Part II
Contributions

3 Our contributions

Contents

3.1 Overview

In this section we list our main contributions. First we present applications for collision attacks that we have successfully implemented. This is followed by our main results for MD5 and SHA-1. Next, we present more general results that apply to certain classes of hash functions. Finally, we outline the remainder of this thesis.

3.2 Chosen-prefix collision applications

Several abuse scenarios were presented based on Wang et al.'s collision attack, such as to mislead integrity checking software [Kam04, Mik04], constructing different PostScript documents [DL05] that collide under MD5, constructing two different *X.509 certificates* [CSF+08] with identical Distinguished Names and identical MD5-based digital signatures but different public keys [LdW05], etc. Although none of these abuse scenarios spoke in favor of continued usage of MD5, the abuse potential of Wang et al.'s identical-prefix collision attack remains limited.

The most important contribution of this thesis is the construction and application of *chosen-prefix collision* attacks. This removes the $IHV_{\text{in}} = IHV'_{\text{in}}$ restriction of identical-prefix collisions: given any two chosen message prefixes P and P', a chosen-prefix collision attack constructs suffixes S and S' such that the concatenated values $P||S$ and $P'||S'$ form a collision.

Using our chosen-prefix collision attack against MD5 we discuss and implement several new abuse scenarios in Chapter 4 that are impossible using identical-prefix collisions. The main abuse scenarios that we present and successfully have implemented are:

- Construction of colliding X.509 certificates with different distinguished names;
- Construction of colliding documents or executables where the malicious payload is not present in the harmless counterpart;

- Finally, our most convincing abuse scenario is the construction of a rogue CA (*Certification Authority*) X.509 certificate that undermines the core of the X.509 public key infrastructure most commonly known for its use to secure websites (`https://`);

3.3 Results on MD5

For MD5 we present a practical chosen-prefix collision attack and an identical-prefix collision attack which is faster than Wang et al.'s collision attack:

Theorem 3.1 (MD5 collision attacks)**.** *There exist an identical-prefix collision attack and a chosen-prefix collision attack against MD5 with average complexities equivalent to* 2^{16} *and* 2^{39} *calls to the compression function of MD5, respectively.*

Furthermore, there exists a near-collision attack against the compression function of MD5, with an average complexity equivalent to $2^{14.8}$ *calls to the compression function of MD5* [10]*, that for given* $IHV_{in} = IHV'_{in}$ *searches for a pair of message blocks that results in*

$$\delta IHV_{out} = (-2^5, -2^5 + 2^{25} + 2^{23} + 2^{26} - 2^{14}, -2^5 + 2^{25} + 2^{23}, -2^5 + 2^{25}).$$

This near-collision attack can be altered to work for any given IHV_{in} *and* IHV'_{in}*. The resulting near-collision attack has the same runtime complexity and searches for a pair of message blocks that result in*

$$\delta IHV_{out} = \delta IHV_{in} + (-2^5, -2^5 + 2^{25} + 2^{23} + 2^{26} - 2^{14}, -2^5 + 2^{25} + 2^{23}, -2^5 + 2^{25}).$$

The proof by construction of Theorem 3.1 is given in Chapter 6.

Lastly, we present a technique that can detect all known feasible identical-prefix and chosen-prefix collision attacks against MD5 given only one message from a colliding message pair:

Theorem 3.2 (Detecting MD5 collision attacks)**.** *Given a message* M*, it is possible to detect, with an average complexity of about* 224 *times the complexity of computing the MD5 hash of* M *and error probability approximately* 2^{-128}*, whether* M *has been constructed by a collision attack based on message differences given in one of the following papers: [dBB93], [WY05], [SLdW07c], [XFL08], [XLF08], [VJBT08], [XF09], [SSA+09b] and [XF10]. Furthermore, future collision attacks can easily be added in the detection if they are not detected already.*

The proof by construction of Theorem 3.2 is given in Section 8.3.

3.4 Results on SHA-1

For SHA-1 we present a near-collision attack which is used to construct an identical-prefix and a chosen-prefix collision attack:

10. MD5's compression function requires 500 arithmetic and bitwise word operations.

Theorem 3.3 (SHA-1 collision attacks)**.** *There exists an identical-prefix collision attack against SHA-1 with an average complexity between* $2^{60.3}$ *and* $2^{65.3}$ *calls to the compression function of SHA-1. Also, there exists a chosen-prefix collision attack against SHA-1 with an average complexity of* $2^{77.1}$ *calls to the compression function of SHA-1.*

There exists a near-collision attack against the compression function of SHA-1 that, for given $IHV_{in} = IHV'_{in}$ *satisfying the bitconditions given in Table 7-6 (p. 169), searches for a pair of message blocks so that the resulting* δIHV_{out} *equals one of the 192 target* δIHV_{diff} *values given in Table 7-5 (p. 167). It has an average complexity equivalent to* $2^{57.5}$ *calls to the compression function* [11]*.*

The proof by construction of Theorem 3.3 is given in Chapter 7.

