T-79.4501 Cryptography and Data Security

Addendum

- Relative key lengths
- Searches, workloads and success probabilities

Relative key lengths

Source: S. Blake-Wilson et al, RFC 3278: Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS), (based on A. Lenstra and E. Verheul (J.Crypt 1999)

Valid until	Symmetric algorithms	Elliptic cryp-tosystems	DH/DSA/RSA
2010	80	163	1024
2030	112	233	2048
2045	128	283	3072
?	192	409	7680
?	256	571	15360

Search workloads and success probabilities

- Exhaustive search
- Preimage search
- Collision search for one function
- Collision search for two functions

Exhaustive key search

- Searching for a secret value used in a cryptosystem: keys, passkeys, etc in a set of size N. E.g., $N = 2^L$, where L is the key length in bits.
- Test based on given input and output; workload is measured in the number of tests to be performed
- Sometimes called as *Dictionary Attack* when the test results are precomputed for all values of the searched parameter
- We assume uniform distribution
- Search over the entire set of size N, then success probability p =1, average workload $w = N/2$ trials
- Success probability p, that is, search is over a set of size Np, average workload:

$$w = p(Np/2) + (1-p)Np = Np - \frac{1}{2} Np^2 .$$

Pre-image search

- One-way hash function H, modelled as a ”random oracle”: given input x the output $y = H(x)$ is picked uniformly at random
- Number of possible outputs N
- Search problem: given y find x such that $y = H(x)$
- After k trials the success probability:

$$p = 1 - (1 - 1/N)^k = 1 - ((1 - 1/N)^N)^{k/N}$$

$$\approx 1 - e^{-k/N} > 1/2 \text{ , for } k > N \ln 2 \approx 0.693N$$

Collision search for the same function

- One-way hash function *H*, modelled as a ”random oracle”: given input *x* the output $y = H(x)$ is picked uniformly at random
- Number of possible outputs *N*
- Search problem: Find x_1 and x_2 such that $H(x_1) = H(x_2)$
- After $H(x)$ has been computed for k values of x the probability p that some value H(x) has appeared at least twice is (see Lecture 2):

$$p \approx \frac{1}{2} = e^{-\ln 2} \quad \text{for} \quad k \approx \sqrt{2N \ln 2} \approx 1.17\sqrt{N}$$

Collision search for two different functions

- Two one-way hash functions H_1 and H_2 with the same target set modelled as "random oracles": given input x the outputs $y_1 = H_1(x)$ and $y_2 = H_1(x)$ are picked uniformly at random
- Number of possible outputs N for both functions
- Search problem: Find x_1 and x_2 such that $H_1(x_1) = H_2(x_2)$.
- Create two sets:

$$A_1 = \{H_1(x) \mid x\} \text{ and } A_2 = \{H_2(y) \mid y\}$$

- Assume (for simplicity) that A_1 has k different elements, and in A_2 the values have been computed for k different y.
- Then the probability p that the sets have at least one element in common is (see Stallings, Appendix 11A and HW5, Problem 2b)

$$p \approx \frac{1}{2} \quad \text{for} \quad k \approx 0.87\sqrt{N}$$