УДК 681.326

ПОВЫШЕНИЕ НАДЕЖНОСТИ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ОСНОВЕ СМАРТ-КАРТ

НЕМЧЕНКО С.В.

Рассматривается использование смарт-карт в гетерогенной распределенной информационной системе. Доказывается целесообразность введения параметра времени в смарт-карты в целях повысить надежность функционирования системы. Анализируются требующиеся изменения как в аппаратном, так и в программном обеспечении смарт-карт.

При реализации проектов создания гетерогенных распределенных информационных систем разработчики на различных этапах все чаще применяют объектные технологии и методологии построения такого рода систем. В качестве примера таких технологий можно назвать CORBA, Java, DCOM и т.п.

В настоящее время в качестве одной из составляющих распределенных систем используются так называемые смарт-карты (СК) (Smart Cards) и, в частности, их разновидность Java Cards [1].

Расширение области применения СК на практике привело к необходимости сделать их еще более "гибкими", более "интеллектуальными", а главное – более надежными. Существует множество различных возможностей сделать это. В данной статье мы рассмотрим возможность введения в СК параметра времени.

Напомним, что под СК мы будем понимать стандартную пластиковую карту (например, типа Visa Card), в которую наряду (или вместо) с магнитной полосой встроен микропроцессорный элемент. Он начинает работать с момента, когда карта вставлена в считыватель. При этом на встроенный в карту чип подается питающее напряжение и происходит коммутация входов и выходов СК и считывателя. До этого момента микропроцессор карты находится в пассивном состоянии. С момента подключения СК становится активным элементом гетерогенной распределенной информационной системы.

Широкое использование СК в различных областях жизни и деятельности современного человека приводит к необходимости усилить меры безопасности, связанные с защитой информации, хранящейся на карте. Приведем несколько примеров использования параметра времени для защиты информации.

– *Временной контроль.* Часто преступления, связанные с использованием СК, происходят в "нерабочее" время. Например, ночью преступник пытается войти в помещение, используя в качестве ключа карту доступа. При введении в карту параметра времени мы получаем возможность регламентировать допустимое время ее использования.

– *Ограничение времени выполнения операций.* Большинство операций, связанных со взломом информации, требуют более длительного времени на их исполнение по сравнению с обычными операциями. Вводя ограничения на допустимое время выполнения определенных операций, мы тем самым повышаем защиту карты.

– *Периодическое изменения ключа.* Известно, что надежность ключа, используемого при кодировании информации, с течением времени снижается. Введя параметр времени в карту, мы даем ей возможность периодически менять свой ключ, повышая его надежность.

– *Временная подпись.* Известны примеры использования СК при осуществлении электронной подписи. Введение параметра времени в электронную подпись позволит в определенной степени повысить ее надежность.

– *Временная атака* [2]. Это новый тип криптоатаки, основанный на том, что современные методы шифрования информации, в том числе и те, которые применяются в СК, построены на большом количестве математических вычислений. При этом, оказалось, что анализ временных характеристик выполняемых вычислений может дать информацию о ключе к шифру. А это уже большой шаг к взлому шифра. Суть защиты от временной атаки заключается в маскировании реальной длительности выполняемых операций. В данном случае это возможно лишь при введении параметра времени в СК.

Введение в карту параметра времени поможет во многом решить как рассмотренные выше, так и другие задачи. При этом возникает необходимость изменения как аппаратного, так и программного обеспечения карты. Рассмотрению этих и некоторых других вопросов посвящена данная статья.

Базовые понятия

Введем некоторые базовые понятия [3].

Событие. Если предполагать, что время течет из прошлого в будущее, тогда можно сказать что "событие" - это точка, находящаяся между прошлым и будущим. Его длительность равна нулю.

Временная переменная (ВП) задает некоторый момент времени. Она может определять конкретный момент времени *(например:* 19-05-2000 17:00:00) или совокупность моментов. Во втором случае переменная может представлять собой либо некоторый цикл *{например:* вторник 10:00:00), либо совокупность моментов, идущих подряд *(например:* 1-е января 2000-го года).

Временные функции описывают взаимоотношения между ВП. Эти функции возвращают значение *true* или *false*. Предлагается ввести две базовые временные функции:

– *PAST(tv)* возвращает значение *true* для всех ВП *(tv),* которые находятся в прошлом по отношению к ВП, описывающей событие "сейчас", и *false* – для всех остальных ВП;

– *FUTUR(tv)* возвращает значение *true* для всех ВП *(tv),* которые находятся в будущем по отношению к

ВП, описывающей событие “сейчас”, и *false* – для всех остальных ВП.

