One “Short” Signature Scheme’s Security Properties

Anton Guselev

Technical Committee on Standardisation
“Cryptographic Information Security”
— (TC 026) —

– CTCrypt 2021 –

ElGamal Digital Signature Scheme

User has **private key** d, **public key** $Q = dP$, P is a point of elliptic curve given over $\mathbb{Z}_p$.

Sign algorithm

Input: (d, Msg)

Output: signature $\bar{r} \| \bar{s}$

1: $e = \mathsf{H}(\mathrm{Msg})$
2: $k \leftarrow \mathbb{Z}_q$
3: $r = x_{kP} \pmod{q}$
4: $s = ke + rd$
5: **return** $\bar{r} \| \bar{s}$

Verification algorithm

Input: $(Q, \mathrm{Msg}, \bar{r} \| \bar{s})$

Output: Verfy info

1: $e = \mathsf{H}(\mathrm{Msg})$
2: $R = e^{-1}sP - e^{-1}rQ$
3: **if** $x_R \neq r$ **then**
4: The signature is **false**
5: **else**
6: The signature is **correct**

- Bit length of the signature is $2n$

The Ways to Shorten ElGamal – One-way function

$f : \mathbb{Z}_p \to \mathbb{Z}_q^*$ – one way function, $f(r)$ has only $b < n$ significant bits for any r.

Sign algorithm

1: $e = \mathrm{H}(\mathrm{Msg})$
2: $k \leftarrow \mathbb{Z}_q$.
3: $\boxed{r = f(x_{kP} \pmod q)}$
4: $s = ke + rd$
5: **return** $\bar{r} \| \bar{s}$

Verification algorithm

1: $e = \mathrm{H}(\mathrm{Msg})$
2: $R = e^{-1}sP - e^{-1}rQ$
3: **if** $f(x_R) \neq r$ **then**
4: The signature is **false**
5: **else**
6: The signature is **correct**

Feature

- Bit length of the signature is $n + b$

The Ways to Shorten ElGamal – Bit-fixation

$\bar{r} = \bar{r}^* \| const$, where $|const| = l$

Sign algorithm

1: $e = \mathrm{H}(\mathrm{Msg})$
2: $k \leftarrow \mathbb{Z}_q$
3: $r = x_{kP} \pmod{q}$
4: **if** $\boxed{\mathrm{LSB}_l(\bar{r}) \neq const}$ **then**
5: **goto** step 2
6: **else**
7: $s = ke + rd$
8: $\boxed{\bar{r}^* = \mathrm{MSB}_{n-l}(\bar{r})}$
9: **return** $\bar{r}^* \| \bar{s}$

Verification algorithm

1: $e = \mathrm{H}(\mathrm{Msg})$
2: $\bar{r} = \bar{r}^* \| const$
3: $R = e^{-1}sP - e^{-1}rQ$
4: **if** $x_R \neq r$ **then**
5: The signature is **false**
6: **else**
7: The signature is **correct**

Features

- Additional time complexity
- Bit length of the signature is $2n - l$

The Ways to Shorten ElGamal – Bit-cutting

Cut $t < n$ bits of s (or r)

Sign algorithm

1: $e = \mathsf{H}(\mathrm{Msg})$
2: $k \leftarrow \mathbb{Z}_q$
3: $r = x_{kP}$
4: $s = ke + rd$
5: $\boxed{\bar{s}^* = \mathsf{MSB}_{n-t}(\bar{s})}$
6: **return** $\bar{r} \| \bar{s}^*$

Verification algorithm

1: $e = \mathsf{H}(\mathrm{Msg})$
2: $cnt = cnt + 1.$
3: **if** $cnt \geq 2^t$ **then**
4: The signature is **false**
5: **else**
6: $\bar{s} \leftarrow \bar{s}^* \| \overline{cnt}$
7: $R \leftarrow e^{-1}sP - e^{-1}rQ$
8: **if** $x_R = r$ **then**
9: The signature is **correct**

Features

- Additional time complexity
- Bit length of the signature is $2n - t$

Finale Scheme – Bring them all

Shortest signature – **three approaches** are combined

Advantages

- Signature length is $n + b - t - l$
- "Classical" cryptoanalysis is not applicable

Disadvantages

- Additional calculations to sign and verify
- Specific attacks may arise

"Short" signature security characteristics are justified in the original paper presented at CTCrypt 2020.