Our attacks against SHA-1 are based on *local collisions* that consist of a single disturbance in some step followed by corrections in the next five steps which cancel out this disturbance. *Disturbance vectors* mark these disturbances and thus the start of local collisions in patterns that are compatible with SHA-1's message expansion. In Section 7.5 we introduce a new precise SHA-1 disturbance vector analysis for the use of near-collision attacks with the following features that improve upon the literature:

- takes into account the dependence of local collision (so far independence has been assumed);
- allows to determine $\delta IHV_{\text{diff}} = \delta IHV_{\text{out}} - \delta IHV_{\text{in}}$ with higher success probabilities than those as prescribed by the disturbance vector;
- allows to determine the optimal message bitrelations that lead to the highest success probability (so far only treated for individual local collisions).

Using our new analysis we show in Section 7.5 that higher success probabilities can be obtained compared to the product of the success probabilities of the individual local collisions.

Furthermore, we present a technique that can detect possibly feasible identical-prefix and chosen-prefix collision attacks against SHA-1 given only one message from a colliding message pair:

Theorem 3.4 (Detecting SHA-1 collision attacks)**.** *Given a message* M*, it is possible to detect, with an average complexity of about* 15 *times the complexity of computing the SHA-1 hash of* M *and error probability approximately* 2^{-160}*, whether* M *is constructed by a possibly feasible collision attack against SHA-1. Furthermore, future collision attacks can easily be added in the detection if they are not detected already.*

The proof by construction of Theorem 3.4 is given in Section 8.4.

11. SHA-1's compression function requires 941 arithmetic and bitwise word operations.

3.5 General results

Our following theorem enables one to construct complete differential paths for a certain class of compression functions.

Theorem 3.5 (Differential path construction)**.** *For the family $\mathcal{F}_{md4cf}$ of compression functions defined in Section 5.3 that includes those of MD4, MD5 and SHA-1 there exists an algorithm which, from given IHV_{in}, IHV'_{in} and a partial differential path beginning at some early step and ending at the last step, constructs complete differential paths.*

Proof is given in Chapter 5 with specific more efficient algorithms for MD5 and SHA-1 presented in Section 6.2 and Section 7.4, respectively. These algorithms were key to the construction of the identical-prefix and chosen-prefix collision attacks against MD5 and SHA-1 mentioned in Theorems 3.1 and 3.3.

As a more general result we can construct chosen-prefix collisions for other non-collision-resistant hash functions:

Theorem 3.6 (Generic chosen-prefix collision attack)**.** *Let H be a hash function with L-bit hash values based on the Merkle-Damgård construction with compression function*

$$Compress : \{0,1\}^L \times \{0,1\}^N \to \{0,1\}^L, \quad 1 \leq L \leq N$$

that uses an IHV and message block of bit size L and N, respectively. Addition and subtraction of IHV-values are defined through some additive abelian group $(\{0,1\}^L, +)$. There exists a chosen-prefix collision attack against H that is faster than a brute force collision attack against H if H satisfies the following criteria:

1. *there is a set $\mathcal{I}$ of at least four IHV differences D for which there exists near-collision attacks against Compress that for fixed IHV_{in} and IHV'_{in} searches for message blocks B and B' such that*

   $$IHV'_{out} - IHV_{out} = IHV'_{in} - IHV_{in} - D$$

   where

   $$IHV_{out} = Compress(IHV_{in}, B), \quad IHV'_{out} = Compress(IHV'_{in}, B').$$

2. *each of the above near-collision attacks can be modified to start with any given $\widehat{IHV}_{in}$ and $\widehat{IHV}'_{in}$ such that the total average complexity in calls to Compress, consisting of the average cost of modifying the near-collision attack together with the average runtime complexity of the modified near-collision attack, is at most $\sqrt{\pi \cdot 2^{L-1}}/4$.*

3. *Given any $IHV_{in}, IHV'_{in} \in \{0,1\}^L$, we can assume that the probability that $Compress(IHV_{in}, B) = Compress(IHV'_{in}, B')$ for uniformly-randomly chosen $B \neq B' \in \{0,1\}^N$ is 2^{-L}.*

4. *there exists a function* $\phi : \{0,1\}^L \times \{0,1\} \to \{0,1\}^K$ *with* $K < L$ *such that the probability* $p_{\texttt{useful}}$ *that* $y - x \in \mathcal{I}$ *for IHV-values* x *and* y *with* $\phi(x,0) = \phi(y,1)$ *is at least* $\max(8 \cdot 2^{K-L}, \pi \cdot 2^{-K})$.

