C'$. If $H(P,r) = C''$ return $P$, otherwise $\perp$

# A CCA-Secure Scheme: Theorem

- **Theorem**

  The above scheme is CCA-secure.

- **Proof**

  Let Eve as a challenge get $C^*, C'^*, C''^*$, where $C^* = f(r^*)$
$C'^* = G(r^*) \oplus P_i$, $C''^* = H(P_i, r^*)$

  We are going to show that Eve cannot get advantage of decryption queries. Therefore, the scheme is CCA-secure if it is CPA-secure, and that we already know.

  Since $H$ is truly random, no one can guess (with only negligible probability) two pairs $P, r$ and $P', r'$ such that $H(P,r) = H(P', r')$, but $P, r \neq P', r'$

# A CCA-Secure Scheme: Theorem (cntd)

- At each step $j$ of the attack, and every string $w \in \{0,1\}^n$ we define $H_j^{-1}(w)$ as follows:

  if $H$ has been queried before about $P, r$ such that $H(P,r) = w$ then set $H_j^{-1}(w) = P, r$

  otherwise $H_j^{-1}(w) = \perp$

  Now we try to simulate the decryption box:

  when queried $C, C', C''$ if $H_j^{-1}(C'') = P, r$ (that determines $C, C'$ uniquely) output $P$, otherwise output $\perp$

# A CCA-Secure Scheme: Theorem (cntd)

- **Claim**

  Eve is unable to tell apart the real and the modified protocols

  Indeed, to detect the difference Eve must come up with $C, C', C''$ such that

  - $C'' \neq C''^*$ since if $H_i^{-1}(C'') = P^*, r^*$ then Eve either breaks $H$, or both protocols return $\perp$, or she asked the disallowed query $C^*, C'^*, C''^*$

  - $C''$ was not returned as the answer by a previous query; thus Eve breaks $H$

  - If $P, r$ are the values determined by $C, C'$ then $H(P, r) = C''$. As $P, r$ have not been asked before, the probability of that is $2^{-n}$