How to use untrusty cryptographic devices

Daniel Kucner

Institute of Computer Science

University of Wrocław

Mirosław Kutyłowski

Institute of Mathematics

Wrocław University of Technology

and CC Signet

Black-Box device

the following data is available for a black-box device:

- specification of a protocol implemented,
- some quality certificates (according to Common Criteria, FIPS, ...)

Black-Box device

the following data is available for a black-box device:

- specification of a protocol implemented,
- some quality certificates (according to Common Criteria, FIPS, ...)

Advantages:

- unchangeable (no viruses, no malicious changes)
- safer and faster then software

Black-Box device

the following data is available for a black-box device:

- specification of a protocol implemented,
- some quality certificates (according to Common Criteria, FIPS, ...)

Advantages:

- unchangeable (no viruses, no malicious changes)
- safer and faster then software

Disadvantages:

- a real black-box – impossible to verify

How do we know that a device is honest?

- verification is extremely complex
- certification authorities need to be trusted
- produced by a foreign manufacturer (under control of a foreign secret service?)

How do we know that a device is honest?

- verification is extremely complex
- certification authorities need to be trusted
- produced by a foreign manufacturer (under control of a foreign secret service?)

the danger is real – kleptography techniques

Diffie-Hellman key exchange

Alice	**Bob**
generate random a	generate random b
$x \leftarrow g^a \bmod p$	$y \leftarrow g^b \bmod p$
send x to Bob	send y to Alice
$k \leftarrow y^a \bmod p$	$k \leftarrow x^b \bmod p$

Kleptography - device (DH)

$(X, Y = \alpha^X \bmod p)$ – adversary's keys.

Device

1. generate random $c_1 \in \mathbb{Z}_{p-1}$
2. return $m_1 = \alpha^{c_1} \bmod p$
3. $z := m_1 \cdot Y^{c_1} \bmod p$
4. return $m_2 = \alpha^{H(z)} \bmod p$

Kleptography - device (DH)

$(X, Y = \alpha^X \bmod p)$ – adversary's keys.

Device

1. generate random $c_1 \in \mathbb{Z}_{p-1}$
2. return $m_1 = \alpha^{c_1} \bmod p$
3. $z := m_1 \cdot Y^{c_1} \bmod p$
4. return $m_2 = \alpha^{H(z)} \bmod p$

Attack

1. Adversary eavesdrops m_1, m_2
2. $z := m_1 \cdot m_1^X \bmod p$
3. if $m_2 := \alpha^{H(z)} \bmod p$

   then return $H(z)$

Kleptography - detection

Different number of exponentiation changes stochastic characteristic of computation time

DH clear device

generate random $c_1 \in \mathbb{Z}_{p-1}$

$m_1 = \alpha^{c_1} \mod p$

DH contaminated device

generate random $t \in \{0, 1\}$

$z := \alpha^{c_1 - Wt} \cdot Y^{-ac_1 - b} \mod p$

$c_2 := H(z)$, $m_2 = \alpha^{c_2} \mod p$

Idea of solution

- combine two or more devices of different manufacturers
- even if each of them is contaminated, the result should be secure

Secure DH with contaminated devices

1. $x_1 \leftarrow \alpha^{k_1} \bmod p$ using D_1
2. $x_2 \leftarrow \alpha^{k_2} \bmod p$ using D_2

Secure DH with contaminated devices

1. $x_1 \leftarrow \alpha^{k_1} \bmod p$ using D_1
2. $x_2 \leftarrow \alpha^{k_2} \bmod p$ using D_2
3. send $x \leftarrow x_1 x_2 \bmod p$ to Bob
4. get y from Bob

Secure DH with contaminated devices

1. $x_1 \leftarrow \alpha^{k_1} \bmod p$ using D_1
2. $x_2 \leftarrow \alpha^{k_2} \bmod p$ using D_2
3. send $x \leftarrow x_1 x_2 \bmod p$ to Bob
4. get y from Bob
5. $z_1 \leftarrow y^{k_1} \bmod p$ using D_1
6. $z_2 \leftarrow y^{k_2} \bmod p$ using D_2
7. $z \leftarrow z_1 z_2 \bmod p$

Proof of SDH security - outline

- if one device is secure then whole is secure
- otherwise adversary has to solve problem:

given $w = u \cdot v \bmod p$

find $r = u + v \bmod p$

Another secure DH ?

1. set in D_1 a generator $\alpha_1 = \alpha$
2. compute $x_1 \leftarrow \alpha^{k_1}$ using D_1

Another secure DH ?