The second criterion can be partly fulfilled by Theorem 3.5 as it provides a way to modify a differential path underlying a near-collision attack to start with any given $\widehat{IHV}_{\text{in}}$ and $\widehat{IHV}'_{\text{in}}$. The proof of Theorem 3.6 depends on the following lemma:

Lemma 3.7 (Birthday search [vOW99])**.** *Let* V *be a finite set of size at least 3. Let* $f : V \to V$ *be a function such that* $\Pr[f(a) = f(b) \mid a, b \in V, a \neq b] = 1/|V|$. *Let* $\texttt{Pred} : V \times V \to \{0,1\}$ *be a boolean predicate defined on* $V \times V$ *such that the probability* $q = \Pr[\texttt{Pred}(a,b) = 1 \mid a, b \in V, a \neq b, f(a) = f(b)]$ *is non-zero.*[12]

Consider the following experiment. For $i = 1, 2, \ldots$, *sample* q_i *uniformly random from* $V \setminus \{q_1, \ldots, q_{i-1}\}$, *evaluate* f *on* q_i *and store* $(q_i, f(q_i))$. *The experiment continues until there exists an* $j \in \{1, \ldots, i-1\}$ *such that* $f(q_i) = f(q_j)$ *and* $\texttt{Pred}(q_i, q_j) = 1$.

Then the expected number of evaluations of f *required to find a pair* $(a,b) \in V \times V$ *such that* $a \neq b$, $f(a) = f(b)$ *and* $\texttt{Pred}(a,b) = 1$ *is* $\sqrt{\pi \cdot |V|/(2 \cdot q)}$.

We refer to [vOW99] for the proof of Lemma 3.7 and a practical probabilistic algorithm to perform a birthday search.

One can directly construct a brute force collision search against H (if it satisfies the third criteria of Theorem 3.6) with an average complexity of $\sqrt{\pi \cdot 2^{L-1}}$ calls to Compress, by using the above lemma with $V = \{0,1\}^L$ and $q = 1$ (a trivial predicate).

Proof of Theorem 3.6. Let $K < L$, $\phi : \{0,1\}^L \times \{0,1\} \to \{0,1\}^K$ and p_{useful} be as in Theorem 3.6. The chosen-prefix collision attack consists of a birthday search followed by a single near-collision attack. For given prefixes P and P', we first append bit strings S_{r} and S'_{r} such that bit lengths of $P||S_{\text{r}}$ and $P'||S'_{\text{r}}$ are both equal to $c \cdot N - K$ for some $c \in \mathbb{N}^+$ (note that $K < L \leq N$). Let IHV_{in} and IHV'_{in} be the intermediate hash values after processing the first $c-1$ blocks, i.e., the first $(c-1) \cdot N$ bits, of $P||S_{\text{r}}$ and $P'||S'_{\text{r}}$, respectively. Furthermore, let B and B' be the last $N - K$ bits of $P||S_{\text{r}}$ and $P'||S'_{\text{r}}$, respectively.

The birthday search is defined using ϕ and a birthday search space $V = \{0,1\}^K$. Let $\tau : V \to \{0,1\}$ be some balanced selector function, i.e., $|\tau^{-1}(0)| = |\tau^{-1}(1)|$. Then the birthday step function $f : V \to V$ is defined as

$$f(v) = \begin{cases} \phi(\text{Compress}(IHV_{\text{in}}, B||v), 0) & \text{if } \tau(v) = 0; \\ \phi(\text{Compress}(IHV'_{\text{in}}, B'||v), 1) & \text{if } \tau(v) = 1. \end{cases}$$

A birthday search collision $f(v) = f(w)$ with $v \neq w$ satisfies predicate `useful` if $\tau(v) \neq \tau(w)$ and $IHV'_{\text{out}} - IHV_{\text{out}} \in \mathcal{I}$ where

$$IHV_{\text{out}} = \text{Compress}(IHV_{\text{in}}, B||v), \quad IHV'_{\text{out}} = \text{Compress}(IHV'_{\text{in}}, B'||w),$$