- Security of the scheme based upon provable security point of view
- Scheme is secure if the discrete logarithm problem is hard
- Forking lemma is a cornerstone
- *In case* $b = 100$, $t = 18$, $l = 18$ *and* $n = 256$ *the length of signature is* 320 *bit*, the advantage of the adversary to forge the signature does not exceed 2^{-35}

  But some times it is hard to understand whether this is enough or not...

Attacks based on collision for H or f will not lead to success. But ...
what about (second) preimage for f?

If user signed at least one message and proper preimage is found then forged signature for any message Msg_1 may be calculated as follows:

1: $e = \mathsf{H}(\mathrm{Msg})$
2: $e_1 = \mathsf{H}(\mathrm{Msg}_1)$
3: $T = e^{-1}sP - e^{-1}rQ$
4: $i = 0$
5: **while** $f(x_A) \neq r$ **do**
6: $\quad i = i + 1$
7: $\quad A = e_1^{-1}eT + i(e_1^{-1}e)P$
8: $s_1 \leftarrow s + ie$
9: **return** $\bar{r} \| \bar{s}_1$

Signature $\bar{r}\|\bar{s}_1$ is a valid signature for Msg_1 as far as the equality $f(x_R) = r$ holds

$$R = e_1^{-1}s_1P - e_1^{-1}rQ = e_1^{-1}(e(k+i)+rd)P - e_1^{-1}rdP = e_1^{-1}ekP + e_1^{-1}eiP = A$$

and given the equation $f(x_A) = r$, we obtain that $f(x_R) = r$.

Estimates

Computation the second preimage in the case of $b = 100$ would require no more than 2^{100} additions and f applications to find second preimage and to mount the attack

If some bits of r are fixed with the constant it is possible to find values that are different from the one used during the sign stage, but applicable during verification procedure

Let's consider a signature $\bar{r}^* \| \bar{s}$, where $r = x_{kP}$, $\bar{r} = \bar{r}^* \| \underbrace{0 \dots 0}_{l}$ for "correct" k

If for any $0 < u < 2^l$ $x_{k'P} = r'$, $\bar{r}' = \bar{r}^* \| \bar{u}$, then the signature $(\bar{r}^* \| \bar{s})$ will be a valid signature for a given message

If second preimage is found and i is the value such that $f(x_{e_1^{-1}e(k+i)P}) = r'$, then forged signature for Msg_1 will be $(r^* \| s_1)$
Correctness of the forgery follows from the equation:

$$R = e_1^{-1}s_1P - e_1^{-1}rQ = e_1^{-1}(e(k+i) + rd)P - e_1^{-1}dP = A$$

Cardinality of the set of admissible values r suitable for the analysis increase from 1 to 2^l

Estimates

If $l = 18$ we will require no more than $2^{100-18} = 2^{82}$ additions and f applications to find second preimage and to mount the attack

The method succeeded in case the comparison of cut bits with given constant is not performed during the verification procedure

If some bits of s are cut it is possible to compute values r_j that are "equivalent" to the "correct" value r

Any second preimage that was found for r_j may be useful to calculate forgery

Cardinality of the set of admissible values r suitable for the analysis increase from 1 to 2^t

Estimates

If $t = 18$ we will require no more than $2^{100-18} = 2^{82}$ calculations of additions and f applications to find second preimage and to mount the attack

Conclusion

- For a given set of parameters "short" signature scheme is secure (in particular model) if length of the private key and the signature are 256 and 320 bits, also additional calculation should be made during the sign and verification
- Application of the overall attack to "short" signature scheme will require about 2^{81} calculations of additions and f applications
- By rough estimates "short" signature scheme security level is equal to 81 bits
- In classical ElGamal framework, it is possible to use a private key of 162 bits to achive such security level and the signature length will be 324 bits.