1. set in D_1 a generator $\alpha_1 = \alpha$
2. compute $x_1 \leftarrow \alpha^{k_1}$ using D_1
3. set in D_2 a generator $\alpha_2 = x_1$
4. compute $x_2 \leftarrow \alpha_2^{k_2}$ using D_2
5. send x_2 to the partner and obtain y

Another secure DH ?

1. set in D_1 a generator $\alpha_1 = \alpha$
2. compute $x_1 \leftarrow \alpha^{k_1}$ using D_1
3. set in D_2 a generator $\alpha_2 = x_1$
4. compute $x_2 \leftarrow \alpha_2^{k_2}$ using D_2
5. send x_2 to the partner and obtain y
6. put y into D_2 and compute $y_2 \leftarrow y^{k_2}$
7. put y_2 into D_1 and compute the key $y \leftarrow y_2^{k_1}$

Another secure DH ?

1. set in D_1 a generator $\alpha_1 = \alpha$
2. compute $x_1 \leftarrow \alpha^{k_1}$ using D_1
3. set in D_2 a generator $\alpha_2 = x_1$
4. compute $x_2 \leftarrow \alpha_2^{k_2}$ using D_2
5. send x_2 to the partner and obtain y
6. put y into D_2 and compute $y_2 \leftarrow y^{k_2}$
7. put y_2 into D_1 and compute the key $y \leftarrow y_2^{k_1}$

$$\mathbf{y = y_2^{k_1} = y^{k_1 \cdot k_2}}$$

Attack on (in)secure DH

$x_2^{1)}, x_2^{2)}, x_2^{3)}$ – observable

$$x_2^{1)} = (x_1^{1)})^{k_2^{1)}}$$
$$x_2^{2)} = (x_1^{2)})^{k_2^{2)}} = (x_1^{2)})^{x_2^{1)}}$$
$$x_2^{3)} = (x_1^{3)})^{k_2^{3)}} = (x_1^{3)})^{x_2^{2)}}$$

then

$$x_1^{2)} = (x_2^{2)})^{f_1} \bmod p$$
$$x_1^{3)} = (x_2^{3)})^{f_2} \bmod p$$

where $f_i = (x_2^{i)})^{-1} \bmod p - 1$

iterate:

$$x_1^{3)} = \alpha^{x_1^{2)}}$$
$$x_1^{2)} \cdot x_2^{2)} \bmod p - 1$$

1. set in D_1 a generator $\alpha_1 = \alpha$
2. compute $x_1 \leftarrow \alpha^{k_1}$ using D_1.
3. set in D_2 a generator $\alpha_2 = x_1$
4. compute $x_2 \leftarrow \alpha_2^{k_2}$ using D_2.
5. send x_2 to the partner and obtain y
6. put y into D_2 and compute $y_2 \leftarrow y^{k_2}$
7. put y_2 into D_1 and compute $y \leftarrow y_2^{k_1}$

ElGamal Encryption

1. pick a random $k \quad : \; 0 < k < p - 1$
2. compute $r \leftarrow \alpha^k \bmod p$
3. compute $s \leftarrow m \cdot y^k \bmod p$

Secure ElGamal Encryption (SEGE 1)

1. compute ciphertext (r_1, s_1) using device D
2. compute ciphertext (r_2, s_2) of message 1 *(on PC)*
3. $r \leftarrow r_1 \cdot r_2 \bmod p$ *(on PC)*
4. $s \leftarrow s_1 \cdot s_2 \bmod p$ *(on PC)*
5. return ciphertext (r, s)

Secure ElGamal Encryption (SEGE 2)

1. find m_1, m_2 so that $m \equiv m_1 \cdot m_2 \bmod p$
2. $(r_1, s_1) \leftarrow Enc_{D_1}(m_1)$
3. $(r_2, s_2) \leftarrow Enc_{D_2}(m_2)$
4. $r \leftarrow r_1 \cdot r_2 \bmod p$ *(on PC)*
5. $s \leftarrow s_1 \cdot s_2 \bmod p$ *(on PC)*
6. return ciphertext (r, s)

Secure ElGamal Encryption (SEGE 2)