Временная логическая функция строится объединением временных функций при помощи классических логических функций *and, or, not.*

Пример 1: Предположим, что имеем две ВП *TV1* и *TV2.* Переменная *TV1* описывает событие, которое во времени находится перед событием *TV2.* Тогда мы можем создать такую временную логическую функцию:

$$T = FUTUR(EVl) \text{ and } PAST(EV2).$$

Таким образом, функция Т возвращает значение *true* на интервале после *TV1,* но перед *TV2* (рис.1).

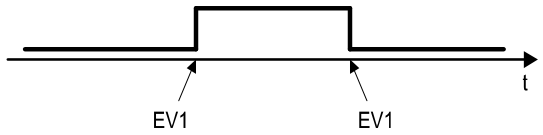

Рис. 1. Пример временной логической функции Т

Базируясь на приведенных выше временных логических функциях можно создать новые временные функции. Например, предлагается ввести функцию *NOW(tv).* Как было отмечено ранее, длительность любого события равна нулю. ВП может объединять совокупность событий *(например, tv* = {год – 2001, месяц – апрель, понедельник}) – функция *NOW(tv)* возвращает булево значение в зависимости от принадлежности события “сейчас” к ВП *tv.*

Таким образом, можно ввести новое понятие: *временной интервал.* “Простейший” интервал является совокупностью событий, идущих подряд одно за другим. *Временной интервал* может состоять из совокупности простейших интервалов.

Пример 2. С понедельника 10:00:00 до пятницы 17:00:00. Такой интервал является временным и будет повторяться каждую неделю.

Аппаратная реализация

Для введения параметра времени (таймера) в состав микропроцессорной карты необходимо, прежде всего, иметь генератор частоты и счетчик, в котором будет храниться текущее время. В данной работе предлагается подсчитывать в качестве значения текущего времени количество тактов после инициализации таймера. Для уменьшения разрядности счетчика и для упрощения его функционирования предлагается также добавить делитель частоты на выходе генератора тактовой частоты. Предполагается, что таймер должен функционировать на протяжении всего времени службы карты.

Приняв во внимание все сказанное выше, введем в состав СК дополнительно следующие блоки:

– генератор тактовой частоты (ГТЧ);

– делитель частоты для получения задающей частоты 1Гц;

– двоичный счетчик;

– электрическую батарею для осуществления постоянного питания таймера.

Структурная схема таймера приведена на рис. 2.

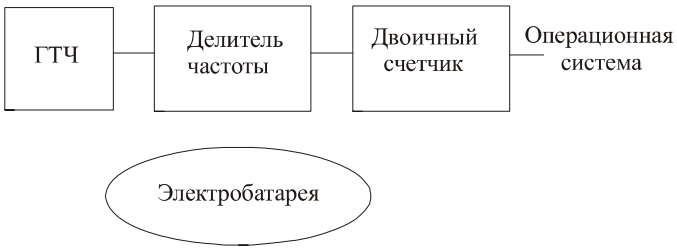

Рис. 2. Структура таймера в СК

Спецификация команд. Формат времени

В данном случае существует три метода представления внутреннего времени карты:

1. Использование значения счетчика, т.е. подсчет количества секунд после инициализации карты.

2. Использование времени в “обычном” формате (секунды/минуты/часы/день/месяц/год).

3. Использование смешанного метода, который представляет собой комбинацию первых двух.

В первом случае мы имеем наиболее “естественный” с аппаратной точки зрения способ задания времени. Однако при этом невозможно использовать циклические даты и интервалы. К тому же это усложняет процесс коррекции времени.

Второй метод упрощает использование всех типов ВП и интервалов. Коррекция времени может быть легко добавлена в процесс перевода значения времени из значения счетчика в “обычный” формат. Но чтобы использовать текущее время, необходимо осуществить операцию перевода из одного формата в другой. А это требует временных и аппаратных затрат. К тому же при использовании нециклических дат усложняется сравнение времени *(past, future)* по сравнению с первым методом.

Третий метод мы не рассматриваем в данной работе, поскольку он серьезно усложняет использование временных команд, так как при этом предполагается одновременное использование обоих методов, рассмотренных выше.

Остановимся на втором методе, поскольку он при относительной простоте отвечает основным требованиям. При этом главным его недостатком являются временные затраты при получении текущего значения времени.

Для реализации сказанного создается новый формат *CTF (Card Time Format).* Его структура приведена в табл. 1. Как следует из таблицы 1, формат CTF использует 42 бита. Он является универсальным и позволяет применять любой тип времени и даты.