12. This implies that collisions $a \neq b$ with $f(a) = f(b)$ and $\texttt{Pred}(a,b) = 1$ exist.

assuming without loss of generality that $\tau(v) = 0$ and $\tau(w) = 1$. As $f(v) = f(w)$ and $\tau(v) = 0 \neq \tau(w)$ implies that $\phi(IHV_{\text{out}}, 0) = \phi(IHV'_{\text{out}}, 1)$, it follows that the probability of a birthday search collision satisfying predicate `useful` is $p_{\texttt{useful}}/2$. Using Lemma 3.7, we can conclude that the average birthday search complexity is

$$\sqrt{\frac{\pi \cdot 2^K}{2 \cdot p_{\texttt{useful}}/2}} = \sqrt{\frac{\pi}{p_{\texttt{useful}}}} \cdot 2^{K/2}.$$

As $p_{\texttt{useful}} \geq \pi \cdot 2^{-K}$, one can expect a 'useful' birthday collision before the entire search space has been exhausted. Furthermore, since $p_{\texttt{useful}} \geq 8 \cdot 2^{K-L}$, the average birthday search complexity is at most half that of a brute force collision attack against H.

A successful birthday search gives us bit strings $S_{\text{b}} = v$ and $S'_{\text{b}} = w$ such that $D = IHV'_c - IHV_c \in \mathcal{I}$, where IHV_c and IHV'_c are the intermediate hash values after processing the first c blocks, i.e., the first $c \cdot N$ bits, of $P||S_{\text{r}}||S_{\text{b}}$ and $P'||S'_{\text{r}}||S'_{\text{b}}$, respectively. To finish we need to construct a near-collision attack that starts with IHV-values IHV_c and IHV'_c and searches for message blocks S_{c} and S'_{c} such that

$$IHV'_{c+1} - IHV_{c+1} = IHV'_c - IHV_c - D = 0 \quad \Rightarrow \quad IHV'_{c+1} = IHV_{c+1},$$

where

$$IHV_{c+1} = \text{Compress}(IHV_c, S_{\text{c}}), \quad IHV'_{c+1} = \text{Compress}(IHV'_c, S'_{\text{c}}).$$

After this successful near-collision attack we have obtained our chosen-prefix collision:

$$H(P||S_{\text{r}}||S_{\text{b}}||S_{\text{c}}) = H(P'||S'_{\text{r}}||S'_{\text{b}}||S'_{\text{c}}).$$

Due to the second criteria of Theorem 3.6, the complexity of constructing and executing this near-collision attack is at most one quarter of the complexity of a brute force collision attack against H.

Our chosen-prefix collision attack against H has an average complexity of three quarters of the complexity of a brute force collision attack against H. □

3.6 Recommendations

Based on the results presented in this thesis, we offer the following recommendations on the use of hash functions.

1. Given the current state-of-art in the cryptanalysis of MD5, we urgently recommend to immediately replace MD5 by a more secure hash function (e.g., SHA-2) in all applications, unless it is known that the application's security does not depend on the collision resistance of MD5.

   Especially in the case of MD5, it has been argued at the time of the first collision attack that such collision attacks do not pose a realistic threat due to attack

complexity, the problem of creating meaningful collisions and other technical constraints. However, we provide several examples in Chapter 4 that show that these arguments did not withstand the progress in cryptanalytic techniques and the creativity of researchers in constructing realistic threats.

2. Given the current state-of-art in the cryptanalysis of SHA-1, we strongly recommend to replace SHA-1 by a more secure hash function (e.g., SHA-2) in all applications, unless it is known that the application's security does not depend on the collision resistance of SHA-1.

   It can be argued, as it has been before with MD5 and was later refuted, that the current collision attacks on SHA-1 do not pose a threat due to too high attack complexities. However, our new differential cryptanalysis presented in Section 7.5 provides the exact analysis of combinations of local collisions that has been lacking so far. Furthermore, we provide several possibilities to improve our first attempt at a SHA-1 collision attack. Thus the SHA-1 collision attack complexity will very likely decrease in the future.

3. Furthermore, we recommend to use hash functions in a randomized hashing scheme (c.f. [HK06]), where possible, in such a manner that the randomness cannot be influenced by a possible adversary.

   In particular this holds for digital signatures as one should never sign any document whose entire contents can be controlled or determined by a possibly malicious party (e.g., see Section 4.2).

4. It is often observed that replacing a cryptographic primitive in applications can be a lengthy process, therefore we recommend to start such a replacement process as soon as indications of security weaknesses arise and not wait until sufficient proof has been given that these security weaknesses can be abused.

5. In the event the above recommendations apply but cannot be implemented in the near-future (e.g., due to compatibility issues, hardware designs, etc.), we offer the recommendation of implementing a barrier against collision-attacks using the algorithms presented in Chapter 8.