1. find m_1, m_2 so that $m \equiv m_1 \cdot m_2 \bmod p$
2. $(r_1, s_1) \leftarrow Enc_{D_1}(m_1)$
3. $(r_2, s_2) \leftarrow Enc_{D_2}(m_2)$
4. $r \leftarrow r_1 \cdot r_2 \bmod p$ *(on PC)*
5. $s \leftarrow s_1 \cdot s_2 \bmod p$ *(on PC)*
6. return ciphertext (r, s)

$$r = r_1 \cdot r_2 = \alpha^{k_1+k_2}$$

$$s = s_1 \cdot s_2 = m_1 \cdot y^{k_1} \cdot m_2 \cdot y^{k_2} = m \cdot y^{k_1+k_2}$$

Secure ElGamal Encryption (SEGE 3)

1. find $m_1, m_2 \ : \ m \equiv m_1 \cdot m_2 \bmod p$
2. $(r_1, s_1) \leftarrow Enc_{D_1}(m_1)$

Secure ElGamal Encryption (SEGE 3)

1. find $m_1, m_2 \ : \ m \equiv m_1 \cdot m_2 \bmod p$
2. $(r_1, s_1) \leftarrow Enc_{D_1}(m_1)$
3. D_2 computes (r_2, s_2), a ciphertext of 1
4. set α of D_3 to r_2
5. set public key of D_3 to s_2
6. $(r_3, s_3) \leftarrow Enc_{D_3}(m_2)$

Secure ElGamal Encryption (SEGE 3)

1. find $m_1, m_2 \;:\; m \equiv m_1 \cdot m_2 \bmod p$
2. $(r_1, s_1) \leftarrow Enc_{D_1}(m_1)$
3. D_2 computes (r_2, s_2), a ciphertext of 1
4. set α of D_3 to r_2
5. set public key of D_3 to s_2
6. $(r_3, s_3) \leftarrow Enc_{D_3}(m_2)$
7. $r \leftarrow r_1 \cdot r_3 \bmod p$
8. $s \leftarrow s_1 \cdot s_3 \bmod p$
9. return ciphertext (r, s)

Secure ElGamal Encryption (SEGE 3)

1. find $m_1, m_2 \ : \ m \equiv m_1 \cdot m_2 \bmod p$
2. $(r_1, s_1) \leftarrow Enc_{D_1}(m_1)$
3. D_2 computes (r_2, s_2), a ciphertext of 1
4. set α of D_3 to r_2
5. set public key of D_3 to s_2
6. $(r_3, s_3) \leftarrow Enc_{D_3}(m_2)$
7. $r \leftarrow r_1 \cdot r_3 \bmod p$
8. $s \leftarrow s_1 \cdot s_3 \bmod p$
9. return ciphertext (r, s)

$$r = r_1 \cdot r_3 = \alpha^{k_1} \cdot r_2^{k_3} = \alpha^{k_1 + k_2 \cdot k_3}$$

$$s = m_1 \cdot y^{k_1} \cdot m_2 \cdot s_2^{k_3} = m_1 \cdot y^{k_1} \cdot m_2 \cdot y^{k_2 \cdot k_3} = m \cdot y^{k_1 + k_2 \cdot k_3}$$

How to get product of exponents?

- if both devices have the same parameters p, α, y, then DH could be broken
- both devices have the same p - as above
- devices have different p - no general algorithm, perhaps special p, p_1, p_2 exist such that for random $x_1 = \alpha_1^{k_1} \bmod p_1$ and $x_2 = \alpha_2^{k_2} \bmod p_2$ we could compute $x = \alpha^{k_1 \cdot k_2}$?

ElGamal Signature Protocol

Sign a message m:

1. compute a random $k \quad (1 \leq k \leq p-1)$
2. $r \leftarrow \alpha^k \bmod p$
3. $s \leftarrow k^{-1}(H(m) - a \cdot r) \bmod p - 1$
4. output the signature $S(m) = (r, s)$

Secure ElGamal Signature

1. Alice sends arbitrary hash h to D_1
2. D_1 generates (r_1, s_1) for parameters p, α, u (random private key)
3. Alice computes k_1 from s_1, r_1, u and h *(on PC)*
4. Alice sets generator of D_2 to r_1
5. D_2 generates (r_2, s_2) for message m
6. $(r, s) = (r_2, s_2/k_1 \bmod p - 1)$ for parameters p, α, x

Conclusions

We have shown how to use devices for

- Diffie-Hellman
- ElGamal Encryption
- ElGamal Signature

to keep safe even if devices are contaminated.

Problems

- what about systems without random numbers? for splitting the secret!
- RSA – well known: split d into $d_1 + d_2$
- could we construct such a protocol for Rabin encryption, signature?