Таблица 1

Формат CTF

	С	Мин	Час	День недели	Неделя	День месяца	Месяц	Год
Кол. бит	6	6	5	3	6	5	4	7
Значение	0-59	0-59	0-23	1-7	1-52	1-31	1-12	0-99

Базовые команды

Операционная система (ОС) карты располагает единственной командой, имеющей непосредственный доступ к значению счетчика тактов. Это команда *get_time*. Она возвращает текущее время в формате *CTF*. Команды сброса и установки значения счетчика отсутствуют. Это повышает надежность внутреннего времени карты (защита от внешних атак). Счетчик запускается сразу после того, как подается питание от батареи. В течение всего времени функционирования карты значение счетчика никогда не меняется извне.

Для удобства использования формата *CTF* существует группа команд для получения определенной части даты *(например: год, час или др.)*. Точнее сказать, это команды преобразования формата *CTF* в нормальный формат, т.е. дата делится на свои составляющие.

Аналогично существует группа команд, выполняющих обратную операцию. Дата в формате *CTF* компонуется из отдельных составляющих. Это позволяет задавать как нормальные, так и частично определенные события *(например, вторник 8 часов утра)*.

Команды временных логических функций

На основе созданных ВП и с помощью функций *past, future, now* пользователь может создавать логические функции. Также можно представлять ВП с помощью команд *start_delay* и использовать потом их посредством команды *delay*.

Все перечисленные выше функции возвращают булево значение. Поэтому их можно объединять при помощи обычных булевых команд *and, or, not*. Таким образом можно создавать достаточно сложные функции, используя простые временные функции.

Хронологическая таблица

Кроме отмеченного выше, предлагается ввести в карту хронологическую таблицу по аналогии с командой *crontab* в Unix. Команде задается функция и время ее запуска. Можно задать циклический запуск функции. Чтобы упростить программирование времени запуска и расширить диапазон возможных величин цикличности, вводится параметр, определяющий цикличность.

Пример 3. Необходимо запускать на исполнение функцию "add" каждый понедельник в 8 часов утра (табл.2).

Пример 4. Необходимо запускать на исполнение функцию "add" каждый первый понедельник месяца в 8 часов утра (табл. 3).

Следует отметить, что в настоящей работе предложен общий подход к решению задачи введения параметра времени в СК. Кроме рассмотренных выше, есть еще

Таблица 2

Пример 3

С	Мин	Час	День недели	Неделя	День месяца	Месяц	Год	Цикл	Функция
0	0	8	1	*	*	*	*	4	add

Таблица 3

Пример 4

С	Мин	Час	День недели	Неделя	День месяца	Месяц	Год	Цикл	Функция
0	0	8	1	*	*	*	*	6	add

ряд вопросов, которые предстоит решить. Так, в частности, остался нерешенным вопрос схемной реализации ГТЧ с заданным параметром точности и стабильности его работы. При этом следует помнить, что в данном случае имеет место жесткое ограничение на такой параметр как физические размеры элементов. В случае ГТЧ этот параметр становится весьма критичным, так как известные реализации генераторов на кварцевом элементе не поддаются микроминиатюризации, достаточной для помещения их в корпус пластиковой карты. Скорее всего, в этом случае речь может идти о реализации миниатюрного RC-генератора. Однако при этом мы сталкиваемся с другой проблемой – повышение точности и стабильности работы такого генератора.

Кроме того, не следует забывать и о такой "мелочи", как необходимость использования сверхплоского элемента питания с достаточно большим сроком работы, как минимум три года – средний срок службы одной карты.

В заключение следует отметить, что сегодня многие разработчики и производители СК вплотную подошли к осознанию необходимости введения в них параметра времени . Об этом свидетельствует хотя бы тот интерес, который вызывали сообщения на данную тему, сделанные автором во Франции и в Нидерландах в 1999 году.

Литература: 1.*Кордонье В., Немченко В.П., Кривуля Г.Ф., Немченко С.В. /* Использование интеллектуальных карт в информационном пространстве современного общества // Радиоэлектроника и информатика. 1998. № 2. С.125-127. 2. *Dhem J.-F. , Koeune F., Leroux P.-A., Mestre P., Quiquater J.-J., Willems J.-L.* Apartical implentation of the timing attack. CARDIS. 1998. 3. *Cordonnier V., Watson A., Nemchenko S.* / Time as an aid to improving security in smart cards / 7th Annual Working Conference on Information Security Management and Small Systems Security. Amsterdam, The Netherlands. 1999.

Поступила в редколлегию 26.02.2000

Рецензент:

Немченко Сергей Владимирович, магистр технических наук, аспирант ХТУРЭ. Имеет диплом DEA Informatique Лилльского университета науки и технологий, Франция. Во время учебы в Лилле в 1998-1999 гг. участвовал в научных исследованиях, связанных с разработкой и совершенствованием смарт-карт. Научные интересы: смарт-карты, техническая диагностика. Увлечения и хобби: автомобилизм, спорт, музыка, путешествия. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (0572) 40-93-26.