3.7 Directions for further research

We also offer some possible directions for further research:

1. Our disturbance vector analysis presented in Section 7.5 might be improved by a more compact definition of differential paths over steps $t_b, \ldots, t_e$ by replacing ΔQ_{t_e} with the pair $(\delta Q_{t_e}, \delta RL(Q_{t_e}, 5))$ which represents several possible allowed values for ΔQ_{t_e}. The definition for the probability of these differential paths should be adjusted accordingly. This improvement will reduce the runtime and memory cost of the analysis.

2. The reduction algorithm of Section 7.5.4 can be extended to remove more differences of a differential path $\mathcal{P}$ as long as:

$$\Pr[\mathcal{P}_{\text{ext}}] = \Pr[\mathcal{P}_{\text{redext}}] \cdot \frac{\Pr[\mathcal{P}]}{\Pr[\mathit{Reduce}(\mathcal{P})]},$$

for all (forward or backward) extensions $\mathcal{P}_{\text{ext}}$ of $\mathcal{P}$ and the analogical extensions $\mathcal{P}_{\text{redext}}$ of $\mathit{Reduce}(\mathcal{P})$. This improvement will reduce the runtime and memory cost of the analysis.

3. As described in Section 7.6.8 one can also use a tunnel if it breaks only message bitrelations over m_{14} and m_{15}. Such tunnels can be used to speed up steps 14 and 15. However, the true effect of these tunnels on the optimization and runtime complexity of the near-collision attack is subject for further research.

4. So far we have analyzed one disturbance vector and made an early attempt at a near-collision attack in Section 7.6. More research is necessary to analyze other disturbance vectors and other combinations of tunnels and speedup techniques. This would be aided greatly by an algorithmic construction of a (near-)optimal near-collision attack given disturbance vector, differential path, message bitrelations and a set of tunnels.

5. The implementation of our MD5 chosen-prefix collision attack [HC] can make use of massively parallel computing architectures, such as CELL SPU (PlayStation 3) and CUDA (graphic cards), that offer a significantly higher performance per cost ratio compared to common computing architectures such as x86, AMD64, PowerPC, etc. However, so far this speedup is strictly limited to the birthday search. Further research would be necessary to determine to what extent the near-collision attacks against MD5 and SHA-1 could benefit from massively parallel computing architectures.

6. The SHA-1 chosen-prefix collision attack complexity presented in Section 7.7 is dominated by the complexity of the birthday search, which is directly related to the size of the reasonably structured set $\mathcal{I}$ of δIHV_{diff}-values that can be efficiently eliminated with near-collision attacks. It is unclear which disturbance vector leads to the largest such set $\mathcal{I}$ while keeping the near-collision attack complexity relatively low. Furthermore, one could even combine several such sets $\mathcal{I}$ from different disturbance vectors similar to how the chosen-prefix collision attack on MD5 uses 32 possible message differences. This could significantly decrease the complexity for SHA-1 chosen-prefix collisions.

7. How the cryptanalytical techniques used for MD5 and SHA-1 can aid the cryptanalysis of other hash functions or other cryptographic primitives.

3.8 Thesis outline

The remainder of this thesis consists of the following parts: Chapter 4 discusses the application of chosen-prefix collision attacks in a number of abuse scenarios. In

particular it describes in detail our most important successful application of a chosen-prefix collision attack in Section 4.2, namely the construction of a rogue Certification Authority certificate.

In Chapter 5 we introduce the definition for the family $\mathcal{F}_{\text{md4cf}}$ of compression functions and differential paths thereof. Furthermore, we prove Theorem 3.5 by presenting an algorithm for constructing complete differential paths which is a generalization of the algorithms for MD5 and SHA-1 presented in Chapter 6 and Chapter 7, respectively.

Chapter 6 focuses on MD5 and provides an algorithm for constructing complete differential paths for MD5 together with a collision finding algorithm algorithm that searches for actual near-collision blocks given a differential path. Finally this section proves Theorem 3.1 by presenting the practical identical-prefix and chosen-prefix collision attacks with the respective claimed complexities.

Similarly, Chapter 7 focuses on SHA-1 and provides an algorithm for constructing complete differential paths for SHA-1. This is followed by the presentation of our SHA-1 disturbance vector analysis. At the end of Chapter 7 we present the construction of a practical near-collision attack which proves Theorem 3.3.

Chapter 8 presents a technique to detect an identical-prefix or chosen-prefix collision given only one message of the collision pair. Such a technique may prevent abuse of collision attacks. We apply this technique to MD5 and SHA-1 in Sections 8.3 and 8.4, respectively.